Visibility-oriented Visualization Design for Flow Illustration

Andrea Brambilla

Dissertation for the degree philosophiae doctor (PhD)
at the University of Bergen

2014

Dissertation date: December 18

Scientific Environment

The research activities presented in this dissertation have been conducted in the Visualization Group at the Department of Informatics, University of Bergen. Part of the work has been carried out in the context of the project "SemSeg - 4D Space-Time Topology for Semantic Flow Segmentation", funded by the European Commission in the Future and Emerging Technologies (FET) Programme within the Seventh Framework Programme for Research (grant number 226042). I have also been enrolled in the ICT Research School at the Department of Informatics at the University of Bergen.

UNIVERSITETET I BERGEN
Det matematisk-naturvitenskapelige fakultet

ICT

Research School In
Information and Communication Technology

Acknowledgments

I would like to express my deepest gratitude to my supervisors Helwig Hauser, Ivan Viola and Øyvind Andreassen. Helwig introduced me to the field of visualization and taught me how to carry out successful research projects. Ivan frequently provided interesting and creative points of view on my research. Øyvind always made sure that my work was built upon solid mathematical and physical bases. But most importantly, I would like to thank them all for keeping up with my extreme stubbornness and for guiding me towards the completion of this thesis.

The realization of this manuscript is in large part due to the support from the people around me. My girlfriend, Martina, deserves my uttermost gratitude. She is the one who hold me up in the darkest moments of my Ph.D. and I would have surely given up without her at my side. I would also like to thank my parents, Aurelio and Stefania, and my sister, Claudia, for being an important part of my life regardless of the geographical distance. They always make me feel at home whenever I visit them (and terribly homesick whenever I leave), and for this I am extremely grateful.

A special thank goes to my ex-colleague and friend Armin Pobitzer, for the frequent technical and scientific discussions, for the even more frequent non-technical and non-scientific discussions, and for sharing with me an overachieving low-budget coffee machine, which still works perfectly after more than 5000 espressos over 4 years.

Thanks also to all my friends here in Bergen for all the parties and all the nights out, for all the days spent in the mountains, and for making the last four years a memorable and delightful experience. Thanks to my current and past colleagues for the fruitful discussions and for making our corridor a great work environment. I would also like to thank a countless number of friends in Italy, for always making me feel welcome and for doing their best to spend time with me whenever it is possible.

Finally, I would like to thank a number of people and organizations who provided datasets, portions of code, and feedback: Josué Quilliou and GexCon AS (Bergen, Norway); Krešimir Matkovič, Jürgen Waser, the VRVis research center and AVL List GmbH (Vienna, Austria); Carl Erik Wasberg and FFI (Kjeller, Norway); Holger Theisel, Maik Schulze and Kai Lawonn (University of Magdeburg, Germany).

I acknowledge the financial support from the University of Bergen, the SemSeg project, the Meltzerfondet and the TU Wien Ausseninstitut.

Abstract

Flow phenomena are ubiquitous in our world and they affect many aspects of our daily life. For this reason, they are the subject of extensive studies in several research fields. In medicine, the blood flow through our vessels can reveal important information about cardiovascular diseases. The air flow around a vehicle and the motion of fluids in a combustion engine are examples of relevant flow phenomena in engineering disciplines. Meteorologists, climatologists and oceanographers are instead concerned with winds and water currents.

Thanks to the recent advancements in computational fluid dynamics and to the increasing power of modern hardware, accurate simulations of flow phenomena are feasible nowadays. The evolution of multiple flow attributes, such as velocity, temperature and pressure, can be simulated over large spatial and temporal domains (4D). The amount of data generated by this process is massive, therefore visualization techniques are often adopted in order to ease the analysis phase. The overall goal is to convey information about the phenomena of interest through a suitable representation of the data at hand. Due to the multivariate and multidimensional nature of the data, visibility issues (such as cluttering and occlusion), represent a significant challenge.

Flow visualization can greatly benefit from studying and addressing visibility issues already in the design phase. In this thesis we investigate and demonstrate the effectiveness of taking visibility management into account early in the design process. We apply this principle to three characteristic flow visualization scenarios: (1) The simultaneous visualization of multiple flow attributes. (2) The visual inspection of single and multiple integral surfaces. (3) The selection of seeding curves for constructing families of integral surfaces. Our techniques result in clutter- and occlusion-free visualizations, which effectively illustrate the key aspects of the flow behavior.

For demonstration purposes, we have applied our approaches to a number of application cases. Additionally, we have discussed our visualization designs with domain experts. They showed a genuine interest in our work and provided insightful suggestions for future research directions.

Related Publications

This thesis is based on the following publications:

Paper A: **Andrea Brambilla**, R. Carnecky, R. Peikert, I. Viola, H. Hauser.
Illustrative flow visualization: State of the art, trends and challenges.
Proc. of *Eurographics 2012-State of the Art Reports*, 75–94, 2012.

Paper B: **Andrea Brambilla**, Ø. Andreassen, H. Hauser.
Integrated Multi-aspect visualization of 3D Fluid Flows.
Proc. of *VMV 2013: Vision, Modeling & Visualization*, 1–9, 2013.

Paper C: **Andrea Brambilla**, I. Viola, H. Hauser.
A Hierarchical Splitting Scheme to Reveal Insight into Highly Self-Occluded Integral Surfaces.
Journal of WSCG, 20, 1, 57–64, 2012.

Paper D: **Andrea Brambilla**, P. Angelelli, Ø. Andreassen, H. Hauser.
Comparative Visualization of Multiple Time Surfaces by Planar Surface Reformation.
Submitted to: *IEEE Trans. Visualization and Computer Graphics*, 2014.

Paper E: **Andrea Brambilla**, H. Hauser.
Expressive Seeding of Multiple Stream Surfaces for Interactive Flow Exploration.
Under revision after a first submission to: *Computers & Graphics*, 2014.

The following publications are also related to this thesis:

Paper 1: Å. Birkeland, S. Bruckner, **Andrea Brambilla**, I. Viola.
Illustrative Membrane Clipping.
Computer Graphics Forum, 31, 3, 905–914, 2012.

Paper 2: J. Parulek, **Andrea Brambilla**.
Fast Blending Scheme for Molecular Surface Representation.
IEEE Trans. Visualization and Computer Graphics, 19, 12, 2653–2662, 2013.

The work described in these papers has been carried out during the Ph.D. period of the thesis' author. The thesis' author is the main contributor of **Papers A** to **E**, which are all co-authored by the main supervisor of the thesis, Helwig Hauser (University of Bergen, Norway). Hauser had an essential role in identifying relevant research problems and actively contributed in the search for effective solutions. He also contributed by constantly providing guidance and advice, in addition to taking part to the frequent fruitful discussions.

Papers A and **C** were written in collaboration with Ivan Viola (Vienna University of Technology, Austria), one of the two co-supervisors of this thesis. Viola contributed with his deep knowledge of *illustrative visualization*, which is a central topic in these papers and in this whole manuscript. The regular exchanges between Viola and the thesis' author were a steady source of inspiration. **Paper A** is also co-authored by Robert Carnecky and Ronald Peikert (Swiss Federal Institute of Technology, Switzerland), who assisted in refining the content of the paper and proposed several improvements.

Papers B and **D** are the result of a collaboration with the second co-supervisor of this thesis, Øyvind Andreassen (Norwegian Defence Research Establishment, Kjeller, Norway). Andreassen acted as a domain expert and provided valuable guidance on the mathematical and physical foundations of these papers. He also contributed to the evaluations of the proposed visualization techniques. Paolo Angelelli (University of Bergen, Norway) co-authored **Paper D**. He contributed with his knowledge of comparative visualization, in addition to writing portions of the paper.

Paper D has been recently submitted to *IEEE Transaction on Visualization and Computer Graphics*. **Paper E** has been submitted to *Computers & Graphics* in July 2014. The reviewers suggested a major revision, which is currently being prepared. The revised paper will be submitted in late October. The versions of these two papers included in the thesis correspond to the original journal submissions.

The thesis' author contributed to **Paper 1** by providing his knowledge of physical simulations in the context of computer graphics. Concerning **Paper 2**, he collaborated with Julius Parulek (University of Bergen, Norway) in developing the mathematical foundations of the paper. In both cases, he also contributed by writing portions of the papers.

Contents

Part I

Overview

Chapter 1

Introduction

The term *flow* can refer to a multitude of concepts across different research fields. This thesis is mainly concerned with *fluid flows*, that is, the motion of fluids in space and time. This kind of phenomena is investigated in a large number of scientific disciplines. For instance, *fluid dynamics* is the branch of physics dedicated to studying the fundamental laws behind the motion of fluids. Medicine is instead concerned with how the flow of blood in our vessels affects our health condition. Climatology, meteorology and oceanography focus on the study of winds and water currents across the globe. In the context of engineering, the performances of several mechanical devices (such as engines, turbines and propellers) are highly dependent on how fluids move around and through the devices themselves. In short, flow phenomena have a significant impact on our daily life. Studying and understanding them can significantly affect our ability to shape the world we live in.

The study of a physical phenomenon begins with a *quantification* process, which results in a description of the phenomenon through numerical quantities [Lem96]. The quantification of flow phenomena has been traditionally carried out by experimental methods, such as dye injection, particle image velocimetry and schlieren photography [TYF07]. Thanks to the recent advancements in computational fluid dynamics and to the increasing capabilities of modern hardware, computational simulations have have become the most common methodology for flow quantification. Setting up and executing a flow simulation algorithm is considerably easier, faster and cheaper than its experimental counterpart [SS14]. Additionally, computational methods are able to reproduce the evolution of multiple flow attributes (such as velocity, temperature and pressure) simultaneously. The maximum size of the simulation domain is related to the available amount of memory. Tens of gigabytes of memory can be found even on commodity hardware. Additionally, out-of-core computation strategies can eliminate the memory limitations. As a consequence, the resulting flow datasets consist of a large number of data points, densely distributed over up to three spatial dimensions, plus a temporal extent. Each data point can be associated with numerous scalar, vectorial and tensorial quantities. Manipulating and studying such a substantial amount of data can be accomplished only with the assistance of automatic or semi-automatic methods.

The field of *visualization* investigates how to explore, understand and communicate data through the use of interactive visual meanings, such as images, animations and diagrams [TGM83]. In a traditional visualization pipeline, there are three main stages [HM90]: (1) The raw data is first preprocessed. (2) The preprocessed data is mapped to visual entities. (3) The visual entities are finally rendered on the visualization device. The branch of visualization dedicated to flow data is called *flow visualization* [Mer87], and its success is confirmed by its 30 years-long history. Its main goal is to support the study of flow phenomena the graphical depiction of the related flow data. Flow visualization techniques are nowadays employed, e.g., for illustrating and studying the blood flow in connection with cardiovascular conditions [ZIH+11, NLB+13, MMV+13]. In the context of mechanical engineering and vehicle design, flow visualization has a prominent role for the analysis and validation of CFD simulations [SPP04, SFB+12, GT14].

When dealing with flow data, typical outputs of the first stage of the visualization pipeline (the preprocessing) are derived attributes (pressure gradient, vorticity, etc.), integral structures (streamlines, stream surfaces, etc.) and flow features (vortex cores, shock waves, etc.). Different types of visual entities can be associated to each kind of data, and disparate rendering algorithms can be applied to the various visual entities. For instance, the raw velocity data can be mapped to arrow glyphs, which are generated on the fly by a geometry shader. An integral surface can be represented by a three-dimensional mesh, rendered via rasterization. Alternatively, the integral surface can be mapped to an implicit function and visualized by a ray-casting algorithm.

1 Visualization Design and Visual Quality

The process of *visualization design* consists in characterizing the preprocessing steps, the mappings to visual entities and the rendering algorithms to be employed [SI94]. In other words, a visualization design defines how the data describing the phenomenon of interest is ultimately represented on the target visualization device (typically a computer monitor). During the design phase, visualization experts have to take into account many factors, such as the type and the size of the data, the effectiveness of the visual entities compatible with the data, and the properties of the selected rendering techniques. The performance of a visualization design can be evaluated from different points of view, from the effectiveness of the overall visualization system to the visual quality of the generated images and animations [Mun09]. The visual quality is often a necessary condition for the usefulness and effectiveness of the visualization system. In fact, visualizations of poor visual quality are unlikely to covey relevant information about the phenomenon under examination.

We focus on the relationship between visualization design and visual quality. Specifically, we deal with the problem of *visibility management*. As mentioned

before, flow simulations can produce substantial amounts of multidimensional and multivariate data. In contrast, the visual resources provided by typical visualization devices are limited to two spatial dimensions, plus colors. A naive approach could be to "squeeze" all the available data in a single visualization. In such a scenario, the results are likely to be afflicted by severe visibility issues. For instance, the displayed visual entities would almost certainly occlude each other. Additionally, the large number of rendering primitives on the screen would make the scene cluttered and difficult to understand. Visibility issues can make a visualization ineffective and completely useless. It is therefore important to set up the visualization pipeline carefully in order to avoid this kind of problems.

2 Addressing Visibility Issues

Visibility management has been extensively investigated in the context of *illustrative visualization* [RBGV08]. Numerous illustrative techniques have been proposed in order to address visibility issues at specific stages of the visualization pipeline. For instance, data reduction strategies can be applied during the preprocessing phase. The most relevant portions of the data can be maintained and emphasized, while the rest can be sub-sampled, aggregated or simply discarded. Various properties of the visual entities, such as position, size and shape, can be optimized in order to reduce cluttering and self occlusion. Additionally, we can find in literature a number of visual entities which allow for multiple data attributes to be depicted at once (e.g., glyphs [dLvW93] and stream tubes [SVL91]). Rendering algorithms can count on specifically designed transparency strategies. Deformations, exploded views and cut-aways [VG05] can be also integrated in the rendering process.

We recognize that visibility management is critical in flow visualization (and in visualization in general). As such, we advocate that visibility management should be treated as a global delivery of the whole visualization pipeline, instead of the outcome of an individual algorithm or processing phase. Flow visualization can greatly benefit from studying and addressing visibility issues already in the design phase. When designing a visualization pipeline, devising a suitable visibility management strategy is of primary importance for achieving expressive depictions of the phenomenon of interest. In this thesis we investigate and demonstrate the effectiveness of taking visibility issues into account early in the design process. The main contribution of this thesis consists of the study and application of this principle to three typical flow visualization scenarios. Specifically, our contribution can be summarized as follows:

- We propose a novel visualization strategy for the simultaneous depiction of multiple aspects of the flow behavior. The underlying idea is the selective visualization of flow attributes according to their impact on the fluid's motion. Visibility issues are further reduced through a coherency-based

aggregation of visual entities. Our approach allows the users to study how the flow attributes of interest affect each other.

- We have investigated different methodologies for easing the visual inspection of single and multiple integral surfaces. Integral surfaces are highly effective in visualizing flow phenomena. We aim at making the surfaces easier to understand, while preserving the informative content as much as possible. Our techniques are based on cutting and deforming the geometry of the surfaces. We adapt vectorial and tensorial flow attributes according to changes in geometry. Additionally, we enable the direct comparison of entire families of integral surfaces.
- We introduce a novel algorithm for seeding multiple stream surfaces in the context of local flow exploration. Seeding is performed by taking into account the reciprocal similarity of a dense set of streamlines. This allows us to identify and visualize characteristic motion patterns in the flow, while consuming a minimal amount of visual resources.

3 Thesis Structure

This dissertation is composed of two parts. The first one (Part I) discusses the main findings and contributions of the work carried out in the context of the author's Ph.D. project. In particular, Chapter 2 introduces the research context, focusing on flow visualization, illustrative visualization, and the nexus between the two topics. Chapter 3 describes the theoretical concepts about the motion of fluids this thesis is based on. Our contribution is detailed in Chapter 4, while Chapter 5 includes a demonstration of the proposed approaches on four different application cases. Chapter 6 concludes the manuscript by summarizing the contribution and by presenting possibilities for future developments.

The second part (Part II) includes a reformatted version of the papers this thesis is based on. Paper A surveys the field of illustrative flow visualization in detail. Paper B discusses our strategy for visualizing multiple aspects of the flow behavior. Paper C and Paper D focus on the analysis and the comparison of integral surfaces. Paper E describes our novel surface seeding technique.

Chapter 2

Related Work

This chapter describes the foundations of visibility management in the context of flow visualization. We begin by providing an overview of flow visualization and the traditional classification of techniques in this field. Next, we introduce the illustrative visualization pradigm, which serves as an effective methodology for addressing perceptual and visibility issues of complex datasets. Then, we briefly discuss illustrative techniques targeted at flow phenomena. We focus on the illustrative visualization of field data and integral structures, since they represent the main kind of flow data this thesis deals with.

1 Introduction to Flow Visualization

Flow visualization is a branch of scientific visualization focused on the exploration and presentation of flow phenomena. The birth of flow visualization dates back to the eighties, and its continuous growth is due to the remarkable complexity of this kind of phenomena. In this thesis we are mainly concerned with fluid flows, i.e., the motion of fluid substances. With the term *flow dataset*, we refer to the data resulting from the quantification of the motion of a fluid, either by direct measurement or by simulation. A flow dataset can include several scalar, vector and tensor fields, each of them representing a specific flow attribute. Flow attributes can be rarely described by analytical formulas. In most cases, a large number of data samples is collected over the 1D, 2D or 3D spatial domain. The sampling locations are usually distributed according to a structured or unstructured grid [MLP+10]. An approximation of the original values can be computed through suitable interpolation schemes. The flow attributes can be either constant (steady flows) or change over time (unsteady flows). In the second case, the sampling has to be performed also along the time dimension.

The exceptional variability of flow datasets, and of flow phenomena in general, makes their visualization highly challenging. Over the years, this problem has been addressed by a multitude of flow visualization techniques from different points of view. Because of the large number of existing approaches, several classifications of the flow visualization field have been proposed. A widespread categorization is presented in the state-of-the-art report by Post et al. [PVH+02]

from 2002. They identified four categories: *direct*, *texture-based*, *geometric* and *feature-based* visualization.

Direct visualization techniques are characterized by a straightforward mapping between the flow data and the corresponding visual representation. These approaches can be considered *local*, in the sense that each visual entity conveys information from a specific location in the spatial domain. Typical examples include color coding, volume rendering and arrow plots.

In texture-based visualization, vector and tensor fields are visualized via *texture advection*. This procedure consists in computing a deformed version of an input 2D or 3D texture (usually a noise texture) according to local flow properties. The appearance of the results usually depends on the input texture and on the advection time. More details on this category of techniques can be found in the surveys by Sanna et al. [SMM00] and Laramee et al. [LHD+04].

Geometric visualization is based on the extraction of geometric structures from the flow data. When the extraction is performed through an integration process, the term *integral structures* is adopted. Integral structures are subdivided in curves, surfaces and volumes. A *streamline* is an integral curve obtained by integrating a steady velocity field from a selected point in the spatial domain, the *seeding point*. It is a line that starts at the seeding point and is everywhere tangent to the velocity vectors. The integration of an unsteady velocity field instead results in a *path line*. A path line depicts the trajectory of a massless particle released in the flow at the seeding point. A *streak line* resembles the physical experiment of continuously releasing colored dye in a fluid flow. It is defined as the set of particles that passed through the seeding point over a certain time interval. A set of seeding points distributed along a curve is called a *seeding curve*. A *time line* is obtained by simultaneously integrating particles from a seeding curve for a certain time.

By increasing the dimensionality of the seeding structure, the dimensionality of the integral structure increases as well. For instance, a *stream surface* is obtained by integrating an instantaneous (steady) flow field from a seeding curve. A *time surface* is given by the simultaneous integration of particles distributed over a *seeding surface*. Integral volumes can be computed as well, but they never attracted considerable attention. Geometric visualization techniques have been surveyed by McLoughlin et al. [MLP+10], while the state-of-the-art report by Edmunds et al. [ELC+12a] focuses on surface-based flow visualization.

Feature-based techniques convey the fluid's motion by visualizing *flow features*. A feature can be generally defined as an object or a structure in the data which satisfies particular requirements. Common examples of features in the context of flow visualization are shock waves, vortices, shear layers and vector field topology. A survey of feature-based flow visualization techniques has been presented by Post et al. [PVH+03]. Approaches based on topological features, such as vector field topology and Lagrangian coherent structures, are surveyed by Laramee et al. [LHZP07] and by Pobitzer et al. [PPF+11].

An additional category has been later added to this classification, i.e., *partition-based* visualization [SJWS08]. As the name implies, this class of approaches is built on the idea of partitioning the flow data prior to the visualization. The partitioning is typically carried out by means of clustering or topological analysis. More details on this category can be found in the work by Salzbrunn et al. [SJWS08].

In general, the flow visualization literature is rich of techniques that can satisfy many disparate needs. Notice that the overall goal of flow visualization approaches is to visually describe a flow phenomenon through interactive depictions of the data. With the continuous improvements of hardware computation capabilities, flow simulations become more and more precise and can tackle larger and large spatio-temporal domain. As a consequence, the amount of data that constitutes a flow dataset becomes increasingly larger. In this context, traditional flow visualization techniques run the risk of incurring cluttering and self occlusion, therefore providing an unsatisfactory description of the phenomenon of interest. The clarity and intelligibility of the visualizations presented to the user need to be taken into account explicitly. This specific kind of problems is a subject of studies within the field of illustrative visualization [RBGV08].

2 Illustrative Visualization

The field of illustrative visualization is concerned with the visualization of scientific data by taking inspiration from traditional handcrafted illustrations [RBGV08]. In fact, artists and illustrators are explicitly trained to communicate information through their artworks. The basic principles and guidelines followed by artists and illustrators can be also applied to the design of computer-generated visualizations. This strategy can lead to significant improvements in terms of expressiveness and visual quality.

Illustrative techniques can be subdivided in four categories, depending on which aspect of the visualization they address. *Perceptual enhancement* techniques aim at improving the visual properties of the generated images, such as depth or shape perception [VGH+05]. These techniques are also known as *non-photorealistic rendering* (NPR) algorithms, emphasizing the fact that they mainly affect the rendering stage of the visualization pipeline [SS02]. For instance, the use of hatching [WS96], stippling [LME+02] or toon shading [GGSC98] in place of the traditional Phong shading can make the shape of 3D objects easier to understand. The boundaries and the depth order between multiple objects can be emphasized by showing contours, silhouettes and halos [IFH+03].

Visibility management and *focus emphasis* approaches deal with issues of data visibility, such as cluttering and occlusion. The basic problem is that the visual resources provided by the target visualization device are insufficient for depicting the desired amount of data all at once. A straightforward solution is to dis-

play only a portion of the data (cutaway, selective visualization, etc.). In case this compromise is unacceptable, transparency and ghosting can be employed in order to reveal hidden portions of the data. Alternatively, the shape and the distribution of visual entities can be altered by applying specific deformation or explosion strategies [VG05]. The main difference between visibility management and focus emphasis is that the latter assumes that a portion of the data (the *focus*) is more important than the rest (the *context*). Therefore, visual resources should be allocated accordingly [Hau03, VKG05].

The fourth category is *visual explanation*. These techniques aim at providing a comprehensive description of the phenomenon of interest through guided visualization and visual storytelling. Visual explanation relies on domain knowledge in order to assign specific meanings to portions of the data. This piece of semantic information is then exploited for mapping the data to suitable visual entities, and to handle rendering and animations. Different levels of interaction can be provided. Fully automatic visualizations are useful for presentation purposes. In this context, the required domain knowledge have to be taken into account during the design and the development phases. Alternatively, semi-automatic and manual approaches let the users control some elements of the visualization. The users are often allowed to specify additional semantic information while exploring the data. For instance, they can identify and label specific features in the data.

The principles of illustrative visualization have been applied in several visualization subfields, such as medical visualization [VBŠ+13, AH11], geological visualization [LNP+13] and visualization of biological data [LMPSV14]. More details about illustrative visualization and its applications can be found in the tutorial by Viola et al. [VG05]. In the next section, we give an overview of illustrative approaches tailored for flow visualization.

3 Related Work

The amount of literature on the topic of illustrative flow visualization is considerable. Here, we discuss selected approaches which are closely related to the topic of this thesis. Specifically, we focus on techniques that address perceptual and visibility issues. For the sake of clarity, we divide our discussion in three parts. We first introduce approaches that deal with field data, such as pressure (a scalar field), velocity (a vector field) and strain rate (a tensor field). Then, we discuss the illustrative visualization of integral curves, from the seeding of individual lines to the management of large bundles of curves. Finally, we present visualization algorithms aimed at the expressive depiction of integral surfaces. The visualization of flow features is not directly addressed in this thesis. A comprehensive discussion about the illustrative flow visualization field can be found in Paper A.

3.1 Techniques addressing field data

Visualization of field data is typically carried out by glyphs, color coding and texture advection. All these strategies work pretty well in case the spatial domain is two-dimensional and only one or two fields have to be visualized. If any of these two conditions is false, specific visualization strategies need to be adopted.

The literature in glyph-based visualization is large, and we refer to the survey by Borgo et al. [BKC+13] for a general overview. In the context of flow visualization, the *flow probe* [dLvW93] is a notable example of glyph-based visualization design for multivariate flow data. The flow probe is a particular kind of glyph that can simultaneously represent up to 6 local properties of the flow. The downside is the high morphological complexity of the glyph. In order to avoid visibility issues, only a small number of probes should be placed in the spatial domain. The opposite strategy is adopted by Kirby et al. [KML99]. Each flow attribute is mapped to a simple type of glyph, such as arrows and ellipsoids, and the spatial domain is densely covered with glyphs. This technique is designed for bi-dimensional domains only. An extension to 3D would lead to intense cluttering and occlusion. A glyph based visualization for 3D flows is described by Bürger et al. [BKKW08]. They employ an importance measure and a clustering algorithm in order to control the density of the glyphs. They depict only one attribute, the velocity, using arrow glyphs and deformed particles. In general, an adequate trade-off between number of variables, visual density and domain dimensionality is necessary in order to design an effective glyph-based visualization of flow data.

Color coding is frequently adopted in combination with other visualization strategies in order to depict local scalar properties. For instance, it is common to color glyphs, textures or integral structures according to the magnitude of the velocity. The extension to 3D, i.e., volume rendering, can be adopted to visualize scalar fields, such as pressure, temperature and, notably, several flow feature detectors [HWM88, JH95, HK99]. Besides few exceptions [PBL+04, SJEG05], volume rendering for flow visualization has been scarcely investigated. However, the basic idea of ray casting is adopted by a number of texture-based and surface-based techniques.

Texture advection is mainly employed for depicting vectorial quantities. *Spot noise* [vW91] and *Line Integral Convolution* (LIC) [CL93] are considered the foundations of this category of approaches. Texture-based visualization is particularly effective for depicting vector field in 2D or over surfaces. The high visual density of this kind of visualization drastically reduces its effectiveness in 3D. In this context, the typical strategy is to focus on specific portions of the flow data. For instance, the physical experiment of injecting dye in the flow can be imitated by adding colored spots to the advected texture [SJM96, TvW03]. Ad hoc techniques can be adopted for controlling the appearance and the behavior of the dye [LTH08, KSW+12]. Another possibility to improve the readability of

the visualization is to modulate the transparency of the texture according to a user-defined importance function [WSE05].

Irrespective of the visualization strategy adopted, importance functions can be also employed to determine geometrical changes of the visual representations. For instance, Correa et al. [CSC07] cut and deform regions of low importance so that the focus regions are shown unobstructed. Sigg et al. [SFCP12] instead adopt an importance-based optimization process in order to determine the best locations for placing cutaways.

3.2 Techniques addressing integral curves

Integral structures are useful for investigating the long-term behavior of fluid flows. In general, integration-based visualizations have to deal with two important design choices: where to seed/place the integral structures and how to visualize them.

Visualization designs based on integral curves, in analogy with glyph-based ones, requires a trade-off between the amount of information per single entity and the overall visual density. Integral curves are commonly rendered as simple lines or as geometrical object, such as ribbons or tubes. In the former case, a large number of curves can be visualized, and illustrative techniques are often employed for improving depth and shape perception. In the latter case, the color, the shape and the texture of the visual representation can be modified according to several flow attributes. However, only a few lines should be depicted in order to avoid visibility issues. The *Stream Polygon* [SVL91] is one of the earliest approaches in the second category. An n-sided polygon is swept along an streamline and deformed according to local flow properties, such as strain rate and vorticity. By connecting the deformed polygons, a *streamtube* is generated. The appearance of a streamtube can be improved by means of, e.g., shading, texturing and limb darkening [SKH$^+$04, SGS05]. Illustrative techniques have been also applied to ribbon-like representation of integral curves [CYY$^+$11].

Concerning the visualization of dense line bundles, the *illuminated streamlines* [ZSH96] technique improves depth perception by applying Phong shading to infinitesimally thin curves. Additional perceptual enhancements can be obtained by employing, for example, halos [EBRI09] or ambient occlusion [EHS13]. Occlusion issues have been addressed by Günther et al. [GRT13], who assign transparency to line segments according to an optimization algorithm. The deformation-based approach by Correa et al. [CSC07], mentioned previously, can be applied also to dense line bundles. Geometrical transformations are adopted by Angelelli and Hauser [AH11] as well, who straighten integral curves within tubular structures (blood vessels, pipes, etc.) in order to ease their visual comparison.

A second class of techniques addresses the seeding and placement of integral curves in the spatial domain. Two subcategories can be identified: techniques for providing an overview of the whole dataset, and techniques focusing on specific

spatial locations. One of the most common strategies in the first category is clustering. The spatial domain is partitioned according to local flow properties, then one or more seeding curves are seeded from each partition [HWHJ99, TVW99]. Alternatively, integral curves can be densely seeded in the spatial domain, and then clustered. Each cluster can then be depicted by just a few representative lines [CYY+11]. Clustering approaches are often hierarchical, therefore the density of the visualization can be directly controlled by the user. On the other hand, the curves distribution strategy depends on the selected clustering algorithm and can hardly be adjusted by the user. A number of techniques address this issue. Jobard and Lefer [JL97] take into account the minimal distance between each pair of curves in order to produce evenly distributed streamlines. Visual improvements can be obtained by adopting a shape similarity measure instead of the Euclidean distance [CCK07]. By applying an analogous approach in screen space, uncluttered visualizations can be obtained also for 3D datasets [LS07].

Techniques for emphasizing specific locations in the domain are based on the common concept of focus+context visualization. In the approach by Fuhrmann and Gröller [FG98], relevant areas of the domain are densely covered with integral curves. Outside those areas, few streamlines are shown in order to provide contextual information. Hauser and Mlejnek [HM03] let the user specify focus regions by brushing over histograms and scatterplots. Streamlines are then seeded from the selected regions. Xu et al. [XLS10] also seed integral curves from important regions, but instead of adopting a user-defined importance measure, they compute the importance field automatically. Specifically, the relevance of a location is given by the directional entropy of the velocity field within a limited neighborhood.

3.3 Techniques addressing integral surfaces

Integral surfaces can convey more information than integral curves, but they also consume a larger amount of visual resources. If the flow exhibits a strong rotational or diverging behavior, integral surfaces can roll up and significantly grow in size, resulting in highly occluded visualizations. In general, visibility is a serious issue even when visualizing a single integral surface. Consequently, a considerable portion of research on this topic is focused on the visualization of a single integral surface. Similarly to integral curves, we can distinguish between rendering and seeding challenges. In the former category, one of the first work addressing visibility issues for integral surfaces is the *stream arrows* technique [LMGP97]. This approach consists in visualizing a stream surface with arrow-shaped cutaways distributed along its length. The cutaways reveal hidden portions of the surface and, at the same time, their shape conveys the direction of the flow. Transparency can also be employed for reducing visibility issues. Notice that traditional alpha blending becomes ineffective if more than two of the surface layers are overlapped. *Angle-based* and *normal-variation* transparency al-

gorithms are able to overcome this limitation [HGH+10]. Non-local transparency techniques have been proven effective as well [CFM+13]. Directional textures [HGH+10] and anisotropic diffusion [CSFP12] can be integrated for conveying directional information. Additional illustrative techniques can be also included [BWF+10] in order to produce comprehensive visualizations.

Early techniques for automatically seeding integral surfaces are mainly concerned with the properties of the seeding location. In other words, the integral surface is used as a tool to visualize certain aspects of the flow passing through the seeding location. Wiebel and Scheuermann [WS05] construct integral surfaces using selected points, called *eyelets*, as seeding structures. An eyelet is placed in a region of high variation of the velocity field, such as a vortex core of a saddle point. The resulting surface conveys the associated motion patterns, e.g., the swirling or the diverging behavior of the flow. In contrast, Theisel et al. [TWHS03], and Peikert and Sadlo [PS09] seed stream surfaces in the proximity of topological structures, such as critical points and separatrices. In this way, they are able to convey information about the topology of the velocity field. This concept has been extended to unsteady flows by Ferstl et al. [FBTW10]. They identify topological features (FTLE ridges) on two-dimensional planes manually placed by the user. By computing streak surfaces from these ridges, they can illustrate separation phenomena in the flow. In recent years, a different idea is being investigated: the seeding curve should be selected so that the resulting surface exhibits certain properties. In most cases, this idea is coupled with strategies for placing multiple surfaces across the domain. Edmunds et al. propose two strategies for densely covering the spatial domain with stream surfaces. In the first strategy [EML+11, ELC+12b], the seeding curves are given by the iso-lines of a scalar field defined on the domain boundaries. Therefore, the surfaces can depict the behavior of the incoming and outgoing flow. In the second approach [ELM+12] instead, they adopt the clustering strategy by Telea and van Wijk [TVW99]. By computing a seeding curve through the center of each cluster, they can guarantee a dense coverage of the spatial domain. The seeding curves are constructed by integrating a vector field orthogonal to both the velocity and the acceleration fields. In this way, the resulting sufaces are able to convey the local rotational behavior of the flow. These seeding strategies can be considered *local*, since the selection of seeding curves is driven only by local properties. In contrast, Martinez Esturo et al. [MESRT13] propose a *global* strategy. They first compute a large number of stream ribbons. An optimization process selects a subset of these ribbons in order to construct the single best stream surface in the dataset, where "best" refers to a combination of surface quality measures. This approach has been later extended [SMG+14] in order to compute a family of locally optimal stream surfaces.

Chapter 3

Theoretical background

In this chapter we give a brief overview on the theory of fluids' motion and flow analysis. The goal is to introduce the theoretical background our visualization designs are based on. We first discuss selected concepts from fluid dynamics, including the fundamental equations and the main quantities that describe the instantaneous motion of a fluid. These notions hold an important role in Papers B and D. Then, we introduce a popular tool-set for investigating the long term behaviour of fluid flows, i.e., integral structures. Integral curves and integral surfaces are described and formally defined. Papers C, D and E are based on these analysis tools.

1 The Motion of Fluids

Here we present selected concepts from fluid dynamics. Given the considerable breadth of this field, we discuss only those notions which are directly related to our work. Interested readers can refer to the exhaustive book by Aris [Ari89]. In the following, we employ the component notation for vectors and tensors. A vector $\mathbf{u} \in \mathbb{R}^n$ has n components $u_1, u_2, \ldots, u_n$ and is written as u_i (where the index i varies between 1 and n). A rank 2 tensor $\mathbf{S} \in \mathbb{R}^{m \times n}$ has $m \times n$ components and is written S_{ij}. Indices are typically denoted by the letters i, j and k. We limit our discussion to the Euclidean space E_3, therefore all the indices vary from 1 to 3, unless otherwise stated. We also adopt the Einstein's summation convention [Ein16]. Whenever an index appears twice in the same expression, summation over that index is implied. Relevant examples include:

- $u_i\, v_i = \sum_{i=1}^{3} u_i\, v_i$ is the dot product between the two vectors $\mathbf{u}$ and $\mathbf{v}$.
- S_{ii} is the trace of $\mathbf{S}$, i.e., the sum of the elements along its diagonal.
- $S_{ij}\, u_j$ is the typical matrix-vector multiplication $\mathbf{S}\,\mathbf{u}$, while $S_{ij}\, u_i = u_i\, S_{ij}$ corresponds to $\mathbf{u}^T\,\mathbf{S}$.
- In the case of additions (or subtractions), the rule does not apply. If u_i and v_i are vectors, $u_i \pm v_i$ is a vector whose components are given by the sum (or difference) of the corresponding components of $\mathbf{u}$ and $\mathbf{v}$.

The motion of a fluid is described by the Navier-Stokes equations [Ari89]. The momentum-conservation equation of Navier-Stokes is given by

$$\frac{Du_i}{Dt} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu\frac{\partial^2 u_i}{\partial x_j \partial x_j} + f_i. \tag{3.1}$$

The vector u_i is the velocity of the fluid. The left-hand side Du_i/Dt of the equation is the material derivative of the velocity. It represents the instantaneous rate of change in velocity of a fluid particle. On the right-hand side we have, in succession, the force due to variations of the pressure p (scaled by the inverse of the density $1/\rho$), the damping caused by the viscosity (ν denotes the kinematic viscosity), and the influence of external forces f_i (such as gravity).

The velocity is one of the most important descriptors of the motion of a fluid. It is given as vector field $u_i : E \times T \to \mathbb{R}^3$ defined over a spatial domain $E \subseteq \mathbb{R}^3$ and a temporal domain $T \subseteq \mathbb{R}$. With $u_i(x_j, t)$ we refer to the instantaneous velocity of the fluid particle located at position x_j at time t. For the sake of clarity, the parameters x_j and t are dropped whenever the meaning is clear. The spatial derivatives of u_i convey additional information about the fluid's motion at a given location. The *velocity gradient* $\mathbf{U} = \nabla\mathbf{u}$ is a 3×3 matrix whose components are given by $U_{ij} = \partial u_i/\partial x_j$. It defines the infinitesimal rate of change of the velocity in space. At any given location in the domain, an infinitesimal displacement dx_j leads to a change in velocity given by $du_i = U_{ij}\, dx_j$. The matrix U_{ij} is square, therefore it can be decomposed into a symmetric and an antisymmetric part, that is $U_{ij} = S_{ij} + \Omega_{ij}$. The symmetric part S_{ij} is the *strain rate tensor*, while Ω_{ij} is the antisymmetric *rotation tensor* (Fig. 3.1):

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \qquad \Omega_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right). \tag{3.2}$$

The rate of strain tensor describes the straining motion of an infinitesimal parcel of fluid (Fig. 3.1c, d). It can be easily interpreted by analyzing its spectral decomposition. As a matter of fact, S_{ij} is real and symmetric, therefore it has real eigenvalues and orthogonal eigenvectors. The eigenvectors of S_{ij} correspond to the principal directions of deformations (expansion or compression) of the infinitesimal parcel. The eigenvalues are proportional to the intensity of the deformations. Positive eigenvalues indicate expansion, while negative eigenvalues indicate compression. For incompressible fluids, the parcel's volume must remain unchanged. Therefore the sum of the eigenvalues (i.e., the trace S_{ii} of $\mathbf{S}$) is 0, indicating that expansion in certain directions is balanced by compression in other directions (shearing motion). The rotation tensor instead represent the rotational motion of the infinitesimal parcel (Fig. 3.1b). Since Ω_{ij} is antisymmetric, it can be mapped to a vector ω_i by the relation $\omega_i = -\epsilon_{ijk}\,\Omega_{jk}$, where ϵ_{ijk} is the permutation tensor [Ari89]. ω_i is known as the *vorticity vector*. On the other hand, the relation $\Omega_{ij} = -1/2\,\epsilon_{ijk}\,\omega_k$ allows us to calculate the rotation tensor

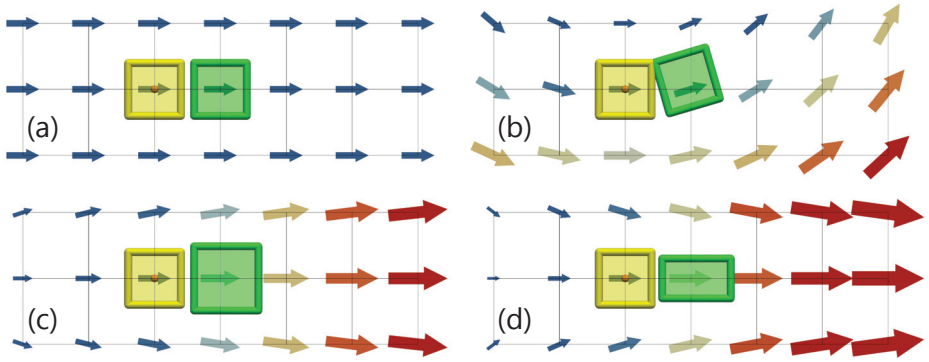


Figure 3.1: The yellow square depicts a small volume of fluid centered at the origin of the coordinate system (orange point). The green square represents the same volume advected for a short time in 4 different types of flow. (a) Laminar flow $u_0 = 1$, $u_1 = 0$. The parcel is only translated, vorticity and strain rate are zero. (b) Rotational flow $u_0 = 1 + 0.3y$, $u_1 = -0.3x$. The rotation is counter-clockwise so the vorticity is a vector pointing outside of this page. The vorticity magnitude is 0.3. There is no deformation, so the strain rate is 0. (c) Expanding flow $u_0 = 1 + 0.2x$, $u_1 = 0.2y$. There is no rotation, so the vorticity is 0. The eigenvectors of the strain rate are aligned with the axes of the coordinate system. The eigenvalues are both 0.2, indicating that the rate of expansion is the same in both directions. (d) Strain dominated flow $u_0 = 1 + 0.3x$, $u_1 = -0.3y$. The vorticity is zero. As in the previous case, the eigenvectors of the strain rate are aligned with the x- and y-axis, but the eigenvalues are respectively 0.3 and -0.3. Notice that (c) is the only case where the volume of the parcel changes (compressible flow).

from the vorticity vector. The direction of ω_i is the axis of rotation of the infinitesimal parcel, while its magnitude is proportional to its angular velocity.

While representing two different aspects of the fluid's motion, vorticity and strain rate are strongly related, and they affect each other's evolution. For instance, the vorticity transport equation for incompressible fluids is given by

$$\frac{D\omega_i}{Dt} = S_{ij}\omega_j + \nu\frac{\partial^2\omega_i}{\partial x_k \partial x_k} + \Gamma_i, \tag{3.3}$$

where Γ_i is the torque induced by the external forces f_i (Eq. 3.1). Besides the damping effect of viscosity, the strain rate transforms the vorticity by means of stretching, shearing and rotation. The interactions between strain rate and vorticity have a significant influence on the flow behavior of the fluid [MK85, VM91, NP98, Wil96]. They assume a particular importance in connection to relevant motion patterns, known as *coherent structures*. Coherent structures are portions of the fluid which present specific homogeneous characteristics. Typical

examples are vortices and shear layers. Vortices are associated with swirling motion patterns in the fluid, which are often characterized by intense vorticity. Several techniques for detecting vortices can be found in literature, e.g., Hunt's Q [HWM88], λ_2 [JH95] and the parallel vector operator [PR99]. Shear layers are instead characterized by prominent shear deformations and can be detected, for instance, with the approach by Haimes and Kenwright [HK99].

These concepts represent the basic knowledge that motivated the visualization designs in Papers B and D. In Paper D, flow attributes are represented in a reformed coordinate system. It is therefore important to investigate how these quantities behave under transformations of the frame of reference. Scalar quantities, such as the pressure or the magnitude of the velocity, are by definition invariant to this kind of transformations. In contrast, the components of vectors and tensors need to be transformed according to the rules of tensor algebra [LR89]. Assume there is a one-to-one bi-continuous mapping $\overline{x}_i = \overline{x}_i(x_j)$ from the points x_i in the original space to the points $\overline{x}_i$ in the reformed space. With $\partial\overline{x}_i/\partial x_j$ and $\partial x_i/\partial\overline{x}_j$ we denote the derivatives of the direct mapping $\overline{x}_i(x_j)$ and the derivative of the inverse mapping $x_i(\overline{x}_j)$, respectively. These derivatives allow us to determine the components of the velocity, vorticity and strain rate in the reformed space. By definition, the velocity represents the instantaneous rate of change of the position of a particle, i.e., $u_i = \partial x_i/\partial t$. The definition holds in any frame of reference, therefore $\overline{u}_i = \partial\overline{x}_i/\partial t$, where $\overline{u}_i$ are the components of the velocity in the reformed coordinate system. By applying the chain rule, we obtain

$$\overline{u}_i = \frac{\partial\overline{x}_i}{\partial t} = \frac{\partial\overline{x}_i}{\partial x_j}\,\frac{\partial x_j}{\partial t} = \frac{\partial\overline{x}_i}{\partial x_j}\,u_j, \tag{3.4}$$

which is the behavior of a *contravariant* vector under transformation. Accordingly, the velocity is a contravariant vector. The vorticity is a contravariant vector as well, while the strain rate is a mixed tensor [LR89] (the demonstrations are omitted since they require more advanced concepts from tensor algebra, which are beyond the scope of this thesis). Therefore, their components in the reformed space are given by

$$\overline{\omega}_i = \frac{\partial\overline{x}_i}{\partial x_j}\,\omega_j, \qquad \overline{S}_{ij} = \frac{\partial\overline{x}_i}{\partial x_h}\,\frac{\partial x_k}{\partial\overline{x}_j}\,S_{hk}. \tag{3.5}$$

Part of the work we present in Paper D is based on these equations. Clearly, the field of tensor algebra extends far beyond this brief introduction. More details can be found in the book by Lovelock and Rund [LR89].

2 The Long-term Flow Behavior

Velocity, vorticity and rate of strain are quantities defined point-wise in the spatial and temporal domain. As previously mentioned, $u_i(x_j, t)$ denotes the velocity

of the fluid particle located at the specific position x_j at the specific instant t. However, interesting aspects of flow phenomena can be revealed by observing the evolution of particles over time. In order to encode this kind of information, *integral structures* are normally adopted.

The set of one-dimensional integral structures, called *integral curves*, includes *streamlines*, *path lines*, *time lines* and *streak lines* [Hau06]. Given a *seeding point* $(\tilde{x}_i, \tilde{t})$ in the spatio-temporal domain, a streamline is a parametric curve $\boldsymbol{\psi}^s : \mathbb{R} \to \mathbb{R}^3$ which solves the following ordinary differential equation:

$$\frac{d\psi_i^s(\xi)}{d\xi} = u_i(\psi_j^s(\xi), \tilde{t}) \quad \text{with} \quad \psi_i^s(0) = \tilde{x}_i. \tag{3.6}$$

The superscript s indicates the type of integral curve, that is, a streamline. We use the superscripts p, k and t for referring to path, streak and time lines respectively. The analytical solution of Equation 3.6 is given by

$$\psi_i^s(\xi) = \tilde{x}_i + \int_0^{\xi} u_i(\psi_j^s(\tau), \tilde{t})\, d\tau \quad \text{with} \quad \psi_i^s(0) = \tilde{x}_i. \tag{3.7}$$

In practice, a streamline is a curve that begins at the seeding location $\tilde{x}_i$ and is everywhere tangent to the *instantaneous* velocity field at time $\tilde{t}$ (Fig. 3.2 left). The integral is usually computed with numerical integration methods, e.g., Euler's method or any technique in the Runge-Kutta category [But08]. Notice that ξ is just an integration parameter. It can be chosen so that it corresponds, e.g., at the arc length of the curve. It is not related to the time axis of the spatio-temporal domain. In fact, a streamline seeded at a time $\tilde{t}$ exists only at the specific time instant $\tilde{t}$.

If the variation of the velocity field over time is taken into account, a path line is produced. Specifically, a path line $\psi_i^p(t)$ is the solution of

$$\frac{d\psi_i^p(t)}{dt} = u_i(\psi_j^p(t), t) \quad \text{with} \quad \psi_i^p(\tilde{t}) = \tilde{x}_i. \tag{3.8}$$

which is given by

$$\psi_i^p(t) = \tilde{x}_i + \int_{\tilde{t}}^{t} u_j(\psi_i^p(\tau), \tau)\, d\tau \quad \text{with} \quad \psi_i^p(\tilde{t}) = \tilde{x}_i. \tag{3.9}$$

A path line depicts the trajectory of a particle released in the flow at the seeding point $(\tilde{x}_i, \tilde{t})$ (Fig. 3.2 right). In this case, the integration parameter t corresponds to the temporal axis of the domain. While a streamline lies in a single 3D "slice" of the 4D spatio-temporal domain (the slice at time $\tilde{t}$), a path line advances also in the temporal dimension as time passes.

Streak lines are fundamentally different from streamlines and path lines. The integration process required for constructing a path line represents the evolution

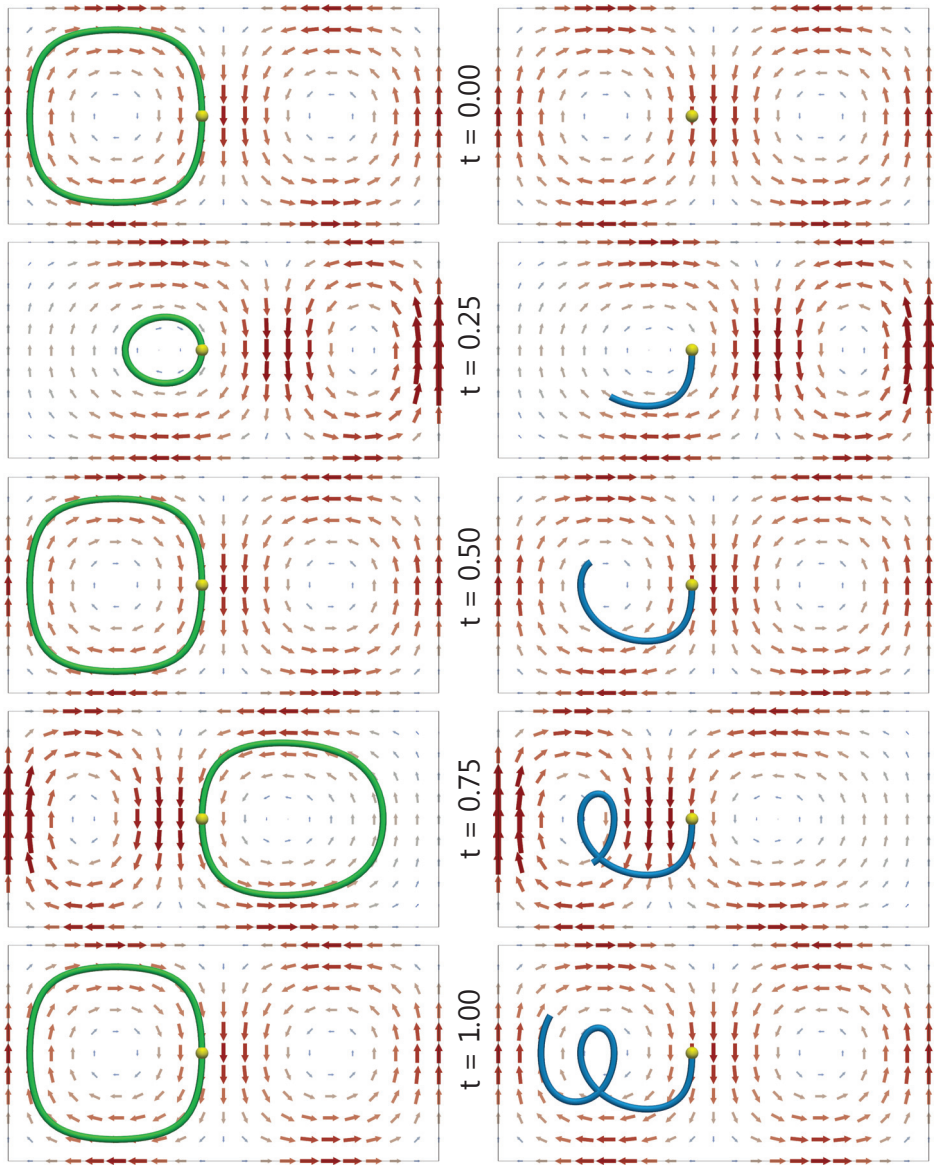


Figure 3.2: Five Streamlines (left) and one path line (right) in the double gyre dataset [SLM05]. The seeding location is in yellow. The value of t refers to the seeding time for the streamlines, while it refers to the integration time for the path line. The path line is seeded at time $t = 0$. The arrow glyphs depict the instantaneous velocity. They are scaled and colored according to velocity magnitude.

of a single particle through the flow. An analogous statement can be made for streamlines, with the difference that only an instantaneous flow is considered. In contrast, the construction of a streak line is based on the simultaneous advection of multiple fluid particles emerging from a fixed point in space as time evolves. Intuitively, a streak line is the imaginary curve connecting the particles that were released at the seeding location at consecutive time steps (Fig. 3.3 left). This is analogous to the physical experiment of continuously releasing colored dye into a moving fluid. In order to formally define a streak line, we need to extend the previously introduced notation. We denote with $\psi_i[\tilde{x}_j, \tilde{t}]$ an integral line with seeding point $(\tilde{x}_i, \tilde{t})$. Consider the function $\boldsymbol{\psi}^k : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^3$ such that

$$\psi_i^k[\tilde{x}_j, \tilde{t}](r, t) = \psi_i^p[\tilde{x}_j, r](t) \quad \text{with} \quad \tilde{t} \leq r \leq t. \tag{3.10}$$

Here, $\psi_i^p[\tilde{x}_j, r]$ is the path line seeded at $\tilde{x}_i$ at time r (Eq. 3.9). In practice, $\psi_i^k[\tilde{x}_j, \tilde{t}](r, t)$ denotes the position at time t of a particle released in the flow at location $\tilde{x}_i$ at time r. The streak line at a certain time τ is obtained by fixing $t = \tau$ and letting r vary between $\tilde{t}$ and t. With the term *family of streak lines* we refer to the set of streak lines obtained by setting specific values of t. Parameter t defines which streak line in the family we are interested in, while parameter r identifies the points along a specific streak line.

Similarly to streak lines, time lines are also constructed by advecting multiple particles through the flow over time. A time line is obtained by simultaneously releasing a set of particles along a *seeding curve* (Fig. 3.3 right). Any curve within the spatial domain can be adopted as a seeding curve. Without loss of generality, let us assume that the seeding curve is parameterizable, i.e., it can be expressed as $\gamma_i : [0, 1] \to \mathbb{R}^3$. A *family of time lines* is described by a function $\boldsymbol{\psi}^t : [0, 1] \times \mathbb{R} \to \mathbb{R}^3$ defined as

$$\psi_i^t[\gamma_j, \tilde{t}](q, t) = \psi_i^p[\gamma_j(q), \tilde{t}](t). \tag{3.11}$$

In other words, $\psi_i^t[\gamma_j, \tilde{t}](q, t)$ denotes the position at time t of a particle released in the flow at location $\gamma_i(q)$ at time $\tilde{t}$. The time line at time τ is obtained by setting $t = \tau$ and letting q vary between 0 and 1.

By increasing the dimensionality of the seeding structure, the dimensionality of the resulting integral structure increases accordingly. *Stream surfaces* and *path surfaces* can be obtained by densely seeding streamlines or path lines (respectively) along a seeding curve. A stream surface starts at the seeding curve and is everywhere tangent to the instantaneous vector field. A path surface instead depicts the trajectories of the particles seeded along the seeding curve. Formally, the stream and the path surfaces seeded from the curve γ_i at time $\tilde{t}$ are given by

$$\Psi_i^s[\gamma_j, \tilde{t}](q, \xi) = \gamma_i(q) + \int_0^{\xi} u_i(\Psi_j^s(q, \tau), \tilde{t})\, d\tau \quad \text{with} \quad \Psi_i^s(q, 0) = \gamma_i(q). \tag{3.12}$$

$$\Psi_i^p[\gamma_j, \tilde{t}](q, t) = \gamma_i(q) + \int_{\tilde{t}}^{t} u_i(\Psi_j^p(q, \tau), \tau)\, d\tau \quad \text{with} \quad \Psi_i^p(q, \tilde{t}) = \gamma_i(q). \tag{3.13}$$

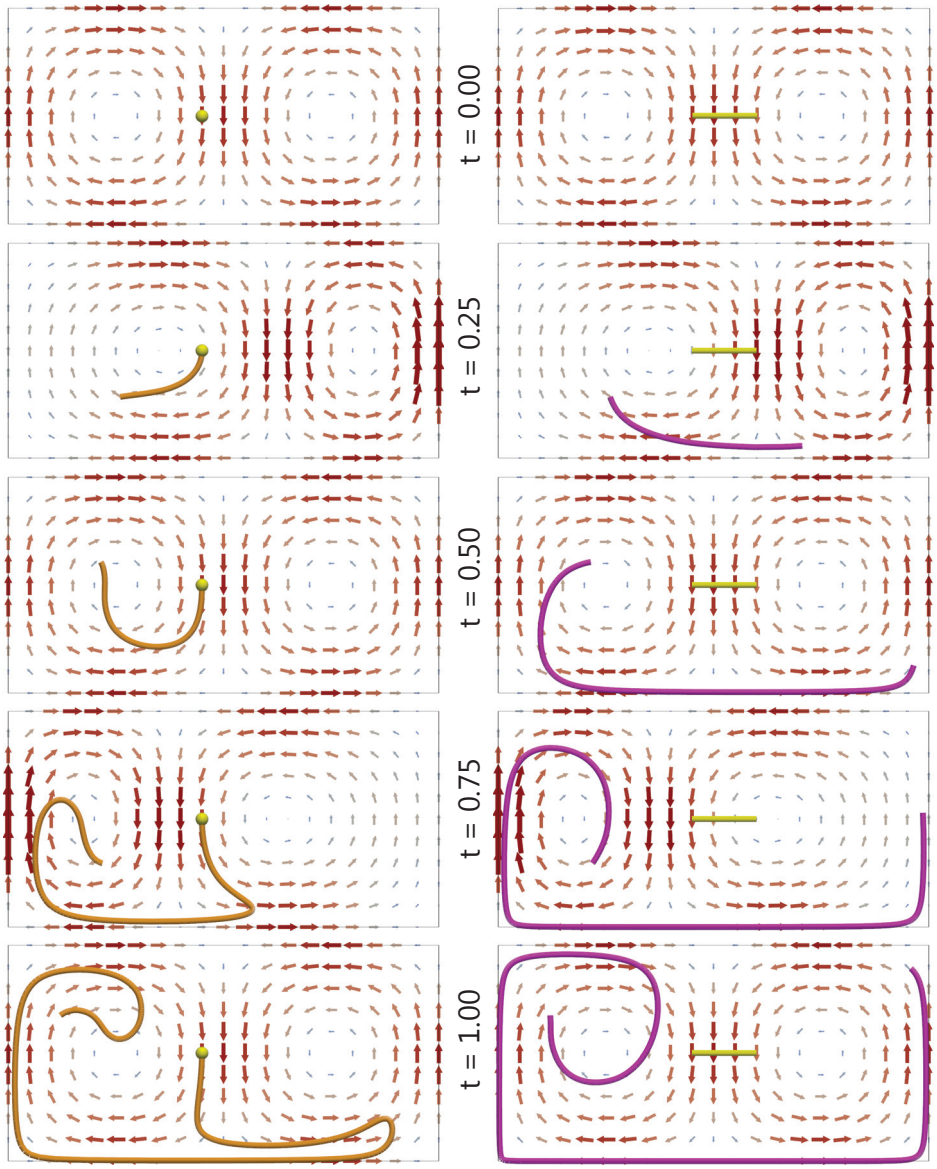


Figure 3.3: Same as Figure 3.2, but with streak lines (left) and time lines (right). The yellow line segment on the right is the seeding curve for the family of time lines. Both streak lines and time lines are seeded at time $t = 0$.

Equation 3.13 can be interpreted both as a family of path lines and as a family of time lines. By setting q to a specific value between 0 and 1, the resulting parametric curve corresponds to one of the path lines in the path surface. Conversely, if t is set to a specific value, the resulting parametric curve is a time line. Similar considerations are valid for Equation 3.12, and the previous arguments are valid for stream surfaces as well. The only difference is that fixing the value of t in Equation 3.12 leads to a time line of the *instantaneous* flow at the seeding time $\tilde{t}$.

A *streak surface* is obtained by continuously seeding particles in the flow along a seeding curve. A family of streak surfaces is defined as:

$$\Psi_i^k[\gamma_j, \tilde{t}](q, r, t) = \Psi_i^p[\gamma_j, r](q, t) \quad \text{with} \quad \tilde{t} \leq r \leq t. \tag{3.14}$$

The streak surface at time τ is the parametric surface obtained by fixing $t = \tau$. By fixing also the value of q, the resulting curve represents one of the streak lines in the streak surface.

In order to construct a *time surface* we require a bi-dimensional seeding structure, i.e., a *seeding surface*. Assuming that the seeding surface allows a 2D parameterization, we can express it as a function $\Gamma_i : [0, 1]^2 \rightarrow \mathbb{R}^3$. Then, a family of time surfaces is expressed by:

$$\Psi_i^t[\Gamma_j, \tilde{t}](p, q, t) = \psi_i^p[\Gamma_j(p, q), \tilde{t}](t). \tag{3.15}$$

Once again, setting $t = \tau$ yields the time surface at time τ.

This entire discussion can be extended to *integral volumes* as well. Integral volumes have been investigated in the context of flow visualization [MBC93, XZC04], but they never attracted considerable attention.

Within this thesis, Paper C and E deal with stream surfaces, while Paper D is mainly concerned with families of time surfaces. Most of the concepts presented in these papers can be potentially extended to other types of integral structures.

In this section we have introduced only the concepts required for understanding our work. A formal treatment of integral curves and surfaces from a mathematical point of view can be found in the book by Zachmanoglou and Thoe [ZT86]. More details about the use of integral structures in visualization can be found in the surveys by McLoughlin et al. [MLP+10] and Edmunds et al. [ELC+12a].

Chapter 4

Contribution

This chapter describes our proposed flow visualization designs. All of them share the property of explicitly addressing visibility issues early in the design process. Visibility is considered a global property of the entire visualization solution. In the following sections, we describe how we applied this principle to three characteristic flow application scenarios. We first present a glyph-based visualization approach that enables the simultaneous investigation of multiple aspects of the flow behavior. Then, we describe our strategies for the visual inspection and comparison of either single or multiple integral surfaces. And finally, we describe a novel method for seeding a set of stream surfaces at a user-specified location of interest within the spatial domain. The last section of this chapter provides a brief summary of our contribution.

1 Multi-aspect Visualization of the Motion of a Fluid

Research question

The motion of a fluid is a complex phenomenon and it is affected by several factors. The formation of relevant motion patterns is determined by the interaction of multiple aspects of the flow behavior, such as swirling motion, shearing, compression and expansion. Each aspect is in turn dependent on several flow attributes, including pressure, temperature, velocity, and others.

Typical analysis techniques normally take into account only a single flow aspect at a time. In the context of fluid dynamics, for example, scalar quantities are often investigated individually, by observing their statistical distributions and properties. The study of vectorial and tensorial quantities is traditionally carried out through the examination of the individual components, one at a time. For instance, the study of turbulent channel flows by Moin and Kim [MK85] includes observations on the correlation between the components of the velocity over a plane parallel to the bottom wall. Vincent and Meneguzzi [VM91] plot the evolution of kinetic energy over time and examine the distributions of both the streamwise velocity and its derivatives. In recent years, this investigation strategy is adopted mainly for the study of turbulence phenomena [JHSM10, SMM11]

Visualization techniques allow us to obtain a more comprehensive view of the various aspects of the fluid's motion. Scalar attributes can be depicted, for instance, through color coding 2D surfaces, or by volume rendering [SET+06]. Texture advection [CL93, HFH+04] and glyphs [Hab90, dLvW93] can be adopted for visualizing vectors and tensors. Integral curves enables the study of particle trajectories, and feature detectors can recognize specific flow patterns. However, the focus is still on a single flow aspect in most cases.

The overall motion of the fluid is the result of the continuous interactions between multiple flow aspects. The ability to observe and study them simultaneously is therefore of primary importance. One may be tempted to visualize all the available data with the goal of depicting all the different aspects of the flow. Such an approach would eventually fail, because cluttering and occlusion would make the visualization impossible to understand. In order to enable the concurrent investigation of multiple aspects of the fluid's motion, specifically designed visualization strategies are necessary.

Proposed approach

We propose a visualization design for the simultaneous visualization of multiple flow aspects. Typical examples includes rotational and straining behaviors, which can lead to the formation of coherent structures. As discussed in the previous chapter, velocity, vorticity and strain rate are strongly related to the fluid's motion and can effectively convey different aspects of the flow behavior (Ch. 3, Sec. 1). Therefore, we select them as representative attributes for illustrating selected aspects of the motion of a fluid.

The flow attributes of interest need to be mapped to appropriate visual entities. The velocity and the vorticity are given as vector fields, while the strain rate as a tensor field. Consequently, we require visual entities that can encode vectorial

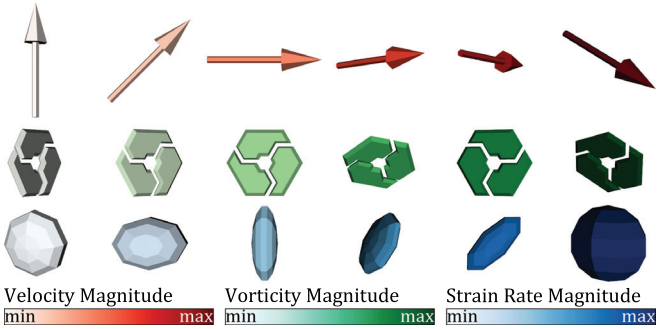


Figure 4.1: The glyphs adopted as visual entities for velocity (top), vorticity (middle) and strain rate (bottom).

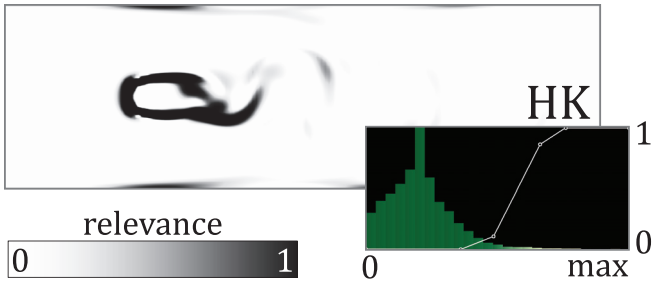


Figure 4.2: The relevance of a flow attribute specified as a function of the values of Haimes and Kenwright's shear layers detector [HK99]. In this dataset, the fluid flows from left to right and hits a square cylinder in the center (see Paper B).

and tensorial quantities. Our goal is to visualize the *local* interactions between multiple flow aspects, in order to understand their impact on the formation of characteristic motion patterns. It is therefore important to choose visual entities which are able to convey local flow properties at specific positions across the spatial domain. These requirements steered our design towards a glyph-based representation of the data. The glyph design is based on the *Design Guidelines* by Borgo et al. [BKC+13]. Our placement strategy may lead to locally dense glyph distributions. In such situations, geometrically complex glyphs, such as the flow probe [dLvW93], would be inappropriate. Therefore, we opted for glyphs of simple shape (D.G. 2 in [BKC+13]), which can be easily understood even when densely packed next to each other. In order to make the visualization as intuitive as possible, we adopt glyphs whose shape recalls the semantic of the associated attribute (D.G. 10 in [BKC+13]). Specifically, we map the velocity to the typical arrow glyph (Fig. 4.1 top). For the vorticity, we designed an ad hoc glyph which shows both the plane and the sense of rotation (Fig. 4.1 middle). The rate of strain is represented by a spherical glyph, anisotropically scaled along the eigenvectors of the strain rate tensor, proportionally to its eigenvalues (Fig. 4.1 bottom). The glyphs are colored according to the magnitude of the related quantity, adopting the common L_2 norm for vectors and the Frobenius norm for tensors. The color scales, selected from the ColorBrewer tool [HB03], are different for each of the attributes of interest, so that the glyphs can be easily distinguished even when densely packed (Fig. 4.4).

A fundamental element of glyph-based visualizations is the placement strategy. In fact, densely distributing glyphs across the domain can result in a highly cluttered scene. Additionally, we have to show multiple kinds of glyphs. Placing different glyphs at the same location can increase the overall degree of clutter-ing and introduce occlusion problems. We propose to address visibility issues in two ways: relevance-based transparency and coherency-based pruning. The

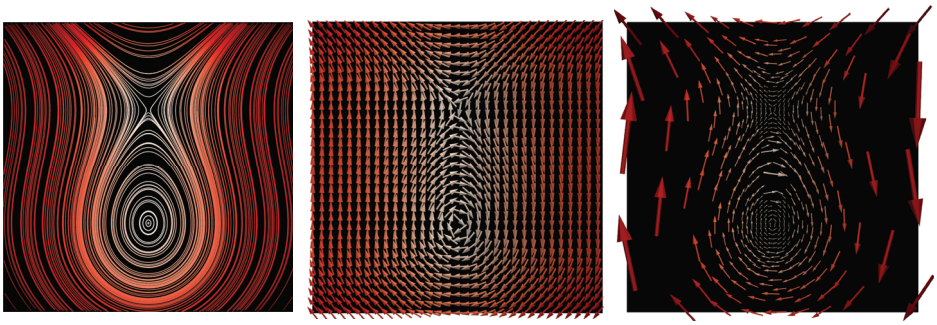


Figure 4.3: Streamlines in a synthetic flow dataset (left). Arrow glyphs placed according to a regular grid (middle). Arrow glyphs distribution resulting from our coherency-based strategy (right).

relevance-based transparency approach follows the basic notion of focus+context visualization. The opacity of each glyph is directly proportional to the importance of the data depicted by the glyph. While most focus+context approaches adopt a single importance measure, we evaluate the importance of the data separately for each attribute. In this way we are able to establish an association between the selected flow attributes and various aspects of the flow behavior. For instance, significant shear deformations can lead to the formation of shear layers. Therefore, visualizing the strain rate in correspondence of these structures can convey useful information about their formation and their evolution. We let the user define the relevance of each of the attributes of interest through a transfer function widget (Fig. 4.2). By employing appropriate feature detectors as the domain of such transfer functions, the user can emphasize specific aspects of the flow. For instance, the Q and the λ_2 vortex detectors [HWM88, JH95] can be used to detect regions of swirling motion. The shear layer detector by Haimes and Kenwright [HK99] can highlight areas dominated by shear deformations. In this way, the visual entities for velocity, vorticity and rate of strain are displayed only where the related attribute is considered important for conveying specific flow aspects. Conversely, unimportant data is suppressed in order to spare visual resources.

The relevance-based transparency results in an effective management of visual resources. However, regions characterized by relevant flow patterns can still exhibit cluttering and occlusion (Fig. 4.4a). Our second strategy for addressing visibility issues exploits data redundancy. The basic idea is that data samples which are spatially close and have similar (*coherent*) values, actually convey the same piece of information. In these situations, it is unnecessary to display a glyph for each data sample, the same information can be conveyed by showing one glyph only. Coherency is actually the foundation or several visualization

Figure 4.4: Multi-aspect visualization of a fluid flow around a square cylinder (see Paper B). (top) Only the relevance-based transparency is enabled. The resulting visualization is still cluttered. (bottom) The coherency-based pruning is able to effectively address the cluttering issue, resulting in a clear and expressive visualization. A demonstration of our technique on 3D scenarios can be found in Ch. 5, Sec. 2.

techniques, including clustering [HWHJ99, TVW99] and vector field topology [HH91]. We introduce the concept of *coherency measure*, i.e., a function that expresses the degree of coherency of a set of data samples with respect to a certain flow attribute. For each of the attributes of interest, we look for compact subsets of the spatial domain with a degree of coherency above a user-specified threshold. Each of these regions is visualized by a single glyph, placed at its center (Fig. 4.3). By varying the coherency threshold, the user can directly control the density of the glyphs. Recall that the ability to show localized information is one of the requirements. Therefore, we consider only regions of spherical shape, and we map the radius of a region to the corresponding glyph's size. In this way, the location and extent of an homogeneous region can be easily inferred by the position and scale of the glyph (Fig. 4.4b).

In our system, we included different coherency measures, which are able to evaluate the coherency of an attribute in terms of its magnitude, its orientation, or both. Details about the proposed coherency measures, and their impact on the resulting visualization, are discussed in Paper B. In order to improve the performance, we exploit the fact that there is no need for the coherency-based pruning in unimportant regions. Based on this, we developed an algorithm which handles both the relevance-based transparency and the coherency-based pruning simultaneously. This strategy allow us to achieve interactive execution times even on a single CPU core.

2 Exploration Strategies for Integral Surfaces

Research question

Integration-based flow visualization is a well established approach for investigating the long-term behavior of fluid flows. For bi-dimensional flows, integral curves are highly effective in conveying the most important trajectories and patterns of the fluid's motion. In the case that the spatial domain is three-dimensional, integral surfaces are also a viable option. Compared to curves, integral surfaces are generally more effective in conveying depth order and morphological information. Additionally, they can be employed themselves as a visualization space for other techniques, such as glyph-based visualization [LMGP97] or texture advection [LGSH06]. On the other hand, integral surfaces can become highly convoluted and significantly deformed in the presence of complex flow patterns. For instance, whenever a stream or a path surface traverses an area of strong swirling motion, such as eddies and vortices, it can easily fold, twist and roll up, therefore concealing information due to self occlusion. In such situations, extracting information from an integral surface can become a slow and cumbersome process, which requires extensive user interaction.

The visual inspection of an integral surface can be a challenging problem by

itself. There is however a second issue related to this first one. An entire flow dataset can be hardly characterized by a single integral surface in a comprehensive way. One possibility is to seed multiple integral surfaces throughout the spatial and temporal domain. Each of the surfaces can be studied individually, but it is often more interesting to investigate the differences between the various surfaces. Another similar situation occurs when dealing with ensembles of flow simulations. An integral surface can be seeded at the same location in each of the simulations of the ensemble. Then, the focus is on identifying how the surfaces differ from each other.

Effective comparative visualizations can be obtained, for example, by side-by-side visualization or object-space juxtaposition [PP95]. However, such an approach would lead to a conflict in the allocation of visual resources. In order to extract information from a single convoluted integral surface, the user needs to explore multiple points of view. One or more clipping planes are often employed, and shading and transparency algorithms have to be finely tuned. In contrast, the comparison is maximally effective if the visual representations of the compared objects have the same configuration. Finding a suitable balance between the ease of comparison and the amount of information conveyed is a crucial design challenge. Notice also that the difficulty of the problem increases if, instead of single integral surfaces, entire families of surfaces have to be compared.

Proposed approach: Visual investigation by surface splitting

We first deal with the problem of the visual investigation of an individual integral surface. We take stream surfaces as the main subject of our study, but the results can be directly extended to other kinds of integral surfaces. As a matter of fact, our technique can be adapted for handling any surface that allows for a meaningful 2D parameterization.

Our visualization design is based on the idea of splitting the surface into a number of pieces. This concept is rather common in the context of illustrative visualization. It has been applied, e.g., for visualizing mathematical surfaces through exploded views [KLMA10], and for the exploration of assemblies [LACS08] and hierarchical medical data [BVG10]. We propose an abstract framework which generalizes the visualization schemes based on splitting. Our approach builds on two key concepts: the *cut space* and the *surface complexity measure*. The cut space is the set of the admissible *cutting curves* (or simply *cuts*). It defines all the possible ways the surface of interest can be split. For instance, if we wish to split the surface using parallel planes (as in Karpenko et al. [KLMA10]) the cut space would include the curves obtained by intersecting the surface with any of the admissible cutting planes. The complexity measure instead is a scalar field defined over the surface. Since our goal is to address visibility issues, the complexity measure has to reflect the degree of occlusion induced by each of the points of the surface. Consider the example of two concentric spheres, observed

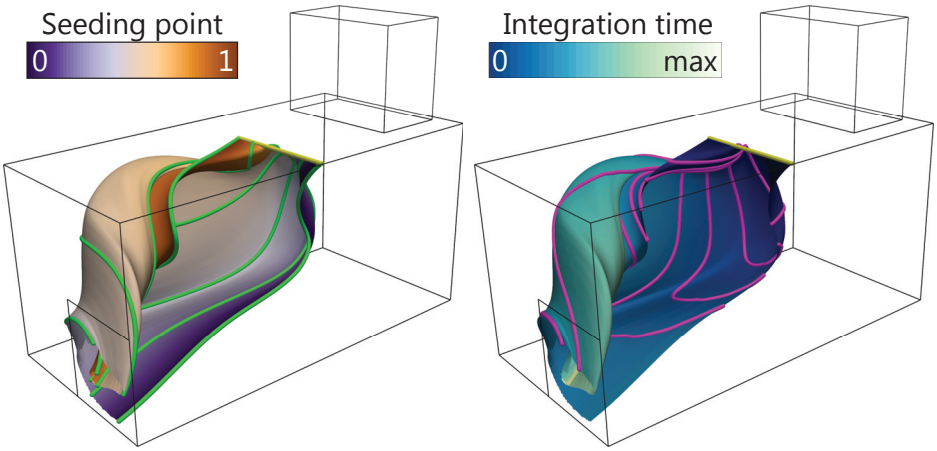
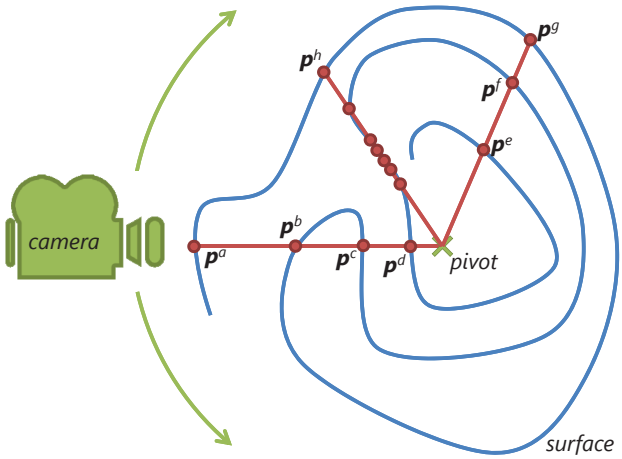


Figure 4.5: Selected streamlines (left) and time lines (right) within a stream surface. The seeding curve is in yellow. The dataset represent the flow of a fluid through a box (Ch. 5, Sec. 1). The fluid flows from the inlet at the top right towards the outlet at the bottom left.

from the outside of the larger sphere. Intuitively, the points of the inner sphere cause a lower degree of occlusion than the points of the outer sphere. Once the cut space and the complexity measure have been defined, the basic idea is to split the surface along the cutting curve which exhibits the highest overall complexity.

In the case of stream surfaces, defining the cut space is challenging. Let us consider the example of assembled objects. In this case, the cut space is given by the contact regions between the elementary components [LACS08]. Analogously, organs represent the building blocks of medical volumetric data [BG06, BVG10]. Integral surfaces are instead continuous objects and there is no clear and intuitive way to express them as an assembly of smaller basic units. Splitting the surface using fixed geometries, such as parallel planes, would effectively reduce visibility issues. However, the individual pieces may be of difficult interpretation since no information about the original flow field is associated with the cutting curves.

In order to define a proper cut space, we take into consideration the parametric definition of a stream surface (Ch. 3, Eq. 3.12). In fact, assuming that the flow is steady, a stream surface can be considered both as a collection of stream lines and as a collection of time lines (Fig. 4.5). We define the cut space as the set of streamlines and time lines in the stream surface. In this way, the interpretation of the resulting visualization would be clear and intuitive. In fact, by splitting the surface along a time line, each point of the cutting curve would have the same integration time. The points in the "first" piece of the surface would have an integration time lower than those along the selected cut. The points in the



point	cpx
p^a	3
p^b	2
p^c	1
p^d	0
p^e	0
p^f	1
p^g	2
p^h	see text

Figure 4.6: Illustration of our complexity measure. The degree of occlusion induced by a point depends on the number of intersection between the surface (in blue) and the line (in red) that connects the point to the pivoting point of the camera (in green).

other piece would have a higher integration time. The situation is analogous in the case the surface is split along a streamline.

Concerning the complexity measure, we have to define a proper way to quantify the degree of occlusion induced by each point of the surface. An exact evaluation of this quantity would require the examination of every possible viewpoint. However, the computational cost of such an approach would be too high for enabling interactive performance, therefore an approximation is necessary. Notice that the large majority of visualization tools adopts a *polar view* for rendering the objects of interest. In practice, a virtual camera rotates around a pivoting point, which is usually placed at the center of the displayed object. Based on this assumption, we evaluate the degree of occlusion of a point when it is at the center of the rendered scene, i.e., when it lies between the camera and the pivoting point (Fig. 4.6). We compute the complexity at a given point as the number of intersections between the surface and the line that connects such point to the pivoting point. In order to avoid degenerate situations where the line and the surface are parallel (Fig. 4.6, point $\mathbf{p}^h$), the contribution of each intersection is scaled by the dot product between the line's direction and the normal of the surface.

Once the complexity field has been computed, we determine the average complexity along the curves in the cut space. Ideally, we would like to split the surface so that both the resulting pieces exhibit a lower degree of occlusion. If the area of the two pieces is significantly uneven, it is likely that the larger one is still highly occluded. In order to take this global aspect into account, we scale

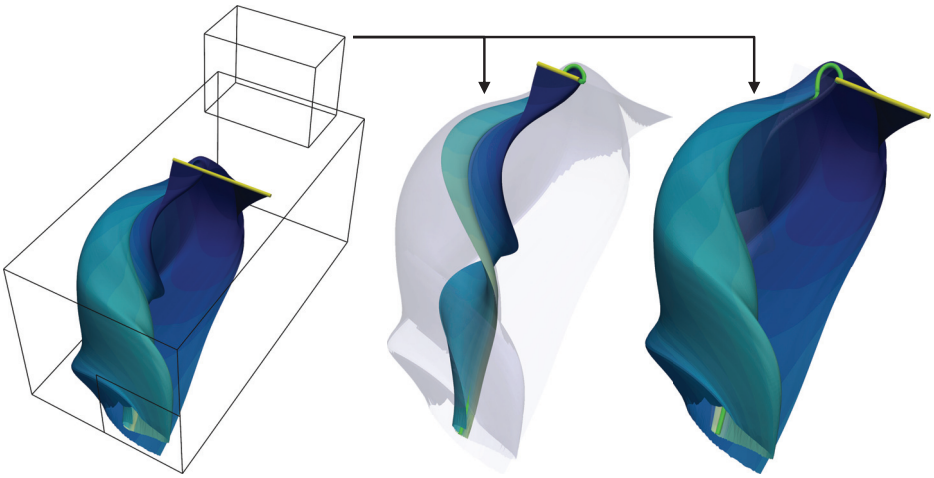


Figure 4.7: The result of our approach on the stream surface from Figure 4.5. The cut is performed along a streamline.

the average complexity of each cutting curve by the ratio between the areas of the two resulting pieces. At this point, we select the cutting curve with the highest scaled complexity value and we perform the cut (Fig. 4.7). Each of the resulting surfaces is again a stream surface with the same parameterization with respect to streamlines and time lines. Therefore, the whole process can be repeated in order to split them further.

Our algorithms runs at interactive rates. More details about our technique and its implementation can be found in Paper C, together with a summary of the execution times.

Proposed approach: Visual comparison of reformed surfaces

Our second approach for the investigation of integral surfaces is in the context of comparative visualization. Specifically, we propose a novel visualization design for the visual comparison of time surfaces. Nevertheless, our solution can be directly applied to other kinds of integral surfaces.

The effectiveness of a visual comparison is dependent on the user's ability to mentally relate the objects under examination. Therefore, the available visual resources should be exploited in order to make such a mental association easier. On the other hand, time surfaces can be highly convoluted and self-occluded. Addressing these issues requires again the investment of visual resources. Here, we propose to embed the time surfaces in a bi-dimensional space by means of a planar reformation algorithm. In this way, we are able to reduce the geometri-

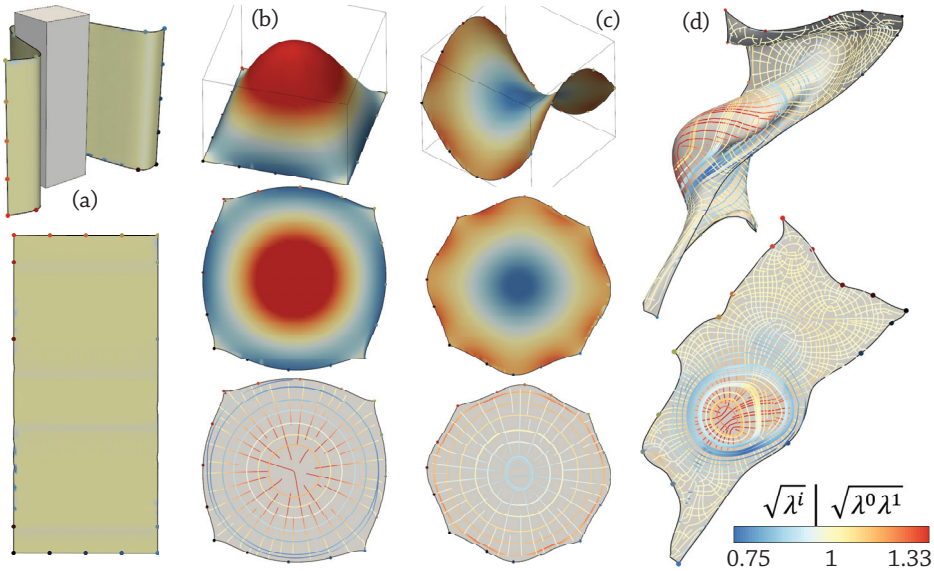


Figure 4.8: Examples of surface reformation. First row: original 3D surfaces. Second and third rows: reformed surfaces with visualizations of the metric tensor field via color coding or tensor lines (see text).

cal complexity of the surfaces, and simultaneously ease their visual comparison. Additionally, embedding the surfaces in a 2D space frees up the third dimension, which can consequently be exploited for comparative purposes [GAW+11].

Since the shape of a time surface carries important information about the underlying flow behavior, we adopt a modified version of the reformation technique by Liu et al. [LZX+08]. The embedding is computed by means of an iterative optimization process, which explicitly aims at minimizing the amount of distortion introduced (see Paper D for more details).

In order to support the visual comparison of time surfaces, we introduce four visualization techniques specifically designed to take advantage of the surface reformation (Fig. 4.8, 4.9, 4.10). First of all, we support the comparison of the original geometries by visualizing the *metric tensor field* of the 2D embedding through color coding and tensor lines (Fig. 4.8). In fact, the reformation process defines a bijective map $\overline{x}_i : \varphi \to \overline{\varphi}$ from the original 3D surface φ to its planar reformation $\overline{\varphi}$. Given an arbitrary point $x_i \in \varphi$, the corresponding point on the reformed surface $\overline{\varphi}$ is $\overline{x}_i = \overline{x}_i(x_j)$. Since the map is bijective, we also have $x_i = x_i(\overline{x}_j)$. At any point on $\overline{\varphi}$, the metric tensor [LR89] is defined as

$$\overline{g}_{ij} = \frac{\partial x_k}{\partial \overline{x}_i} \frac{\partial x_k}{\partial \overline{x}_j}. \tag{4.1}$$

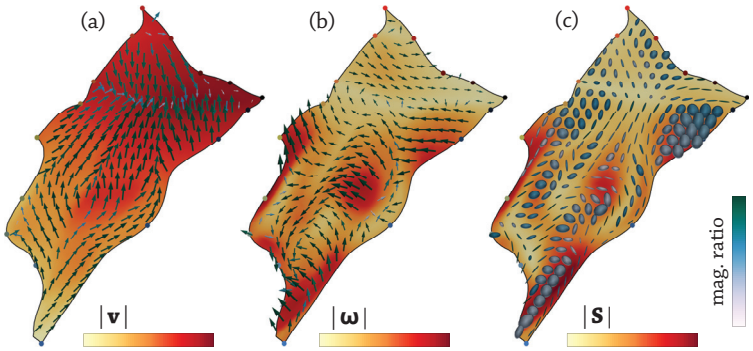


Figure 4.9: Glyph based visualization of the velocity, the vorticity and the rate of strain over a reformed surface. The surface is colored according to the magnitude of the related quantity. The glyphs are colored according to the relative orientation of the vectorial/tensorial quantity with respect to the surface.

It expresses the magnitudes and the directions of the deformation induced by the reformation. Notice that $\overline{g}_{ij}$ is symmetric and positive definite, therefore its spectral decomposition produces real eigenvalues λ^0, λ^1 and orthogonal eigenvectors e_i^0, e_i^1. The deformation in the direction of one of the eigenvectors is proportional to the square root of the corresponding eigenvalue. Let $dA \subset \varphi$ be an infinitesimal area on the 3D surface, and $d\overline{A} \subset \overline{\varphi}$ the corresponding area in the reformed space. We have $dA = d\overline{A}\,\sqrt{\lambda^0\,\lambda^1}$. Therefore, by color coding the surface according to the quantity $\sqrt{\lambda^0\,\lambda^1}$ we get an idea of the deformation introduced by the planar reformation (Fig. 4.8). Additionally, by visualizing the tensor lines of $\overline{g}_{ij}$, colored according to the square root of the related eigenvalues, we convey both the magnitude and the principal directions of the deformation.

Our second visualization solution enables the comparison of flow attributes over the reformed surfaces (Fig. 4.9). Scalar properties, such as temperature and pressure, are invariant under change of coordinate system, therefore they can be directly visualized on the reformed surfaces via, e.g., color coding. In contrast, vectors and tensors need to be transformed in order to correctly preserve their relative orientation in the reformed space. For the same reasons discussed in Section 1, we focus on velocity, vorticity and rate of strain, but our approach can be applied to any other scalar, vectorial and tensorial quantity. The rule for transforming these quantities have already been discussed in the previous chapter (Ch. 3, Sec. 1). Figure 4.9 shows a glyph-based visualization of the velocity, the vorticity and the strain rate fields over a reformed time surface. Notice that, because of the reformation process, the "off-surface" (normal) component of these quantities is lost. However, we can still convey this information through the color of the glyphs. Taking the velocity as an example, we set the glyphs'

color according to the ratio $||u_i^{tan}||/||u_i||$, where u_i^{tan} is the projection of the velocity u_i on the local tangent plane of the surface. The ratio assumes a value of 1 when u_i is tangent to the surface, and a value of 0 when u_i is orthogonal to the surface. Any vectorial or tensorial quantity is treated analogously.

Our third strategy enables the comparison of entire families of time surfaces by taking advantage of the spatial dimension freed up by the planar reformation. Given a family of reformed time surfaces, we stack them one above each other, in an iterative fashion (Fig. 4.10 right). We place the first surface (i.e., the seeding plane) at the origin of the coordinate system. Once the position of the n-th surface $\overline{\varphi}^n$ has been determined, $\overline{\varphi}^{n+1}$ is placed parallel to $\overline{\varphi}^n$ and displaced along the normal direction. The displacement can be either user defined or proportional to the difference in integration time between the two surfaces. In order to determine the position and orientation of $\overline{\varphi}^{n+1}$ in the plane, we exploit the fact that any two time surfaces in the same family are in a one-to-one correspondence. Recall that a family of time surfaces is defined in terms of path lines seeded at the seeding plane (Ch. 3, Sec. 2). A point $x_i^n \in \varphi^n$ is associated with point $x_i^{n+1} \in \varphi^{n+1}$ if and only if they belong to the same path line. This correspondence is preserved also in the reformed space. Based on these considerations, we define the position and orientation of $\overline{\varphi}^{n+1}$ in the plane by minimizing (in a least square sense) the Euclidean distances between each pairs of corresponding points in $\overline{\varphi}^n$ and $\overline{\varphi}^{n+1}$.

One or more stacks can be displayed and explored simultaneously. Since the surfaces within a stack can occlude each other, we provide a transparency-based focus+context visualization. The user can select a scalar attribute of interest and interactively modify the transparency of the surfaces through a transfer function widget. The stacked visualization allows us to study the variation of a scalar attribute over a family of time surfaces. Additionally, the juxtaposition of multiple stacks enables the comparison of the attribute's evolutions across different families of time surfaces (Ch. 5, Sec. 4).

In case we need to compare a large number of surface families, the stacked visualization may become unpractical, due to its high consumption of visual resources. This problem is addressed by our fourth visualization approach, which summarizes the evolution of a user-selected scalar attribute through statistical measurements (Fig. 4.10 bottom left). In fact, given a point on one of the surfaces, we can evaluate the selected scalar attribute at the corresponding points on all the other surfaces in the family. In other words, each point can be associated with a distribution of values. For each point, we evaluate the arithmetic mean μ, the median med, the empirical variance σ^2 and the interquartile range IQR of the related distribution [KFH10]. We display these quantities by means of 3D box plots over a user-selected reference surface in the family (Fig. 4.10 bottom left). Specifically, a round glyph encodes the value of μ, while two black circles are placed at $\mu \pm \sigma$. The value of med is depicted by a flat square glyph, while IQR is represented by an elongated square cylinder. The statistics-based visualization

is considerably more compact than the stack-based one, therefore it enables the comparison of a larger number of surface families by juxtaposition (Ch. 5, Sec. 4).

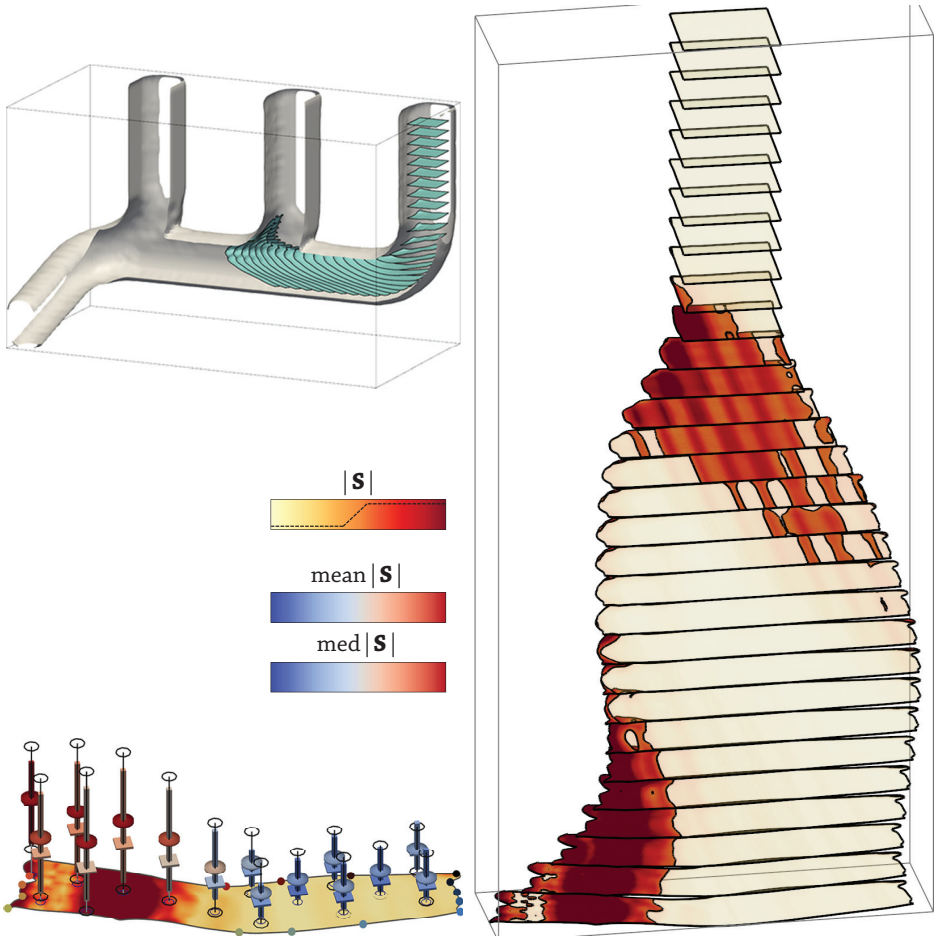


Figure 4.10: (top left) A family of time surfaces from a simulation of an exhaust manifold (Ch. 5, Sec. 1). (right) Stacked visualization of the surface family. The color and the transparency depend on the magnitude of the strain rate tensor. (bottom left) Statistics-based visualization of the magnitude of the rate of strain over the last time surface in the family.

3 Seeding Multiple Integral Surfaces

Research question

The long-term behavior of fluid flows can be effectively analysed by distributing one or more integral structures over the spatio-temporal domain. Here, we focus specifically on the problem of interactive flow investigation, i.e., we assume the user is interested in studying specific regions of the flow domain that are interactively selected. In the case of steady flows, streamlines and stream surfaces are effective tools for carrying out such an investigation. Integral surfaces are generally more expressive than integral curves, but they pose a number of challenges, such as a high computational cost and a significant consumption of visual resources. Here we focus on the problem of selecting an appropriate seeding curves for a set of stream surfaces. This is a particularly challenging issue, especially if multiple stream surfaces have to be seeded simultaneously. In fact, manually specifying arbitrarily shaped seeding curves in 3D can be a cumbersome task for the user. Additionally, there is no intuitive correlation between the seeding curves and the geometry of the resulting surfaces. Consider also that cluttering and self-occlusion can occur even when visualizing a single stream surface, as a consequence of swirling or shearing flow motion. Defining a set of seeding curves that leads to an expressive surface-based visualization, without incurring in visibility issues, is often very difficult.

Even when investigating a specific portion of a flow dataset, the fluid can exhibit several distinct motion patterns. A single stream surface may be unable to capture all the aspects of the flow behavior. Attempts have been made to maximize the informative content of a stream surface [EHS13], but there is no guarantee that the resulting visualization is uncluttered and occlusion-free. Due to visibility issues, the user may be able to perceive only a portion of the whole informative content.

Seeding multiple surfaces in the region of interest can lead to the desired characterization of the fluid flow. A visualization approach based on multiple stream surfaces needs to address all the aforementioned design challenges in order to be effective. Specifically, the system should be intuitive, easy to use, and require minimal user interaction. The burden of determining a set of seeding curves should be handled automatically, with the goal of generating expressive and easy to understand stream surfaces. In order to maximize the ratio between informative content and computation time, we should avoid computing multiple stream surfaces which convey the same piece of information. Since we are addressing a data exploration problem, interactivity is also a requirement.

Proposed approach

In accordance with the previous requirements, we designed a novel strategy for seeding multiple stream surfaces around a given point of interest. We ask the user to specify just the location to investigate and a small set of intuitive parameters, i.e., the number of stream surfaces to compute and their desired width and length. For determining a set of suitable seeding curves we follow two basic guidelines.

First of all, each surface should convey one and only one aspect of the flow behavior. This makes the interpretation of the surfaces easier. For instance, if a stream surface traverses a single vortical area, its shape would convey information about the swirling fluid's motion in that specific region. If we seed a stream surface so that it traverses several regions of different behavior, the shape of the resulting surface would become far more convoluted. Additionally, the final geometry would be the result of the combined action of different motion patterns, therefore it would be impossible to extract specific information about any of them.

Secondly, we aim at capturing a different aspect of the flow behavior with each surface. In fact, stream surfaces are expensive to compute and consume significant amounts of visual resources. Let us assume that a stream surface has been seeded so that it illustrates, e.g., the vortical motion of the fluid in a given region. Adding a second surface that depicts the same vortical behavior, could provide a few more details, but at a considerable cost. By following this guideline, we avoid wasting computational and visual resources on redundant pieces of information.

Notice that these two guidelines are also at the basis of most partition-based techniques, such as clustering and vector field topology. An example of the application of this idea to the construction of stream surfaces is shown in Figure 4.11. The stream surfaces selected by following these guidelines (Fig. 4.11c) are easy to understand, they convey a substantial amount of information, and the overall degree of occlusion is minimal.

The principal motion patterns are identified by evaluating the similarity between streamlines that are densely seeded around the location of interest. Specifically, we designed a visualization pipeline which includes the following steps:

- A number n of streamlines are seeded from a *seeding plane* placed at the location of interest.
- The similarity between each pair of streamlines is evaluated, producing an $n \times n$ similarity matix.
- The matrix is processed by a *multidimensional scaling* (MDS) algorithm [Kru64, CC10], which results in a mapping from the seeding plane to a multidimensional Euclidean space.
- The Jacobian of the resulting mapping lets us generate a *similarity tensor field* on the seeding plane.
- The desired seeding curves are given by selected tensor lines aligned with

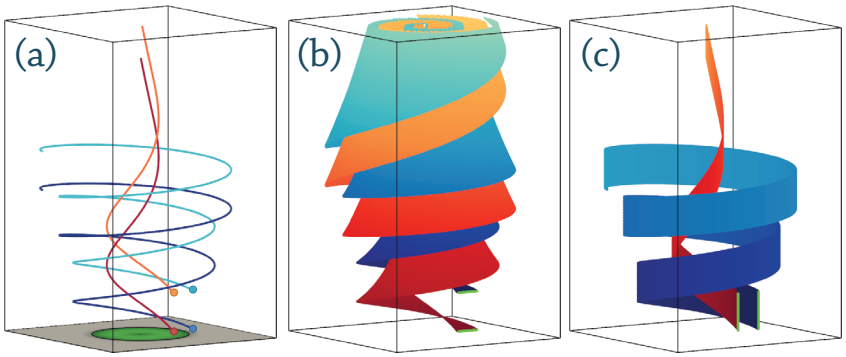


Figure 4.11: (a) Four streamlines in a synthetic dataset. The flow exhibits two different motion patterns, i.e., a converging behavior in the center and a diverging behavior in the outer regions. (b) Two stream surfaces constructed so that each of them depicts both the motion patterns. (c) Two stream surfaces constructed according to the proposed guidelines.

the "minimum" eigenvectors of the similarity tensors (i.e., the eigenvectors associated with the minimum eigenvalues).

In this way we enforce that adjacent streamlines in the resulting stream surfaces are as similar as possible to each other. Conversely, the similarity between streamlines from different stream surfaces is minimized. An example is shown in Figure 4.12. The dataset depicted here represents the flow of a fluid through a box (Ch. 5, Sec. 1). The fluid flows in the box from the top left and leaves the box through an outlet at the bottom right.

We start by constructing a *seeding plane* centered at a user-specified point of interest (Fig. 4.12a, in grey). Previous research [ELM+12] showed that a desirable property of seeding curves is to be orthogonal to the flow direction. Therefore, we generate the seeding plane so that its normal is aligned with the velocity at the selected location. The size of the plane is directly proportional to the desired width of the stream surfaces specified by the user.

In the second step, we seed a certain number of streamlines from the seeding plane. The seeding points are uniformly distributed over the plane. Assuming that the flow dataset is specified over a grid, which is often the case, our system can automatically determine the number of streamlines as a function of the grid's resolution. The integration time corresponds to the desired surface length specified by the user.

The similarity between each pair of streamlines is computed by means of the *Hausdorff distance* [Hau14]. We choose this metric because of its geometric interpretation: two streamlines are never separated by a distance greater than their Hausdorff distance. Two streamlines are considered similar if their Hausdorff

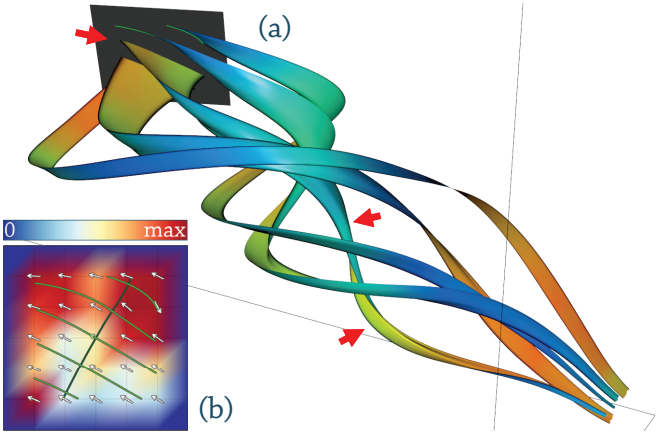


Figure 4.12: (a) Stream surfaces in the flow through a box dataset. The seeding plane is colored in grey. (b) The seeding plane colored according to the minimum eigenvalues of the similarity tensor field. The white arrows depict the corresponding eigenvector field. The light green lines are tensor lines aligned with this field. They are adopted as seeding curves for the stream surfaces.

distance is small, i.e., if they stay close to each other over their whole length. Additionally, Rössl and Theisel [RT12] showed that the Hausdorff distance can reveal topological information about the vector field.

By computing the Hausdorff distance between each pair of streamlines, we obtain a similarity matrix. We process this matrix with a multidimensional scaling algorithm (MDS) [Kru64, CC10]. This operation maps every streamline to a single point in a multidimensional Euclidean space. In this space, the distance between a pair of points is proportional to the dissimilarity of the corresponding streamlines. The streamlines are also in a one to one correspondence with their seeding points on the plane. The composition of these two mappings is a map from the seeding plane to the multidimensional Euclidean space. Notice that the seeding plane is a 2D parametric surface. By computing the derivatives of the mapping with respect to the 2D parametrization of the plane, we can compute the Jacobian J_{ij} at every point of the seeding plane.

We define the *similarity tensor field* by computing the 2×2 tensor $J_{ji}J_{jk}$ over the seeding plane. This field can be easily interpreted by computing the spectral decomposition of the tensors. At any location on the plane, the streamline most similar to the current one can be found by following the direction the "minimum" eigenvector, i.e., the eigenvector corresponding to the minimum eigenvalue (white arrows in Fig. 4.12b). In contrast, following the "maximum" eigenvectors leads to the most dissimilar streamline. In both cases, the eigenvalues express how dissimilar the streamlines are.

We construct a first seeding curve as the tensor line passing through the user-defined point of interest and tangent to the minimum eigenvectors (Fig. 4.12b, central light green line). Adjacent streamlines within the resulting stream surface (Fig. 4.12a, red arrows) are as similar as possible to each other. If more surfaces are requested, we compute the tensor line through the same point but tangent to the maximum eigenvectors (Fig. 4.12b, dark green line). We uniformly distribute a suitable number of points along this curve. The additional seeding curves are obtained by computing new tensor lines through these points, following the minimum eigenvectors (Fig. 4.12b, light green lines).

Notice that several modules of our pipeline can be parallelized. For instance, we have implemented the streamline computation using OpenMP, which enables parallelization on multicore processors. The MDS and the computation of the similarity tensor field are instead executed in parallel on the GPU. As a consequence, all the necessary computations can be performed at interactive rates. The user is free to modify the location, the number and the desired size of the stream surfaces while directly observing how the visualization changes.

More details about our seeding strategy can be found in Paper E. The paper also include an analysis of the execution times and a thorough description of the parameters involved.

4 Remarks

In this chapter, we have presented and discussed a number of visualization designs. The common factor of all of our visualization strategies is that visibility management is addressed as part of the main design challenges. Notice that each of our approaches has very specific goals which arise from the application domain:

- Our glyph-based visualization (Sec. 1) targets the problem of the simultaneous investigation of multiple aspects of the flow behavior.
- The visual inspection and the comparison of one or more integral surfaces is addressed by our techniques based on geometry splitting and reformation (Sec. 2)
- We support the interactive analysis of the long-term flow behavior with our seeding strategy for integral surfaces (Sec. 3).

It is important to emphasize that, while designing these visualizations, the overall objective was to provide the user with solutions to specific problems, not to develop generic visibility management techniques. Visibility management is considered as one of the requisites of the entire visualization solution.

By adopting this design strategy, it is likely that the solution will include one or more components that address visibility issues explicitly, but there is no guarantee that this will be the case. For instance, the coherency-based pruning from our work on multi-aspect flow visualization (Sec. 1) can be also considered as a

stand-alone technique for addressing cluttering and occlusion. It can be potentially reused in other visualization scenarios. In contrast, our seeding approach for stream surfaces (Sec. 3) is able to generate occlusion- and clutter-free visualizations without addressing these problems explicitly. Visibility issues are prevented instead of being resolved a posteriori.

These examples highlight another important consideration. If visibility issues are tackled as stand-alone problems, the resulting visibility management techniques have to comply with the constraints imposed by the existing visualization solution. For instance, assume a visualization expert designs a glyph-based visualization which includes a certain placement strategy. If the results turn out to be too cluttered, the expert can decide to adopt a visibility management technique on top of the existing design. But at this point, the visibility management technique would be unable to modify the placement of the glyphs, since that is determined by an existing component of the pipeline. In contrast, addressing the visibility issues in the early stages of the design provides a wider range of potential solutions. Referring to the previous example, the visualization expert could, for instance, opt for a placement strategy which satisfies the initial requirements *and* addresses cluttering and occlusion.

Chapter 5

Demonstration and Discussion

We have developed software prototypes based on the visibility-oriented visualization designs described in the previous chapter. In order to demonstrate the effectiveness of our designs, we have employed our prototypes for investigating a multitude of application scenarios, including analytical flow datasets and CFD simulations. By showing that the software prototypes are able to tackle the problems they were designed for, we indirectly validate the visibility-oriented design process. Due to space limitation, this chapter presents and discusses only a selection of the application cases we have addressed during our research activities. We first give a brief overview of the four flow dataset we have selected for this demonstration. Then we proceed to describe the results delivered by each of our visualization designs. More demonstration cases can be found in the related papers.

1 Flow Datasets

For this demonstration we have selected four application scenarios which allow us to showcase the capabilities of our visualization strategies. All the flow datasets discussed here are generated by CFD simulations. The data is structured according to a Cartesian grid, which can be either the one adopted for the original simulation or the result of a resampling process.

The first scenario is a simulation of a fluid flowing through a box-like structure (Fig. 5.1 bottom). The inlet is a rectangular tube on the top of the box while the outlet is a square hole on one of the box's faces. The motion of the fluid is mainly laminar at the inlet and along the bottom wall. The flow in the rest of the box is characterized by significant rotation patterns, as we will show in Sections 2, 3 and 5.

The second case is a time-dependent simulation of the exhaust manifold of a combustion engine (Fig. 5.1 top). The exhaust gases enter the manifold through one of the three inlets on the top, traverse the horizontal pipe, and then leave the manifold through the outlet at the bottom. The performance of the combustion engine is affected by the efficacy of the manifold in expelling the gas. For this reason, it is important to investigate flow phenomena that can potentially reduce the speed of the fluid or, in general, increase the time the fluid spend in

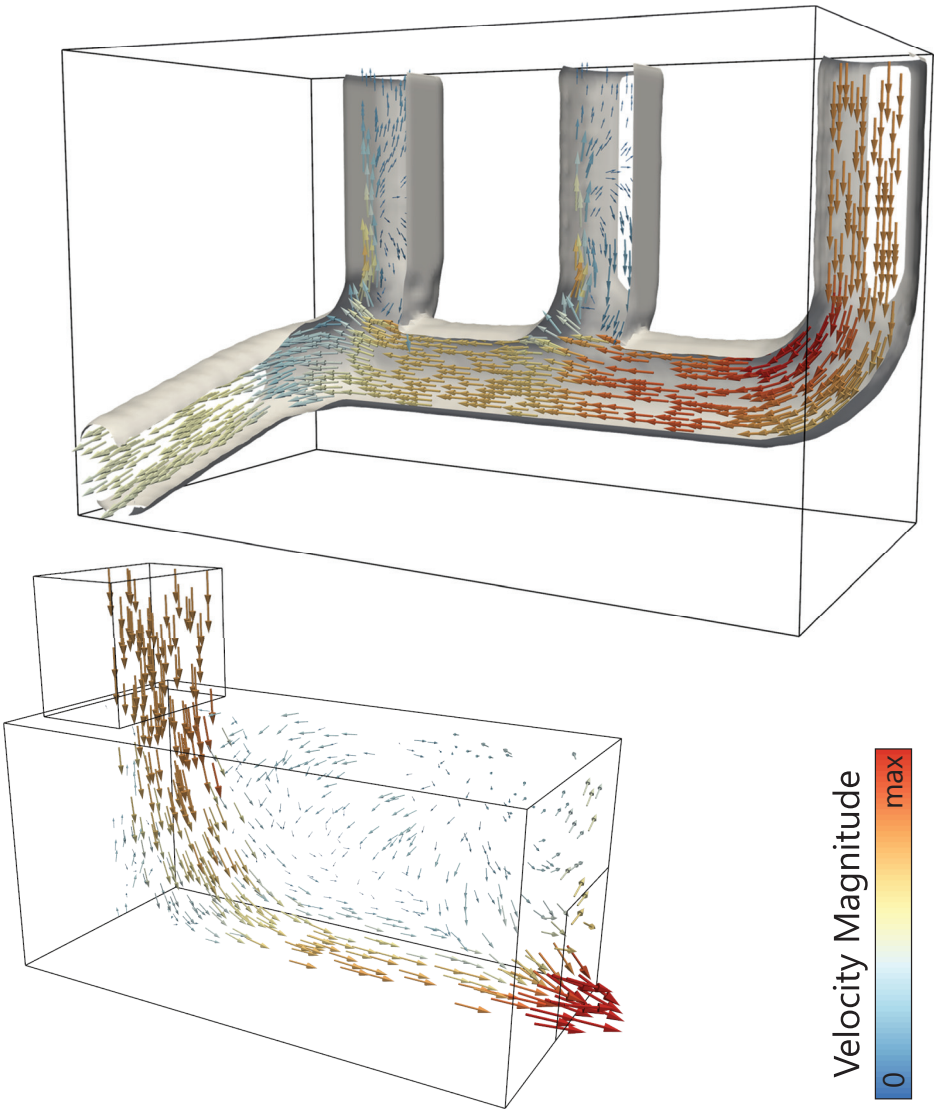


Figure 5.1: (Top) One timestep of the exhaust manifold dataset. The right-most inlet is currently open while the other two are closed. (Bottom) Overview of the flow in a box dataset. In both cases, arrow glyphs are shown at selected locations in order to illustrate the instantaneous velocity fields. The glyphs are scaled and colored according to the velocity magnitude.

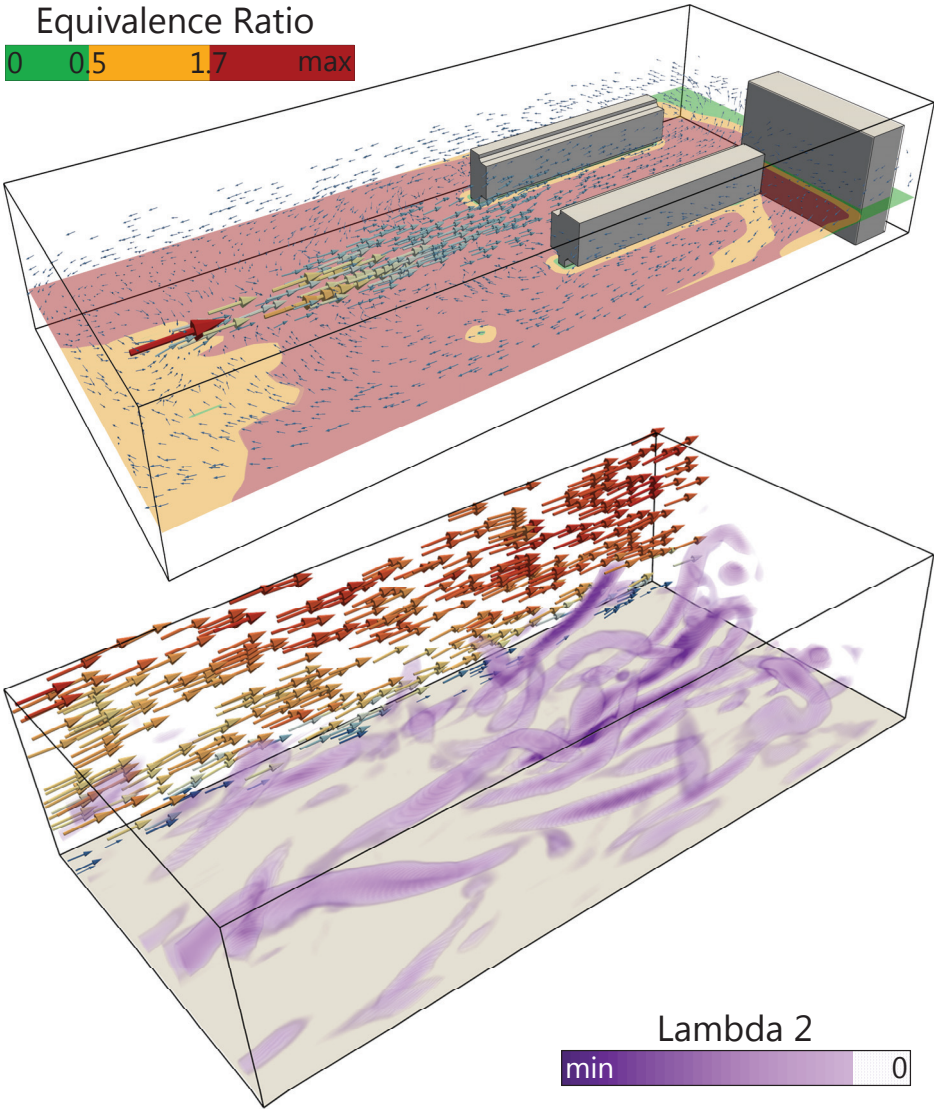


Figure 5.2: (Top) The gas leak dataset. The gas leak is located in correspondence with the large red arrow. A planar slice is colored according to the *equivalent ratio* field. (Bottom) A timestep of the turbulent channel dataset. Streamwise vortices along the bottom wall are exposed via volume rendering of the λ_2 scalar field [JH95]. In both pictures, arrow glyphs are placed for providing an overview of the velocity, as in Fig. 5.1.

the manifold. Notably, a turbulent flow behavior, characterized by swirling and straining motion, can reduce the overall velocity of the exhaust gases. Additionally, the inflow in each inlet is regulated by an exhaust valve. The valves open periodically, and whenever one of the valves is open, the other two are closed. Ideally, the gas should flow from the active inlet towards the outlet only. However, it may happen that a portion of the fluid flows towards the inactive inlets, increasing the pressure in those locations (backpressure). The elevated pressure can contrast the expulsion of gas through the corresponding inlet at a later time, when it opens. Our study of the exhaust manifold is discussed in more details in Sections 2 and 4.

The third case is a simulation of a gas leak in a closed room on an oil platform. In Figure 5.2 top, the gas leak is located in correspondence with the large red arrow. In addition to the velocity field, this dataset includes the *equivalence ratio* scalar field. The equivalence ratio is related to the normalized concentration of gas in the spatial domain. In this particular case, an equivalence ratio between 0.5 and 1.7 means that the air-gas mixture is flammable. For values above 1.7, the mixture is unbreathable. Domain experts are interested in examining how the gas spreads in the room. Specifically, they are interested in investigating how the presence of obstacles affects the dispersion of the gas. It is also important to locate and analyze the regions where the gas mixes with air. Sections 3 and 5 describe our study of this dataset.

The fourth application case is a time-dependent simulation of a statistically stationary flow in a turbulent channel. Referring to Figure 5.2 bottom, the fluid flows from the left towards the right side. The flow is periodic with respect to the front and back faces. The domain is bounded by two walls (no-slip boundaries) on the upper and lower side (only the bottom half of the channel is shown in Figure 5.2). The flow along the walls is mainly turbulent. Here, the features of interest are represented by typical turbulent structures, such as streamwise and *hairpin* vortices, and shear layers. In this dataset, Taylor's hypothesis [KC08] is nearly satisfied. In other words, the properties of the turbulent structures change slowly over time (the total derivative is almost zero). An analysis of the turbulent channel is described in Section 4.

The simulations of a flow through a box and of the exhaust manifold are courtesy of AVL List GmbH, Austria. The gas leak dataset is courtesy of GexCon AS, Norway. The simulation of the turbulent channel flow is courtesy of the Norwegian Defence Research Establishment (FFI).

2 Analysis of Multiple Flow Aspects

We have applied our technique for visualizing multiple flow aspects (Ch. 4, Sec. 1) to the flow through a box and to one timestep of the exhaust manifold dataset. For the flow through a box, we aim at generating an overview of the motion of the

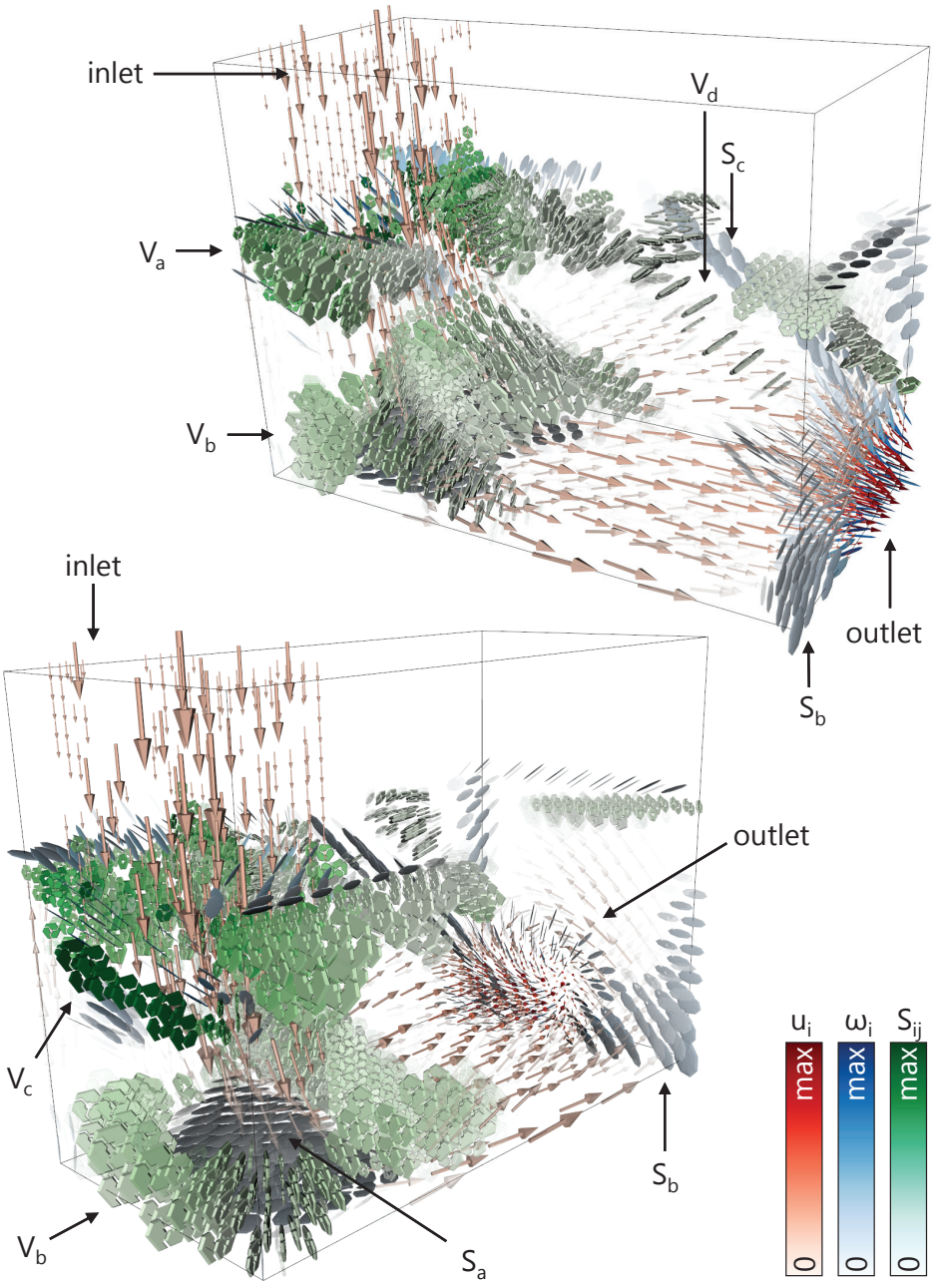


Figure 5.3: Visualization of multiple flow aspects in the flow through a box, from two different points of view. Areas denoted with a V exhibit a vortical flow behavior, while strain-dominated areas are labelled with an S (see Sec. 2 for a detailed discussion).

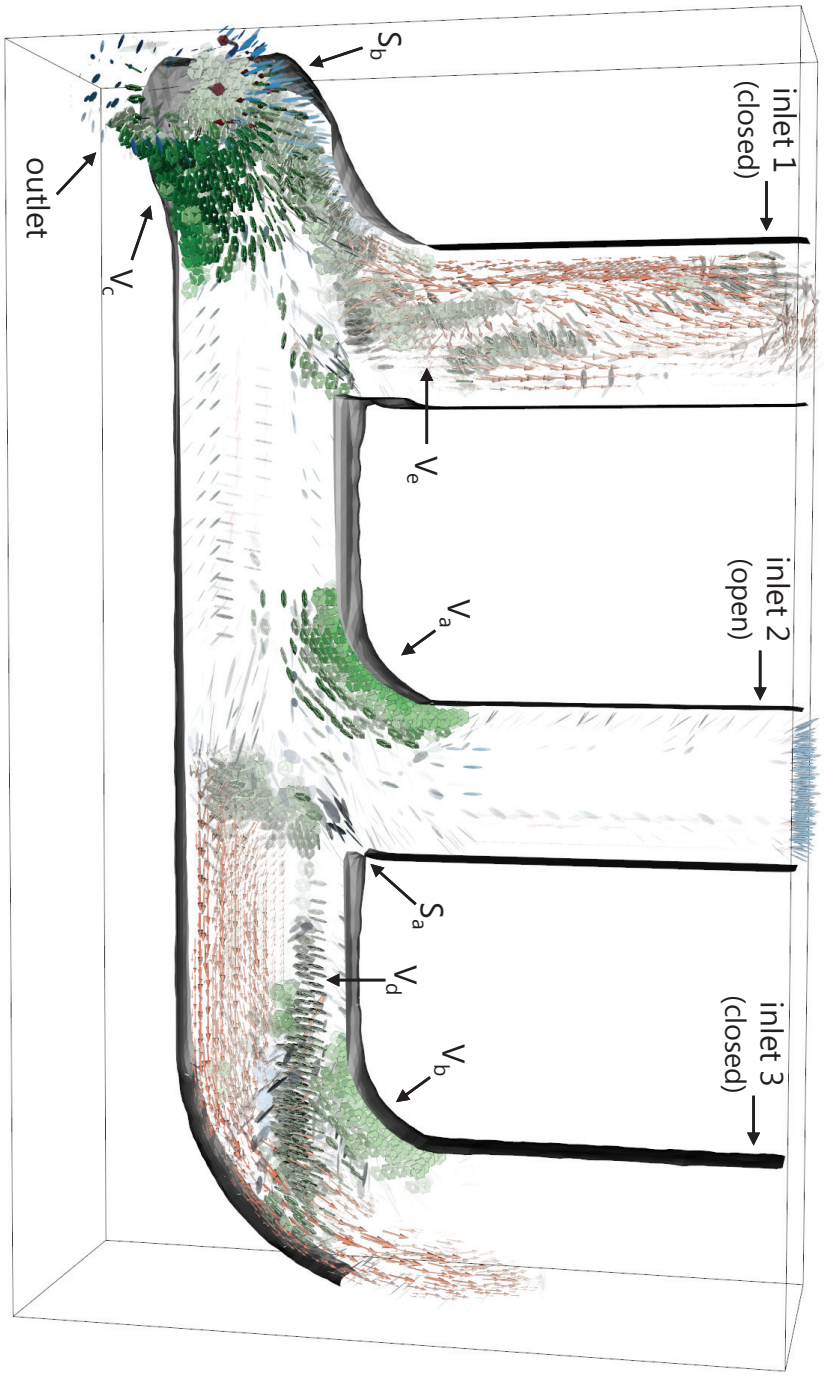


Figure 5.4: Multiple flow aspects in one timestep of the exhaust manifold dataset. The central inlet is currently open.

fluid through the spatial domain, including information about vortical and strain-dominated regions. Therefore, we have defined the relevance of the attributes of interest as follows. The flow velocity is considered relevant when it has a high magnitude. This lets us emphasize the most prominent flow of the fluid, from the inlet to the outlet. The vorticity is visualized in correspondence with positive values of the Q feature detector [HWM88] in order to identify vortical regions. In areas where Q is negative, that is, in strain dominated regions, the strain rate is depicted instead. The result is shown in Figure 5.3. First of all, we can see that the region where the inlet connects to the main body of the box is characterized by a swirling flow behavior (V_a in Fig. 5.3). The collision between the inflow and the bottom wall of the box causes elevated strain (S_a). Extensive vortices are generated around this strain dominated area (V_b). The color of the glyphs indicates that the strength of the swirling motion in V_a and V_b is only moderate. In contrast, a very strong vortical motion can be detected in a thin elongated region close to the left wall (V_c). Notice that a portion of the fluid moves upwards from V_b against the left wall. The high vorticity in V_c is caused by the collision of this upward flow ad the downward flow from the inlet. The flow at the outlet is particularly fast. The surrounding area is characterized by elevated strain (S_b), indicating that the low velocity particles close the the right wall are dragged towards the outlet. The flow from the upper part of the box towards the outlet is mainly vortical (V_d). Interestingly, we can see that such vortical motion causes the detachment of a shear layer from the back wall (S_b).

In the case of the exhaust manifold, we visualize the velocity vectors whenever they are directed away from the outlet, either vertically or horizontally. This condition can be easily expressed in terms of the individual components of the velocity vectors. In this way, we can detect the motion patterns that can generate back pressure (Sec. 1). The vorticity is considered relevant in correspondence with negative values of the λ_2 vortex detector [JH95], while negative values of Q are again adopted for the rate of strain. Figure 5.4 shows the result of our technique. In the selected timestep, the central inlet (inlet 2) is active while the other two (inlet 1 and 3) are closed. Our visualization clearly shows the presence of strong swirling behaviors in the inner part of the bends (V_a and V_c). This pattern is present also in the third inlet pipe (V_b), even though the corresponding inlet is inactive. Also, in the outer part of the bends, elevated strain can be detected (S_a and S_b). As mentioned before, vortical and straining behaviors are undesired since they can reduce the effectiveness of the manifold and reduce the engine's performance. The arrow glyphs emphasize the flow patterns that can lead to back pressure. It is clear that a portion of the fluid flows towards the first and the third inlets. An additional issue is exposed by our technique. When the fluid's particles in the inactive inlets start moving in the correct direction (towards the outlet), they induce a notable production of vorticity (V_d and V_e). When the corresponding inlet is open, such turbulent structures can significantly affect the flow of the exhaust emissions.

A high resolution version of the pictures in this section can be found in the Appendix of Paper B.

3 Visual Inspection of Integral Surfaces

We demonstrate our splitting-based strategy for the visual inspection of integral surfaces (Ch. 4, Sec. 2) over the flow through a box and the gas leak datasets. Figure 5.5 shows three consecutive cuts of a stream surface in the flow through a box. The seeding curve is a straight line placed slightly downstream from the inlet, close to the top wall. Colors represent the integration time. The surface captures two distinct aspects of the flow behavior: the almost laminar flow close to the bottom wall and the vortical motion in the upper part of the box. The rotational behavior makes the surface roll up, leading to significant occlusion. By cutting the surface with the proposed technique (along the streamline in green), the inner part of the convoluted surface region is separated from the outer part. Notice that the outer part is almost free of occlusion, since each of its points can be directly observed from at least one viewpoint. The inner part instead is still occluded, so we cut it again. The second cut (cyan, along a streamline) separates the innermost part (bottom right) from the rest of the surface. A third cut (purple, along a streamline) separate the two remaining layers of the roll. We are now able to study the initial surface by observing the four leafs of the splitting tree. For instance, we can now clearly see that the inner region of the vortical structure is characterized by higher integration times than the outer one. In other words, the particles in the outer region reach the outlet rather quickly, while the particles in the inner region move significantly slower.

Figure 5.6 shows instead four cuts of a stream surface in the gas leak dataset. The surface is seeded along a straight line placed slightly downstream from the leak. We see that the fluid rapidly flows between the two lateral obstacles and hits the one at the end of the room. The surface then divides in two and, at later integration times, one of the branches is dragged in the vortical regions in front of the obstacles. Rotational flow patterns are important since they affect the mixing between air and gas. Already at the first cut (green, along a streamline), the surface part characterized by the swirling flow behavior is separated from the rest. The second cut (cyan, along a time line) separates the low integration time values from the higher ones. The sharp discrepancy of integration times is due to the strong divergence of the flow when moving around the right obstacle. It appears that the fluid particles passing under the obstacle travel much faster than those passing behind the obstacle. A further cut of the convoluted surface (purple, along a time line) shows that the vortical structure in front of the left obstacle is reached sooner than the inner swirling area in front of the right obstacle (notice that the viewpoint is changed). Additional cuts enable the exploration of the core of the swirling region. These results were discussed with and supported

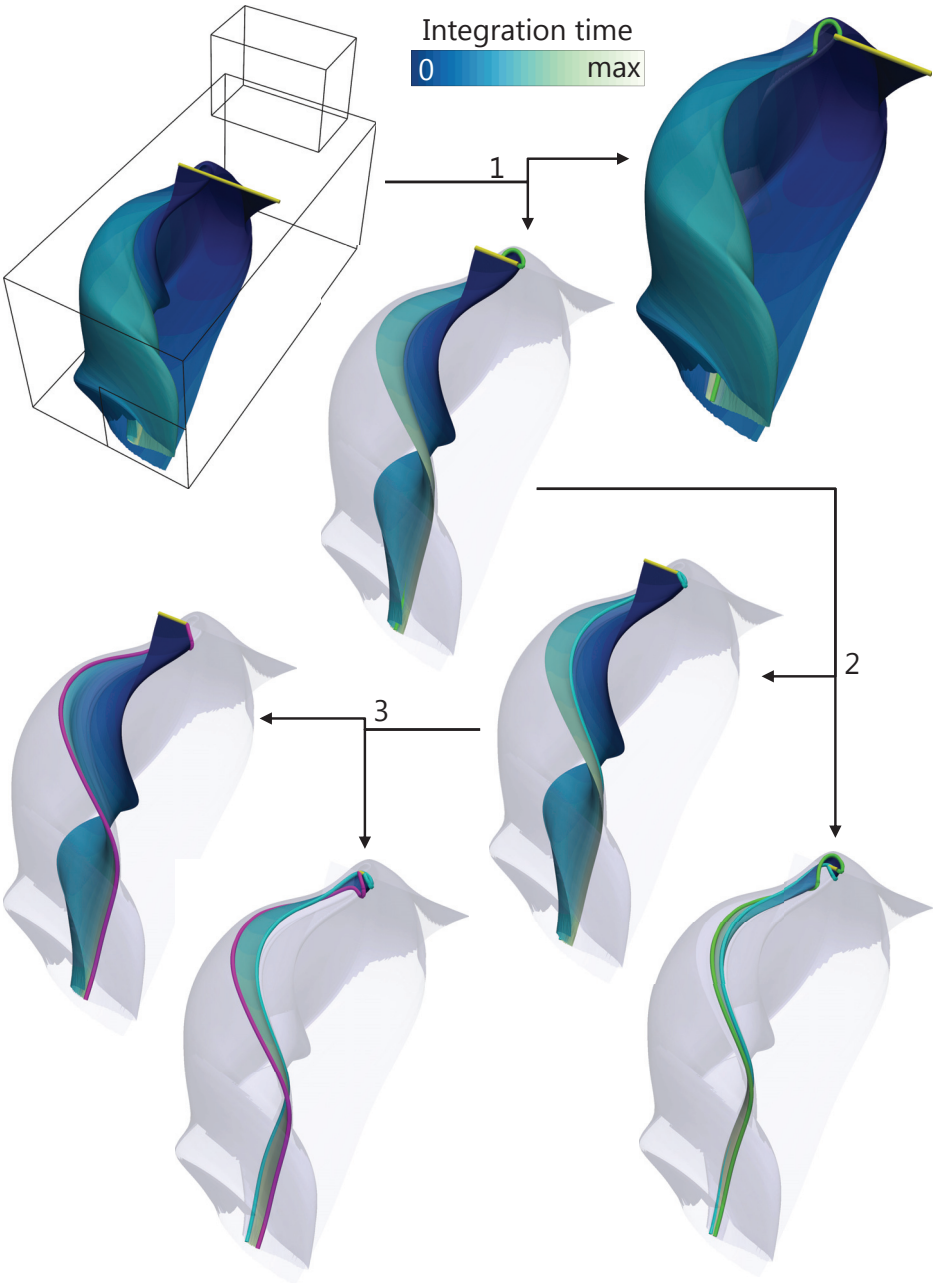


Figure 5.5: Splitting a stream surface of the flow through a box. All the three cuts are along streamlines.

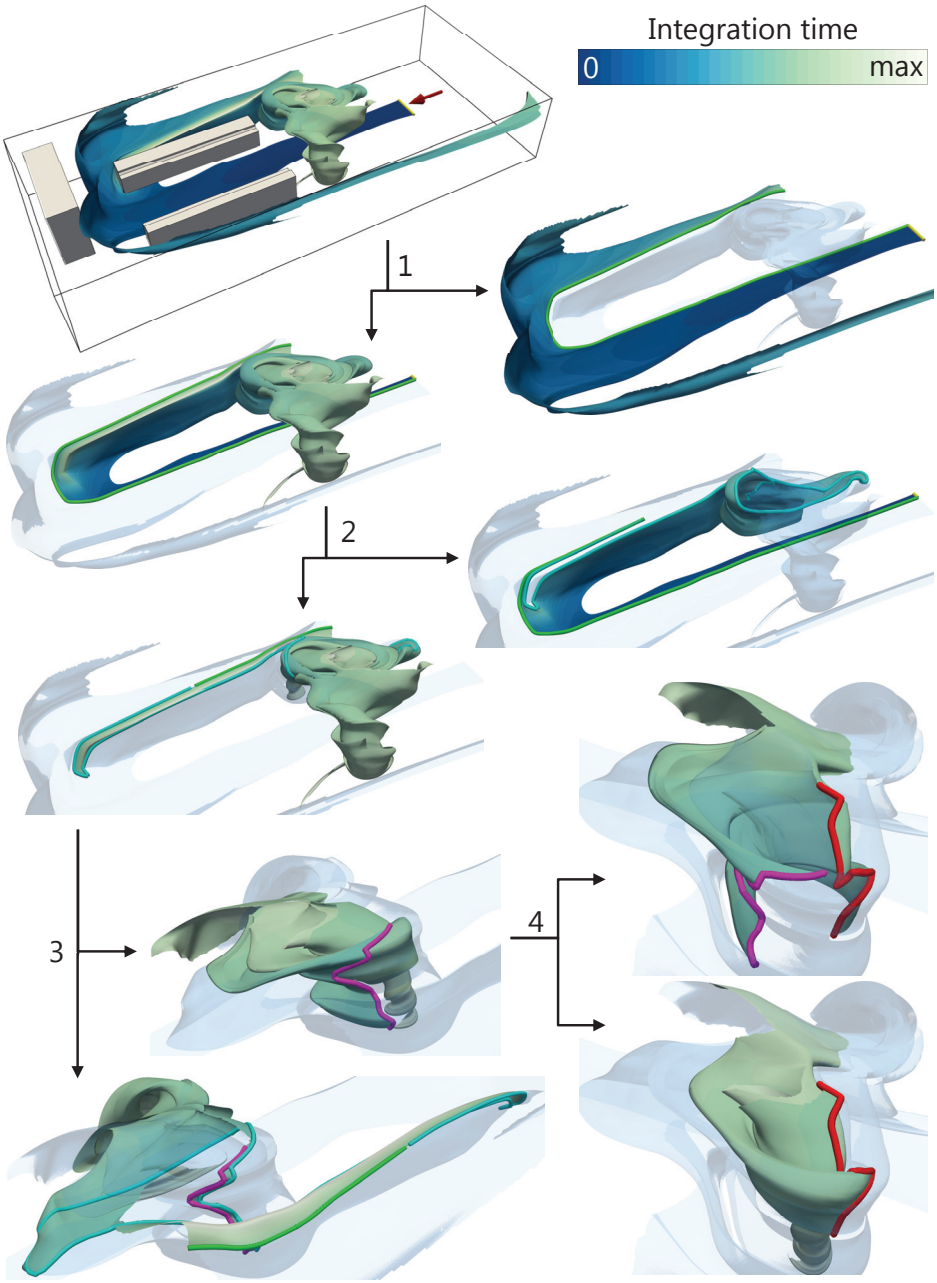


Figure 5.6: A stream surface in the gas leak dataset is split four times. The first split is along a streamline, the following three are along timelines.

by a domain expert from GexCon AS, the company that computed the simulation of the gas leak.

4 Comparative Visualization of Surface Families

We showcase the capabilities of our comparative visualization strategies (Ch. 4, Sec. 2) by presenting our studies of the turbulent channel and of the exhaust manifold datasets. Figure 5.7 shows a comparison of two time surfaces from the turbulent channel. Surface φ has been seeded in correspondence with a hairpin vortex (Fig. 5.7a), detected with the Q criterion [HWM88]. Surface ψ is instead seeded between two streamwise vortices. The surfaces are visualized in the reformed 2D space, together with tensor lines of the metric tensor field and the corrected velocity, vorticity and strain rate fields. The tensor lines of the metric tensor field show that only small deformation are introduced in the top part of the surfaces. This is an indication that, in correspondence with those portions of the surfaces, the motion of the fluid is mainly laminar. The directions of the velocity vectors (Fig. 5.7d) and the low magnitudes of vorticity and strain rate support this interpretation. The abrupt changes in the directions of velocity and vorticity in the top part of $\overline{\varphi}$ denote the presence of a thin layer which moves faster than the areas above and below it (shearing). The strain rate field (Fig. 5.7f) shows that the surface is stretched by this fast moving layer, especially in the area just below the layer.

In the lower parts of both surfaces, we can identify regions characterized by significant distortions (Fig. 5.7c, red and blue lines). The tensor lines indicate the presence of protruding regions in the original surfaces, analogously to Figure 4.8b (Ch. 4, Sec. 2). In $\overline{\varphi}$, The red tensor lines are roughly aligned, meaning that the protrusion has a larger extent along that direction, as can be seen in Figure 5.7b. To the lower left and to the right side of the protrusion, the vorticity and the velocity are aligned, which suggests the presence of vortex cores, in accordance with the *helicity* criteria [PR99]. These correspond to the two vortex legs of the initial hairpin vortex. The principal direction of the strain rate is aligned with the cores, and this is compatible with the presence of streamwise vortices [MK85]. In $\overline{\psi}$, the red tensor lines show that the distortions around the protrusion have approximately the same intensity in every direction. The velocity and the vorticity exhibit patterns similar to swirling critical points, with large variations in orientation. A reasonable interpretation is that the protrusion is generated by a turbulent structure which is aligned orthogonally to the surface, possibly one of the streamwise vortices adjacent to the seeding plane. The color of the ellipsoid glyphs reveals that the stretching in the normal direction is limited, indicating that the vortex and the time surface advance through the flow at approximately the same rate.

Figures 5.8 and 5.9 show respectively a stack-based visualization and a statistics-

Figure 5.7: Visual comparison of two time surfaces. (a) The two time surfaces (green) in the turbulent channel dataset. The seeding planes (in blue) are placed in correspondence with a hairpin vortex (left) and two streamwise vortices (right), detected with the Q criterion [HWM88] (in purple). (b, c) The original and the reformed surfaces with the tensor lines of the metric tensor field. (d, e, f) Visualizations of the reformed velocity, vorticity and strain rate.

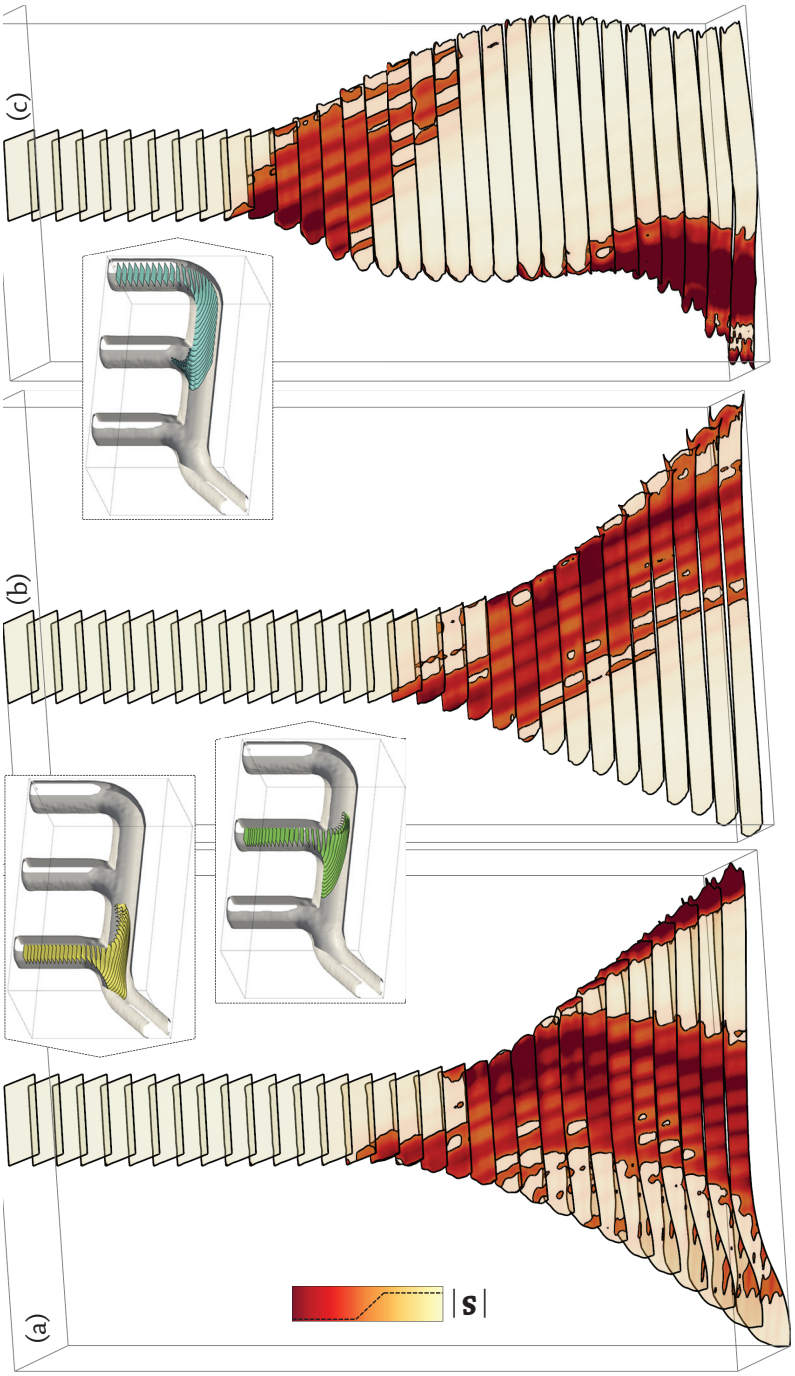


Figure 5.8: Stacked visualization of three families of time surfaces from the exhaust manifold dataset.

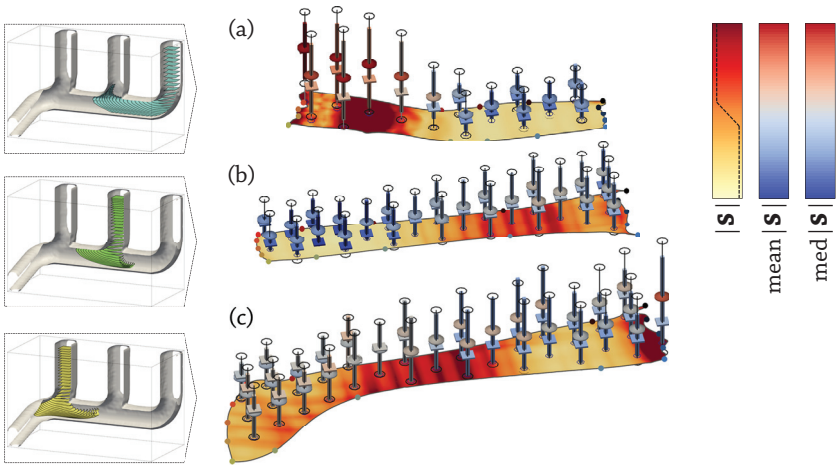


Figure 5.9: Statistics-based visualization of the same families of time surfaces from Figure 5.8.

based visualization of three families of time surfaces computed in the exhaust manifold dataset. Each family of time surfaces has been seeded at the beginning of a different inlet. The surface integration is started when the gases in the corresponding inlet reach the maximum speed, i.e., a few moments after the inlet opens. The integration time is the same in all the three cases. In our visualizations, we focus on the magnitude of the strain rate, since intense strain can reduce the fluid's velocity and it can also be an indication of backpressure. The surface family in the rightmost inlet (Fig. 5.8c) traverses two regions of high strain magnitudes. The one at earlier integration times is a consequence of the curved shape of the manifold. The one at later integration times is instead due to the upstream motion of fluid particles against the left wall of the central inlet, which also causes an increase of backpressure. A similar behavior can be identified in the surface family in the central inlet. The fluid particles on the right side collide with the bottom wall of the manifold, and a part of them moves towards the right inlet (Fig. 5.8b). The leftmost inlet is the one that exhibits the most problematic flow behavior (Fig. 5.8a). The intense strain in the central area of the surfaces indicates that a conspicuous amount of gas flows back towards the other inlets.

The statistics-based visualization (Fig. 5.9) provides a compact overview of the strain rate magnitude across the families of time surfaces. In Figure 5.9a and b, we see that the largest values of mean strain are concentrated, respectively, in the right and in the left portions of the surfaces. As previously discussed, these areas are associated with backpressure phenomena. Concerning the leftmost

inlet, the average magnitude of the strain rate is rather high across the whole surface. Since intense strain can affect the performance of the manifold, this surface family should be investigated more thoroughly. In all the three cases, we can see that, while σ^2 and IQR generally agree, μ tends to be larger than med to various degrees. This indicates that the strain rate magnitudes are more or less normally distributed, but there are a few surfaces with significantly larger values. In practice, the strain is particularly intense only in confined local regions in the dataset, e.g., in correspondence with the curved tube on the right. Identifying and studying these regions can provide useful insight for optimizing the shape of the manifold and improving its performance.

Notice that, after an initial pre-computation phase, our visualization strategies have real-time performance. Our results have been discussed with a domain expert, i.e., my co-supervisor Øyvind Andreassen. His feedback can be found in Paper D. The paper also includes more details about the visualization techniques and a more extensive discussion of the application scenarios.

5 Seeding Sets of Integral Surfaces

In order to demonstrate our technique for seeding multiple stream surfaces, we compare it to two state-of-the-art approaches. Most of the existing seeding algorithms for stream surfaces aim at providing an overview of the flow behavior over the whole domain. In contrast, our technique is aimed at the interactive investigation of the flow at specific locations of interest in the domain. We have decided to perform a comparison with the two strategies by Edmunds et al. [EML$^+$11, ELM$^+$12], since they can be easily reworked in order to provide interactive exploration capabilities. Edmunds et al.'s technique from 2011 [EML$^+$11] computes the seeding curves as iso-lines of a scalar field defined on the boundaries of the domain. The scalar field is given by the dot product between the velocity at the boundary and the normal of the boundary surface. We have adapted this technique by computing the scalar field and the iso-lines on the seeding plane instead of the domain boundaries. The second seeding strategy [ELM$^+$12] instead includes two phases: a clustering of the spatial domain and the construction of the seeding curve. The clustering is necessary for determining the location of the seeding curves. In the case of interactive exploration, the location of interest is given by the user, therefore the clustering is unnecessary. The seeding curves are constructed by integration over the curvature vector field. Therefore, we construct a first seeding curve by integrating the curvature vector field from the user-selected point of interest. If more than one surface is requested, we compute the vector field orthogonal to both the curvature and the velocity vectors. We determine new points of interest by following the orthogonal vector field in forward and backward direction. Additional seeding curves

are generated by integrating the curvature vector field from the new points of interest.

Figure 5.10 shows the result obtained by seeding five stream surfaces in the flow through a box. The goal is to depict the vortical flow behavior upstream from the outlet. Our proposed approach lead to an uncluttered and easy to understand visualization. The central surface has a rather linear trajectory and a pronounced twisting behavior, indicating that it is close to the vortex core. The other surfaces instead follow wider and more curved trajectories. They lie in outer regions of the swirling area. The techniques by Edmunds et al. produce surfaces which are frequently overlapped. The resulting visualizations present high degrees of occlusion, therefore the amount of information conveyed is limited.

The result obtained on the gas leak dataset is shown in Figure 5.11. Five surfaces are seeded downstream from the gas leak, close to the left obstacle. The surfaces generated by our seeding strategy clearly convey five distinct patterns in the fluid's motion around the obstacle. The results obtained with Edmunds et al.'s techniques capture approximately the same information, but the visualizations are more cluttered and difficult to interpret.

We have presented and discussed our seeding strategy with a domain expert from GexCon AS. The ease of use and the clarity of the results were between the most appreciated qualities of our system. More details about the expert's evaluation can be found in Paper E.

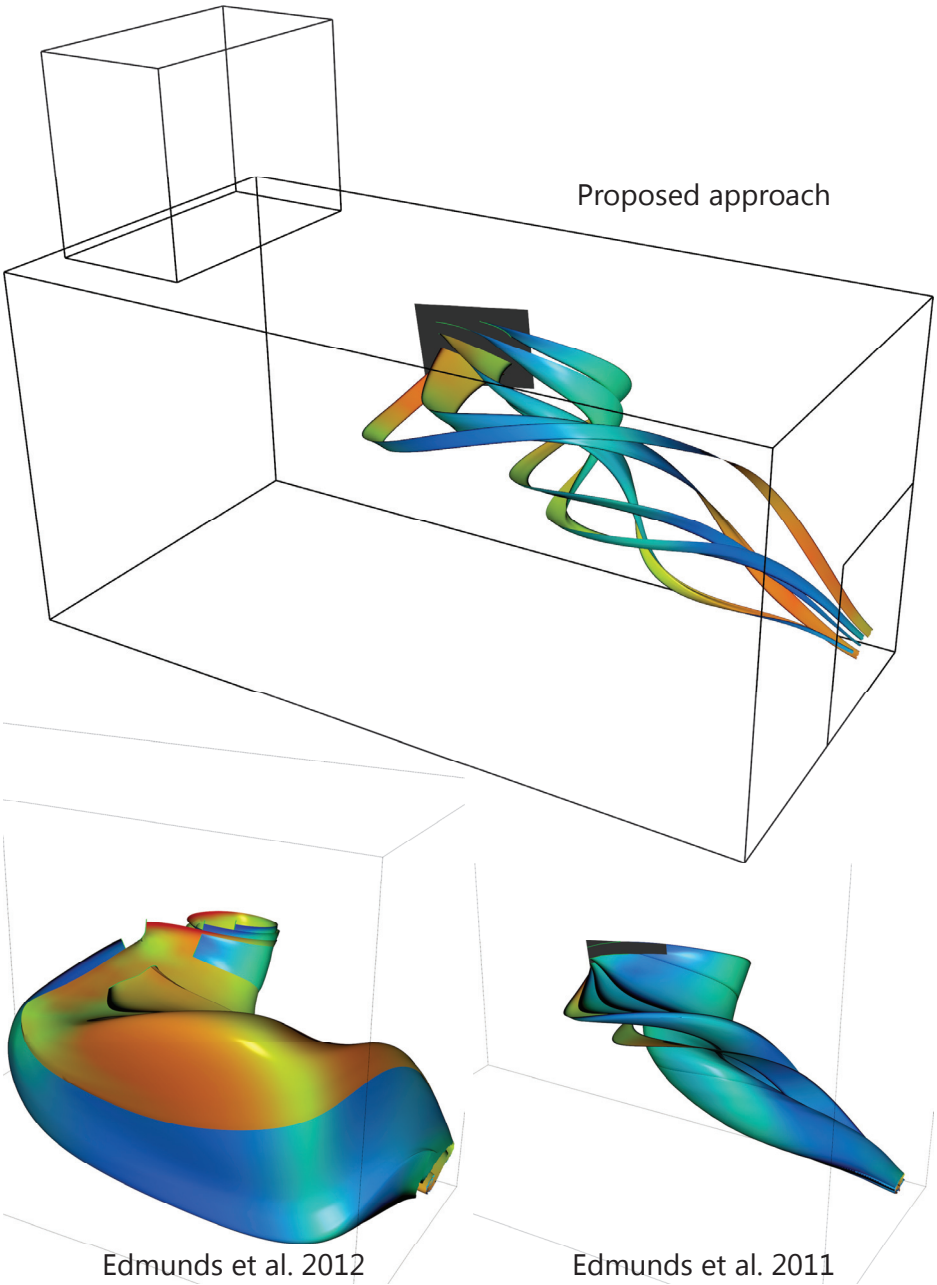


Figure 5.10: Seeding multiple stream surfaces in the flow through a box. (Top) Our approach. (Bottom) Results obtained with the techniques of Edmunds et al. [ELM+12, EML+11].

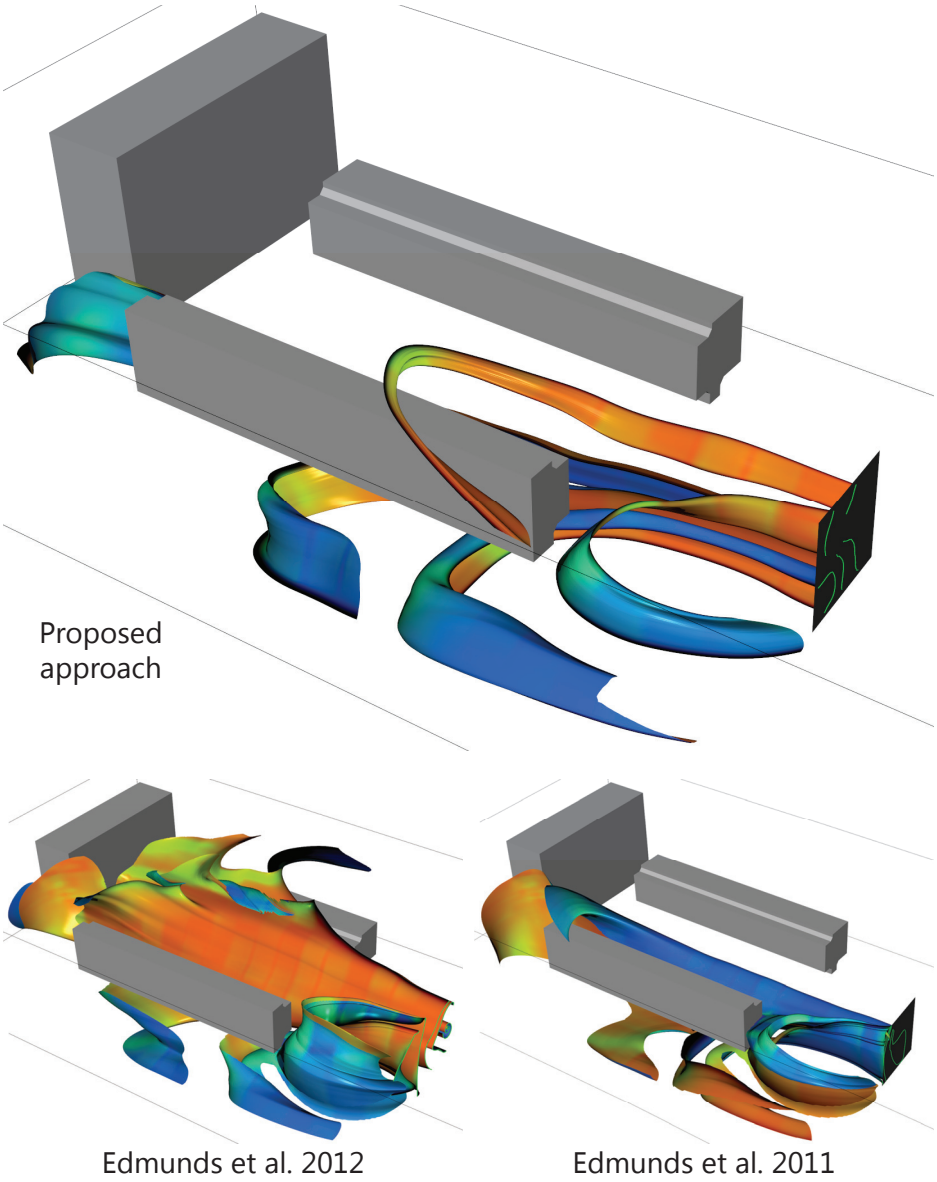


Figure 5.11: Multiple stream surfaces in the gas leak dataset. (Top) Our approach. (Bottom) Results obtained with the techniques of Edmunds et al. [ELM+12, EML+11].

Chapter 6

Conclusion and Future Work

Producing expressive and comprehensive visualizations of fluid flows is an arduous task. Most of the challenges arise from the intrinsic complexity of this kind of phenomena, in terms of both dimensionality (4D) and number of variables involved (pressure, temperature, velocity, vorticity, and so on). Thus, a vast amount of data is necessary for representing the motion of a fluid accurately. Since the visual resources provided by visualization devices are limited, flow visualization techniques need to adopt proper strategies for handling the sheer size of the data being visualized. Visibility issues, such as cluttering and occlusion, occur when these strategies fail in allocating the available visual resources properly.

Visibility issues have been traditionally addressed as stand-alone problems. In this thesis, we instead advocate that visibility management should be treated as a general requirement of an entire visualization solution. Accordingly, the resolution of visibility issues should be tackled already in the visualization design phase. In the course of this Ph.D. project, we have investigated a number of flow visualization problems and we have designed suitable visualization solutions in accordance with this design guideline. By addressing visibility issues early in the design phase, we were able to design expressive visualization approaches in three specific application scenarios:

- We have addressed the simultaneous visualization of multiple aspects of the motion of a fluid. Our glyph-based visualization allows the user to observe and to study the interactions of vectorial and tensorial quantities across the domain, and to deduce their impact on the flow behavior.
- We have designed a number of strategies targeted at the visual inspection and comparison of integral surfaces. The proposed techniques are effective in tackling occlusion issues in different application cases, from the analysis of a single surface to the comparison of entire families of integral surfaces.
- We have proposed a novel seeding technique for supporting the user in studying the long-term behavior of fluid flows. Our seeding method is able to generate clear and expressive stream surfaces while keeping occlusion and cluttering under control.

The proposed approaches are effective in conveying the desired information in a clear and uncluttered way. This is a clear indication that our designs were

successful and that the principle of addressing visibility issues in the design phase represents a valuable and promising design methodology.

1 Lesson Learned

One of the main outcomes of this thesis, and of the related Ph.D. project, is a demonstration of the potential of *visibility-oriented visualization designs* in the context of flow visualization. We have shown that this design principle can lead to effective visualizations of fluid flows.

We should keep in mind that the overall performance of a visualization approach depends on its ability to communicate information about the phenomenon of interest to the user. The final goal of the design phase is to maximize the expressiveness of the resulting visualization. Our work shows that an appropriate visibility management is a key factor in achieving this goal, and that this aspect can be successfully integrated in the design process. An improper treatment of visibility issues can essentially hinder the expressiveness of an otherwise effective visualization approach.

Our investigation of visibility-oriented visualization designs also reveals a valuable property of this principle. When visibility issues are treated as stand-alone problems, the solutions need to comply with the constraints imposed by previously designed components of the visualization pipeline. In contrast, most of these constraints are lifted if visibility management is addressed as part of the visualization requirements. The design choices can then lean towards a stand-alone solution (such as our coherency-based pruning, Ch. 4, Sec. 1), or towards a unified management of both the visibility and the other visualization requirements (as in our seeding strategy, Ch. 4, Sec. 3). Irrespective of which path is followed, the advantage is that the entire design is driven by the actual needs of the user, and not by limitations imposed, e.g., by a certain preprocessing operation or by the adoption of a specific visual entity.

Novel visibility management techniques can arise as a by-product of such a design process. Notably, the effectiveness of these derived techniques has been already studied and demonstrated on the visualization problem under consideration. Such visibility management techniques can become reliable visualization tools, and they can be reused as parts of the solutions to other application scenarios.

2 Future Work

This thesis represents an important step in the study of visibility-oriented visualization design. Our research was focused on selected problems in flow visualization, and we showcased the effectiveness of this design principle on a number

of application scenarios. For a discussion of the possible improvements of our visualization designs, we refer to the related papers in Part II of this thesis.

Our long-term goal is to provide a complete characterization of the visibility-oriented design concept, and to promote it as an essential design guideline for flow visualization. Therefore, we plan to first investigate its potential applicability to other flow visualization challenges. For instance, revealing topological information of three- and four-dimensional vector fields is known to be a challenging task. The direct visualization of topological structures is normally afflicted by large degrees of occlusion. In such a case, addressing visibility issues as early as possible in the design phase, could result in clear and expressive visualizations that would allow the users to study the topological aspects of fluid flows. By investigating disparate flow visualization scenarios, we would be able to identify and describe the benefits induced by the visibility-oriented design, and the related limitations as well.

Another prospective research direction would be to extend our study to other subfields of visualization. Visibility issues are in fact a common problem in visualization. The advocated design principle can be directly applied to the design of visualization techniques for, e.g., medical, biological and geological data. The basic idea can be also exploited in the context of information visualization, where the spatial extent of the phenomenon of interest, if any, is treated just as one of the many variables. In all these situations, we need to determine if a visibility-oriented design can actually lead to successful visualizations, and what are the related pros and cons.