

Illustrative Flow Visualization of 4D PC-MRI Blood Flow and CFD Data

Von der Fakultät für Mathematik und Informatik
der Universität Leipzig
angenommene

D I S S E R T A T I O N

zur Erlangung des akademischen Grades

DOCTOR RERUM NATURALIUM
(Dr. rer. nat.)

im Fachgebiet

INFORMATIK

vorgelegt

von Dipl.-Inf. (Bioinf.) Silvia Born

geboren am 18.05.1980 in Roth

Die Annahme der Dissertation haben empfohlen:

1. Prof. Dr. Kenneth I. Joy (UC Davis, USA)
2. Prof. Dr. Gerek Scheuermann (Universität Leipzig)

Die Verleihung des akademischen Grades erfolgt auf Beschluss des Rates der Fakultät für Mathematik und Informatik vom 16.6.2014 mit dem Gesamtprädikat *summa cum laude*.

Zusammenfassung

Das zentrale Thema dieser Dissertation ist die Anwendung illustrativer Methoden auf zwei bisher ungelöste Probleme der Strömungsvisualisierung. Das Ziel der Strömungsvisualisierung ist die Bereitstellung von Software, die Experten beim Auswerten ihrer Strömungsdaten und damit beim Erkenntnisgewinn unterstützt. Bei der illustrativen Visualisierung handelt es sich um einen Zweig der Visualisierung, der sich an der künstlerischen Arbeit von Illustratoren orientiert. Letztere sind darauf spezialisiert komplizierte Zusammenhänge verständlich und ansprechend zu vermitteln. Die angewendeten Techniken werden in der illustrativen Visualisierung auf reale Daten übertragen, um die Effektivität der Darstellung zu erhöhen.

Das erste Problem, das im Rahmen dieser Dissertation bearbeitet wurde, ist die eingeschränkte Verständlichkeit von komplexen Stromflächen. Selbstverdeckungen oder Aufrollungen behindern die Form- und Strömungswahrnehmung und machen diese Flächen gerade in interessanten Strömungssituationen wenig nützlich. Auf Basis von handgezeichneten Strömungsdarstellungen haben wir ein Flächenrendering entwickelt, das Silhouetten, nicht-photorealistische Beleuchtung und illustrative Stromlinien verwendet. Interaktive Flächenschnitte erlauben die Exploration der Flächen und der Strömungen, die sie repräsentieren. Angewendet auf verschiedene Stromflächen ließ sich zeigen, dass die Methoden die Verständlichkeit erhöhen, v.a. in Bereichen komplexer Strömung mit Aufwicklungen oder Singularitäten.

Das zweite Problem ist die Strömungsanalyse des Blutes aus 4D PC-MRI-Daten. An diese relativ neue Datenmodalität werden hohe Erwartungen für die Erforschung und Behandlung kardiovaskulärer Krankheiten geknüpft, da sie erstmals ein dreidimensionales, zeitlich aufgelöstes Abbild der Hämodynamik liefert. Bisher werden 4D PC-MRI-Daten meist mit Werkzeugen der klassischen Strömungsvisualisierung verarbeitet. Diese werden den besonderen Ansprüchen der medizinischen Anwender jedoch nicht gerecht, die in kurzer Zeit eine übersichtliche Darstellung der relevanten Strömungsaspekte erhalten möchten. Wir haben ein Werkzeug zur visuellen Analyse der Blutströmung entwickelt, welches eine einfache Detektion von markanten Strömungsmustern erlaubt, wie z.B. Jets, Wirbel oder Bereiche mit hoher Blutverweildauer. Die Grundidee ist hierbei aus vorberechneten Integrallinien mit Hilfe speziell definierter Linienprädikate die relevanten, d.h. am gefragten Strömungsmuster, beteiligten Linien ausgewählt werden. Um eine intuitive Darstellung der Resultate zu erreichen, haben wir uns von Blutflußillustrationen inspirieren lassen und präsentieren eine abstrakte Linienbündel- und Wirbeldarstellung. Die Linienprädikatmethode sowie die abstrakte Darstellung der Strömungsmuster wurden an 4D PC-MRI-Daten von gesunden und pathologischen Aorten- und Herzdaten erfolgreich getestet. Auch die Evaluierung durch Experten zeigt die Nützlichkeit der Methode und ihr Potential für den Einsatz in der Forschung und der Klinik.

Abstract

This thesis' central theme is the use of illustrative methods to solve flow visualization problems. The goal of flow visualization is to provide users with software tools supporting them analyzing and extracting knowledge from their fluid dynamics data. This fluid dynamics data is produced in large amounts by simulations or measurements to answer diverse questions in application fields like engineering or medicine. This thesis deals with two unsolved problems in flow visualization and tackles them with methods of illustrative visualization. The latter is a subbranch of visualization whose methods are inspired by the art work of professional illustrators. They are specialized in the comprehensible and esthetic representation of complex knowledge. With illustrative visualization, their techniques are applied to real data to enhance their representation.

The first problem dealt with in this thesis is the limited shape and flow perception of complex stream surfaces. Self-occlusion and wrap-ups hinder their effective use in the most interesting flow situations. On the basis of hand-drawn flow illustrations, a surface rendering method was designed that uses silhouettes, non-photorealistic shading, and illustrative surface stream lines. Additionally, geometrical and flow-based surface cuts allow the user an interactive exploration of the surface and the flow it represents. By applying this illustrative technique to various stream surfaces and collecting expert feedback, we could show that the comprehensibility of the stream surfaces was enhanced – especially in complex areas with surface wrap-ups and singularities.

The second problem tackled in this thesis is the analysis of blood flow from 4D PC-MRI data. From this rather young data modality, medical experts expect many advances in the research of cardiovascular diseases because it delivers a three-dimensional and time-resolved image of the hemodynamics. However, 4D PC-MRI data are mainly processed with standard flow visualization tools, which do not fulfill the requirements of medical users. They need a quick and easy-to-understand display of the relevant blood flow aspects. We developed a tool for the visual analysis of blood flow that allows a fast detection of distinctive flow patterns, such as high-velocity jets, vortices, or areas with high residence times. The basic idea is to precalculate integral lines and use specifically designed line predicates to select and display only lines involved in the pattern of interest. Traditional blood flow illustrations inspired us to an abstract and comprehensible depiction of the resulting line bundles and vortices. The line predicate method and the illustrative flow pattern representation were successfully tested with 4D PC-MRI data of healthy and pathological aortae and hearts. Also, the feedback of several medical experts confirmed the usefulness of our methods and their capabilities for a future application in the clinical research and routine.

Acknowledgements

Many colleagues and friends have supported me during my thesis – either with professional advice, with data, or simply by making the daily work or conferences more fun. First of all, I would like to thank my two doctoral advisors: Prof. Dirk Bartz for raising my interest in the field of medical and illustrative visualization and Prof. Gerik Scheuermann for not hesitating to step in as my new advisor after Dirk passed away very suddenly in March 2010. I thank Gerik for his professional advice and for always encouraging and motivating me when it was necessary. I doubt that I had finished this thesis without his support.

In this context, I would also like to thank Prof. Jürgen Meixensberger and the ICCAS directorate for giving me the opportunity to continue my research at ICCAS.

When writing an application-oriented thesis, one strongly depends on motivated experts who are willing to give insight into their respective fields – and equally important – who are willing to provide data. I found such collaborators in Prof. Michael Markl (Northwestern University Chicago), Prof. Matthias Gutberlet (Heart Center Leipzig), Dr. Markus Rütten (DLR Göttingen), and Dr. Emil Ladar (radiologist, former ICCAS researcher) who filled some of my domain knowledge gaps and gave valuable and motivating feedback. Many thanks to Prof. Markl for the 4D PC-MRI datasets, to Dr. Markus Rütten for the delta wing and the combustion chamber dataset, and to Wieland Reich for the periodic orbit datasets.

Also, I would like to thank the ICCAS Visual Computing group: Daniela Franz for being my companion when trying to figure out how this research thing works; Albert Pritzkau for being a great person surprising me ever again with new talents and for never turning me down when, e.g., his graphic design skills were needed; Matthias Pfeifle for his constant effort on making our visualization framework better and better and for helping me to solve some tricky programming problems. Further, thanks to the other ICCAS researchers who gave the daily research institute madness a bright side, especially during lunch breaks, social events, Girls’ Day tours or super-spontaneous software demos.

Thanks to the BSV group for their great work on the FAnToM framework and for making VIS conferences even more fun. A special thanks to Alexander Wiebel for his great support during the preparation of the VIS 2010 paper.

Also, I would like to thank the unknown paper reviewers who gave useful comments and advice and sometimes drew my attention to formerly unconsidered aspects of my work. They also contributed to this thesis.

Finally and most importantly, a big thank you to my family and friends for their support in the last years – and to Jens for never (really) losing patience with me and my thesis.

Contents

1	Introduction	1
2	Basics	7
2.1	Vector Fields	7
2.1.1	Integral Lines	8
2.1.2	Integral Surfaces	10
2.1.3	Flow Features and Vector Field Topology	12
2.2	Flow Visualization Techniques	14
2.2.1	Direct Flow Visualization	16
2.2.2	Dense and Texture-Based Flow Visualization	16
2.2.3	Geometric Flow Visualization	17
2.2.4	Feature-Based Flow Visualization	18
2.3	Illustrative Techniques in Flow Visualization	22
2.3.1	Perceptual Enhancement	23
2.3.2	Visibility Management	24
2.3.3	Focus Emphasis	25
2.3.4	Visual Explanation	26
I	Illustrative Visualization of Stream Surfaces	29
3	Illustrative Stream Surfaces	31
3.1	Related Work	32
3.2	Visualization Design	35
3.3	Method Description and Implementation	37
3.3.1	Silhouettes	38
3.3.2	Halftoning	40
3.3.3	Illustrative Surface Stream Lines	41
3.3.4	Movable Cuts and Slabs	43
3.4	Results	45
3.4.1	Delta Wing	45
3.4.2	Periodic Orbit	50
3.4.3	Combustion Chamber	50
3.4.4	Performance	53
3.4.5	Expert Feedback	53
3.4.6	Limitations	54
3.5	Discussion and Future Work	54

II Visualization of 4D PC-MRI Blood Flow Data	55
4 Background	57
4.1 Medical Background	57
4.1.1 Human Cardiovascular System	57
4.1.2 Cardiovascular Diseases	60
4.2 4D Phase-Contrast MRI	62
4.2.1 Introduction to MRI	62
4.2.2 Velocity Measurement with 4D PC-MRI	70
5 Visual Blood Flow Analysis with Line Predicates	77
5.1 Related Work	80
5.2 Concepts	82
5.2.1 Point and Line Predicates	82
5.2.2 Line-based and Derived Predicates	83
5.3 Analysis of Aortic and Cardiac Blood Flow	84
5.3.1 Workflow and Data Preprocessing	84
5.3.2 Line Predicates for Analysis of Steady Blood Flow	88
5.3.3 Line Predicates for Analysis of Unsteady Blood Flow	92
5.3.4 Flow Structure Visualization	94
5.4 Results and Evaluation	95
5.4.1 Aorta Cases	95
5.4.2 Heart Cases	101
5.4.3 Evaluation	110
5.5 Discussion and Future Work	113
6 Illustrative Visualization of 4D PC-MRI Blood Flow	115
6.1 Related Work	117
6.2 Abstract Visualization of Blood Flow Features	118
6.2.1 Design of an Abstract Flow Depiction	119
6.2.2 Finding Representatives	119
6.2.3 Illustrative Rendering of Representatives	122
6.2.4 Illustrative Rendering of Vortices and Context	124
6.3 Results	125
6.4 Discussion and Future Work	130
7 Conclusion and Outlook	133
List of Figures	135
Bibliography	137

Introduction

When asking people for their most important sense, the majority would probably name their vision. Most of the stimuli we receive from our environment are visual and we use about 20% of our brain to process this flood of information [192]. Our visual system is a highly-parallel supercomputer that allows us to efficiently detect colors and patterns, to recognize objects and faces and, eventually, give a meaning to the seen.

These highly-developed visual abilities are taken advantage of in the scientific field of *visualization*. Here, static or interactive visual representations are used to give people an understanding of data and contexts. We come across data visualizations every day when we, for example, analyze weather maps during the daily weather forecast or when we look for the desired bus on the network plan of the public transport system. These visualizations allow us to comprehend complex information a lot faster than by analyzing raw numbers or purely textual explanations [118].

However, visualization is more than these relatively simple everyday examples. In various scientific, engineering or economical fields people struggle with the large amounts of complex data. Visualization provides software tools facilitating the handling and analysis of the data and allows a quicker and easier knowledge gain. Based on the type of data that is processed, different branches of visualization are distinguished: *information visualization* deals with abstract data that is not necessarily spatially ordered (e.g., social media data, financial data, texts); *scientific visualization* is applied to data related to physical phenomena, as they appear in various application fields. One of these fields is fluid dynamics. Its data and research questions are dealt with in the area of *flow visualization*, which is the central topic of this thesis.

In fluid dynamics, flow phenomena related to engineering problems or research questions in the natural sciences are examined. Before the advent of computers, fluid dynamics research was conducted theoretically and in experiments. Back then, researchers visualized flow either by sketching their ideas and insights with pen and paper (see Fig. 1.1a) or by imaging physical experiments (see Fig. 1.1b). With the increasing power of computers, the simulation of flow (computational fluid dynamics, CFD) became a valuable addition to plain theory and experimental research. CFD rendered experiments possible that were previously unfeasible due to, e.g., high risk or high costs. Further, CFD allowed the multiple repetition of experiments with varying parameters. In short, CFD fundamentally changed the work of fluid dynamics experts. However, with the possibility to easily conduct flow simula-

tions, a new challenge arises in analyzing the growing amount of simulation results. Here, (computational) flow visualization steps in and develops tools and methods to help researchers with the handling and analysis of their data. This includes the mere exploration of data, the specific detection of interesting data characteristics or patterns, the confirmation or disproval of existing or the formulation of new hypotheses.

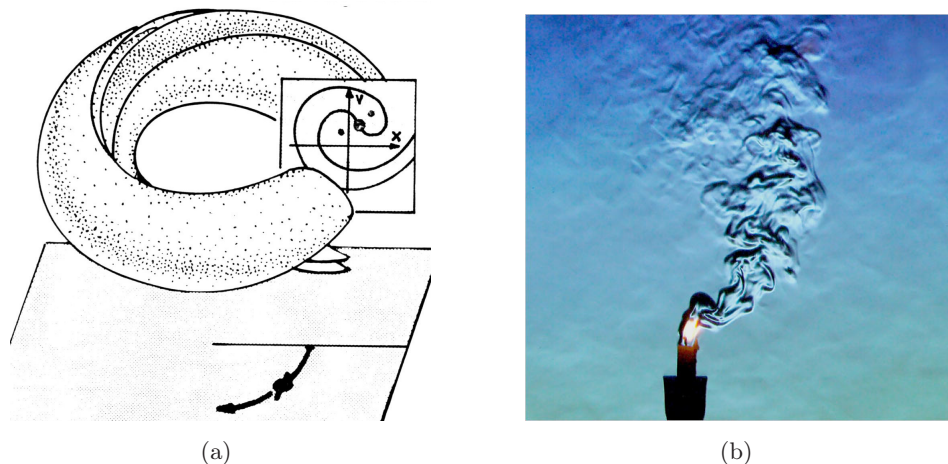


Figure 1.1: Non-computational flow visualization. (a) Sketch of a torus by Abraham and Shaw [1], used with permission by R. Abraham, (b) Schlieren photography of a candle by Settles [157], used with permission by G. Settles.

In the last years, fluid dynamics has proven to be valuable for the medical research of cardiovascular diseases. Besides the simulation of blood flow with CFD, the measurement of an individual's blood flow has become possible with a flow-sensitive MR imaging technique (4D phase-contrast magnetic resonance imaging, 4D PC-MRI). 4D PC-MRI delivers volume datasets containing blood flow velocities in three dimensions over an averaged heart cycle. The availability of these snapshots makes several new applications in diagnostics, therapy, and research conceivable. For example, medical doctors could be put in the position to examine a patient's hemodynamics quantitatively and qualitatively and draw direct conclusions about the particular prognosis and the indicated therapeutical measures. Also, 4D PC-MRI results could be used to enhance the evaluation of current therapies or surgical techniques or even to simulate certain interventions and adapt them to the patient-specific situation. In order to fulfill the vision of using 4D PC-MRI to enhance the treatment of patients with cardiovascular diseases, researchers need to answer several basic questions: How does pathological blood flow differ from healthy blood flow? How do cardiovascular diseases develop and how could this be detected from the 4D PC-MRI measurements? How do specific therapies affect the blood flow situation and how could this be used to predict the risk of a relapse?

One can assume that it is already possible to acquire data containing the necessary information for answering some of these questions. But again, data acquisition

is only the first step. The crucial point is the appropriate analysis and visualization tool, which is needed to elicit the required information from the data, but is not yet at hand. Thus, it is the task of the fluid dynamics and visualization community to develop the analysis methods and with that contribute to an improved understanding and treatment of cardiovascular diseases. This thesis is also meant to help towards this goal.

When developing tools for the medical domain, it is important to consider specific characteristics of this application field. In medical flow research, but especially in the clinical routine, the users might not have an extensive fluid dynamics expertise and are usually pressed for time. Therefore, special demands need to be placed on the method's runtime and robustness, a comprehensible and unambiguous result display, and a simple usability without tedious parameter tuning. So far, standard flow visualization techniques used in technical domains are primarily applied to the medical flow data. However, these do not fulfill the demands mentioned above. It is time-consuming to explore and gain overview over a 4D PC-MRI dataset. Furthermore, the methods depend on the users' fluid dynamics expertise, which is required to generate useful results. An automatic analysis is also not yet foreseeable, but aspired for larger studies and the clinical routine [110].

For this reason, we developed an interactive tool for the visual analysis of 4D PC-MRI data that gives a quick overview over the main flow characteristics. Based on precalculated integral lines and detected vortices, the users can interactively explore the blood flow by selecting integral lines according to properties of interest, such as specific velocities, flow paths, or residence times. With the chosen global perspective, the user can more quickly examine the dataset and identify flow specialties or anomalies. We show this by presenting several case studies from aortic and cardiac blood flow and expert feedback.

The identified flow patterns of interest are then displayed in a style that was motivated by hand-crafted illustrations. Illustrations, as they can be found in medical textbooks, are designed to be easy to understand. The applied drawing and abstraction techniques find their way into software visualization tools (illustrative paradigm) and here help to improve comprehensibility. In our case, the illustrative rendering of flow patterns gives rise to uncluttered and clear images informing about the blood flow situation.

The illustrative paradigm can also be applied to more general flow visualization problems not necessarily being inside the medical domain. For example, integral surfaces, and more specifically stream surfaces, are intuitive and established tools to visualize flow. However, as soon as the flow becomes more complex, self-occlusions occur and the surface's comprehensibility is reduced. As a result, stream surfaces become useless in the most interesting flow situations.

So, we tackle this perceptual problem with illustrative methods as well. Textbooks and theses that were produced in the pre-computational fluid dynamics era, contain very simple sketches of very complex flow phenomena. We applied the used drawing methods to the rendering of computational stream surface models and enhanced these methods towards interactive tools by utilizing the potentials of

computer graphics. Thus, we could achieve an improved stream surface rendering and could show that illustrative methods can reduce the perceptual problems in this context. Especially, interactivity, which is not possible with static textbook illustrations, allows users to explore and better understand the flow represented by the stream surface.

In summary, the central topic of this thesis is the development of methods facilitating users the access to the knowledge contained in their medical and non-medical fluid dynamics data. The suggested interactive methods for visual analysis make powerful use of illustrative techniques and with that improve the insight into the data and the knowledge gained in the respective application domains. The detailed contributions are listed in the specific chapters presenting the different methods. Yet, this thesis' main contributions are:

- the improved comprehensibility of complex stream surfaces by applying artistic and illustrative rendering techniques and enhancing these methods towards interactive focus & context-methods like movable, intersecting cuts. These tools allow the user-driven examination of the flow represented by the surface and gives insight into the surface's hidden features.
- the development of a visual analysis method for 4D PC-MRI blood flow data of the aorta and the heart. Based on precalculated integral lines and a toolbox of line predicates, this method allows the user to identify the main flow patterns based on properties like vortices, flow paths, velocity, or residence time. The method serves as an interactive query tool for medical experts, but may also be advanced towards an automatic analysis method. The evaluation showed that with this method flow aspects in healthy and pathological datasets could easily be spotted. These aspects have previously been detectable only with a lot of effort – if at all.
- the abstract depiction of vortex regions and flow features captured by the previous method to support non-flow experts with blood flow interpretation.

The remainder of this thesis is structured as follows. The thesis continues in Chapter 2 with foundations relevant for all topics of this thesis. It covers an introduction to the mathematical basics of vector fields and gives an overview of the methods of flow visualization and illustrative visualization. The remainder of the thesis is divided into two main parts. Part I deals with the illustrative rendering of stream surfaces extracted from non-medical flow data. Part II addresses the work on the analysis and visualization of medical 4D PC-MRI blood flow data. This part begins with Chapter 4, providing some more specific background about the cardiovascular system and its diseases as well as the acquisition of 4D PC-MRI data. In Chapter 5, the work on the visual analysis of aortic and cardiac blood flow with line predicates is presented. In Chapter 6, the previous analysis results are transferred into abstract and illustrative depictions. Chapter 7 concludes the thesis and gives various suggestions for future work.

The presented work has been published in peer-reviewed international conference proceedings and journals:

BORN, S., WIEBEL, A., FRIEDRICH, J., SCHEUERMANN, G., AND BARTZ, D. Illustrative Stream Surfaces. *IEEE TVCG* 16, 6 (2010), 1329–1338. (Honorable Mention at IEEE Visualization 2010)

BORN, S., PFEIFLE, M., MARKL, M., AND SCHEUERMANN, G. Visual 4D MRI Blood Flow Analysis with Line Predicates. In *Proc. of IEEE Pacific Visualization* (2012), pp. 105 – 112. (Best Paper Award)

BORN, S., PFEIFLE, M., MARKL, M., GUTBERLET, M., AND SCHEUERMANN, G. Visual Analysis of Cardiac 4D PC-MRI Blood Flow Using Line Predicates. *IEEE TVCG* 19, 6 (2013), 900-912.

BORN, S., MARKL, M., AND GUTBERLET, M., AND SCHEUERMANN, G. Illustrative Visualization of Cardiac and Aortic Blood Flow from 4D MRI Data. In *Proc. of IEEE Pacific Visualization* (2013), pp. 129-136.

This thesis addresses two distinct topics in the fields of illustrative and flow visualization. The first topic comprises the improvement of the perceivability of complex stream surfaces as they appear in various flow research areas. The second topic is the analysis and illustrative depiction of medical blood flow (4D PC-MRI). In this chapter, foundations relating to both areas are presented. More specifically, vector fields and the related mathematical concepts are introduced and an overview over main flow visualization techniques is given. Further, the basic idea and approaches of illustrative visualization are presented by means of illustrative techniques tackling flow research problems. This shows that these two areas already overlap to a certain extent. More specific work related to the new methods presented in this thesis will be discussed in the respective chapters.

2.1 Vector Fields

The mathematical basics about vector fields, integral structures, and vector field topology are taken from Marsden et al. [111] and Asimov [5]. For more details, we refer to these publications.

According to Asimov, “a vector field arises in a situation where, for some reason, there is a direction and a magnitude assigned to each point in space” ([5], p.1). More mathematically put, a vector field $v : D \rightarrow \mathbb{R}^d$ is a map that assigns a vector to each point in a spatial d -dimensional domain $D \subset \mathbb{R}^d$ (usually, $d = 2$ or $d = 3$). This definition describes a steady flow. An unsteady flow can be represented by a time-dependent vector field where a vector is assigned to every spatial point for every time point: $v : D \times T \rightarrow \mathbb{R}^d$ with the time span $T = [t_0, t_n]$.

The main characters of flow visualization, as discussed in this thesis, are velocity vector fields in three-dimensional space. For the remainder of this thesis, we therefore consider vector fields in the spatial domain $D \subset \mathbb{R}^3$ and with vectors representing velocity $v = (u, v, w)$ at the respective points in D . The term *vector field* is used as synonym for *velocity vector field*. Further, we will assume that all vector fields are at least Lipschitz-continuous.

A vector field is *unsteady* if the velocities of the described flow change over time. In a *steady* velocity vector field the velocities remain constant. This is not only the case in steady flow but also when considering one isolated time point of an unsteady velocity vector field. One can think of a time-dependent vector field as a movie where each frame is a steady (i.e., non-time-dependent) vector field [5].

The spatial domain D of the vector field is usually a grid. The most trivial grids are regular, rectilinear, and Cartesian grids, which allow fast traversal and access, and efficient calculations of derived vector field quantities. Blood flow measurements with 4D PC-MRI (see Sec. 4.2) are given with respect to regular or even Cartesian grids. For CFD simulations structured curvilinear or unstructured grids (composed of tetrahedra or hexahedra or a mixture) are often used. These grids are necessary to account for the specific geometry in or around which the flow is simulated. Further, the size and type of polyhedra can be adapted, so that areas are well-represented where interesting flow phenomena are expected (e.g., at the borders). Point location and the computation of derived data are less efficient with these complex grids [92].

The gradient of a vector field v is given by its *Jacobian* J , which is the field's first derivative matrix:

$$J = \nabla v = \begin{pmatrix} \frac{\delta u}{\delta x} & \frac{\delta u}{\delta y} & \frac{\delta u}{\delta z} \\ \frac{\delta v}{\delta x} & \frac{\delta v}{\delta y} & \frac{\delta v}{\delta z} \\ \frac{\delta w}{\delta x} & \frac{\delta w}{\delta y} & \frac{\delta w}{\delta z} \end{pmatrix} \quad (2.1)$$

It can be decomposed into a symmetric matrix $S = \frac{1}{2}(J + J^T)$ and an anti-symmetric matrix $\Omega = \frac{1}{2}(J - J^T)$. The velocity gradient tensor ∇v is oftentimes used as a synonym for the Jacobian.

The Jacobian is used in topological analysis and several feature extraction techniques, since it allows to detect and classify singularities or to derive necessary field quantities, such as divergence, vorticity, helicity, or acceleration. Vorticity ω describes the local curl or flow rotation of the vector field: $\omega = \nabla \times v$. It is itself a vector that describes the flow rotation in speed and direction. Helicity is computed by projecting vorticity onto the local velocity vector: $\frac{\omega v}{|\omega||v|}$. Helicity is therefore the cosine between vorticity and velocity vector and is a scalar value. These properties are of relevance, e.g., for vortex detection (see Sec. 2.2.4 and 5.3.1).

2.1.1 Integral Lines

An autonomous, ordinary differential equation (autonomous ODE) defined on $D \subset \mathbb{R}^3$ is an equation of the form

$$\frac{dS}{dt} = v(S) \quad (2.2)$$

where $S : B \rightarrow D$ is a curve parametrized by some interval B , and $v : D \rightarrow \mathbb{R}^3$ is C^1 . For an initial condition $S(0) = a$ with $a \in D$, there exists a solution or curve S of the ODE with $S'(t) = v(S(t))$ for each t in B . This solution is unique since every curve $R : B \rightarrow D$ that fulfills the initial condition $R(0) = a$ and the equation $R'(t) = v(R(t))$ is the same as the curve S [5].

If v is a steady velocity vector field as defined earlier, the curve S is the course of an infinitesimally small and massless particle seeded into the steady vector field at position $a \in D$. Further, the curve S is tangential everywhere to the vector field. We call such curves *stream lines*.

For a non-autonomous ODE of the form

$$\frac{dP}{dt} = v(P, t) \quad (2.3)$$

$P : D \rightarrow \mathbb{R}^3$ again denotes a curve parametrized by the interval B . In contrast to the previous autonomous ODE, the function v takes time $t \in T$ as second argument: $v : D \times T \rightarrow \mathbb{R}^3$. For the initial condition $P(\tau) = a$ with $a \in D$, there exists the solution $P'(t) = v(P(t), t)$ [5].

If v is now a time-dependent vector field, the curve P depicts the course of an infinitesimally small and massless particle that is emitted into the field at starting time $\tau \in T$ and starting point $a \in D$. Again, the curve runs tangentially to the velocity vectors in v . We call these curves *path lines*.

In a real-world example like a river, a stream line represents the course of a particle emitted into a constantly flowing river at a specific position. A path line describes the path of a particle emitted in a changing river (e.g., a weir has been opened) at a specific time point and position.

For a steady vector field, the definitions of stream lines and path lines are identical. Accordingly, stream lines are equivalent to path lines with time kept fixed ($t = t_i$). Also, stream lines are determined fully in the spatial domain and independent of time. To emphasize that the stream line parameter is not equivalent to time, we denote it with s .

To sum this up, the following formal definitions of path lines and stream lines are used in the remaining thesis. In an unsteady vector field $v : D \times T \rightarrow \mathbb{R}^3$, a path line P is defined as

$$\begin{aligned} P_{a,\tau} : T_{a,\tau} &\rightarrow D \\ P_{a,\tau}(\tau) &= a \\ t &\mapsto P_{a,\tau}(t) \\ \frac{dP_{a,\tau}}{dt}(t) &= v(P_{a,\tau}(t), t) \end{aligned} \quad (2.4)$$

for starting position $a \in D$, starting time $\tau \in T$ and integrated over time $t \in T$. The time interval $T_{a,\tau} \subset T$ is the maximal lifespan of the particle in D during T . Two path lines are equivalent if they describe the path of the same particle.

Stream lines depict the trace of the virtual particle in a *steady* vector field $v : D \rightarrow \mathbb{R}^3$:

$$\begin{aligned} S_{a,t_i} : B_a &\rightarrow D \\ S_{a,t_i}(0) &= a \\ s &\mapsto S_{a,t_i}(s) \\ \frac{dS_{a,t_i}}{ds}(s) &= v(S_{a,t_i}(s), t_i) \end{aligned} \quad (2.5)$$

$B_a \subset B$ is the maximal parameter interval of the stream line in D at a fixed time $t = t_i$. As a consequence of the previously described uniqueness of the ODE solution, stream lines cannot intersect and two stream lines are equivalent if they run through the same position x .

Apart from path lines and stream lines, two further features lines exist in a time-dependent vector field: streak lines and time lines.

Path lines and stream lines can be simply derived by integration since they describe the traces of single particles. In contrast, streak lines and time lines describe the course of several particles through an unsteady vector field. *Streak lines* represent particles that were seeded into the flow at the same position a , but at different time points τ . In the real world, a streak line occurs when dripping, e.g., color in a river at the same position for several time points. For a steady vector field, streak lines and stream lines (and with that also path lines) are identical. *Time lines* however describe a line of simultaneously seeded particles at a specific point in time and can be described as the advancing front of neighboring path lines. In contrast to streak lines, the particles of a time line do not originate from a common position but from the same time point. In the real world, a thread floating in a river represents a time line (see Fig. 2.1 for an overview).

Interestingly, path lines and stream lines are commonly used in computational flow visualization because they are rather easy to compute. In real-world experiments however it is hard to follow the path of single particles. Hence, streak lines and time lines are more prevailing in these setups.

Weinkauff et al. have recently presented a way to generate streak lines as tangent curves of a derived vector field, which they call streak line vector field [198]. Further, they introduced a general scheme to formulate all four types of feature curves as tangent curves [196].

Since the vector field v is usually given in discrete form (i.e., it is not continuously differentiable), integral lines are generated with numerical integration techniques, such as the fourth-order Runge Kutta integrator. On this topic and the computation of feature curves in general, a large amount of literature exists. More details are given in the state-of-the-art report by McLoughlin et al. [113].

Periodic Orbit A stream line may visit the same position $x \in D$ more than once: $S_{a,t_i}(s_1) = S_{a,t_i}(s_2) = x$. If further $v(x) \neq 0$ and $S_{a,t_i}(s) = S_{a,t_i}(s + kT)$ with $k \in \mathbb{Z}$ and $T \in B_a$, then this stream line is periodic and called *periodic orbit*. A stream line that does not visit a position more than once is called *regular*. Poincaré maps are tools to characterize different types of periodic orbits [131]. For this, a two-dimensional plane is positioned in a three-dimensional flow so that the periodic orbit intersects it at its center. The map is constructed by tracking consecutive intersections of non-periodic stream lines with the plane in the vicinity of the periodic orbit. The evolution of these intersections allows to characterize the periodic orbits as (spiraling) sources and sinks, (twisted) saddles etc. (see Fig. 2.2 for an overview).

2.1.2 Integral Surfaces

Analogously to the feature curves, four types of feature surfaces exist for an unsteady vector field: path surfaces, stream surfaces, streak surfaces, and time surfaces. For

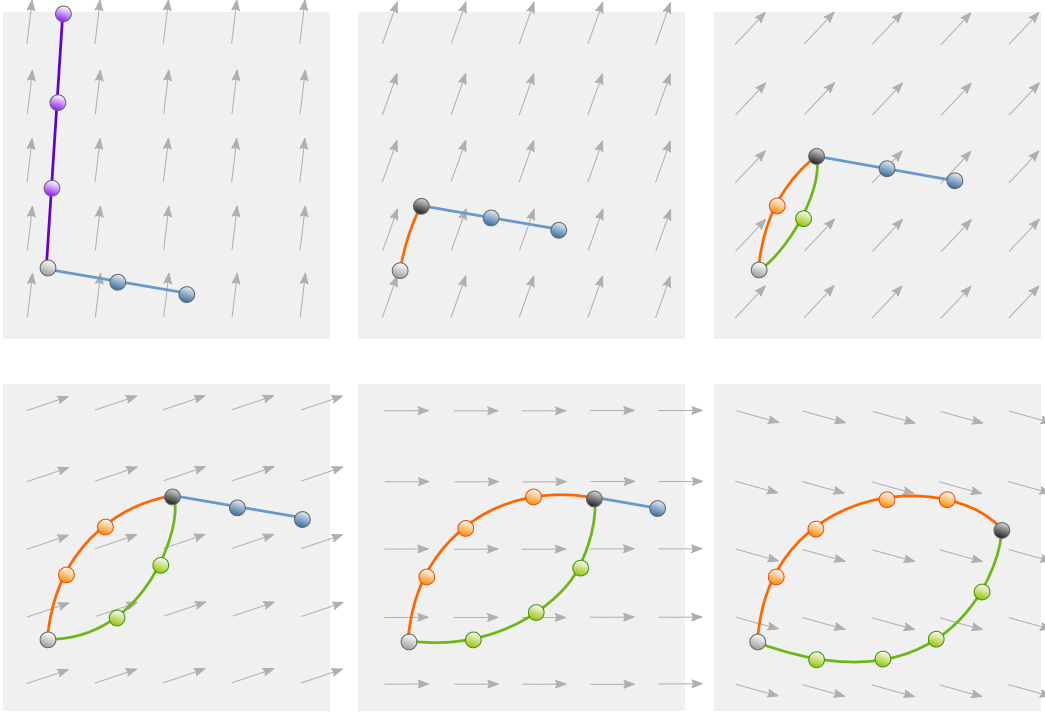


Figure 2.1: Different types of feature curves and their evolution over six successive timesteps of an unsteady vector field (from top left to bottom right). Stream line (purple), path line (orange), and streak line (green) are seeded at the position marked with a light gray sphere. The dark gray sphere represents the current position of the particle (which coincides for path line and streak line). The path line depicts the course of one particle seeded into the vector field at the first timestep. The streak line connects particles that are emitted into the vector field at successive timesteps. The blue line represents a time line.

this thesis only stream surfaces are of relevance. However, the others are also explained shortly.

A *stream surface* is a continuum of stream lines or, in other words, the locus of a set of stream lines seeded from a common curve [62]. Therefore, a stream surface is also tangential everywhere to the underlying vector field (see Fig. 2.3).

Just like stream surfaces being an extension of stream lines, *path surfaces* extend the concept of path lines. Instead of representing the trace of a single particle from one position in space, the path surface depicts the course of several particles seeded from a three-dimensional curve. More specifically, the particles are emitted at a fixed point in time and their course is integrated over time. The path surface therefore shows the evolution of a particle line in unsteady flow and contains all positions of the particles during their lifetime [114].

Time surfaces on the other hand display the joint flow traversal of particles located on a surface, similar to a bed sheet fluttering in the wind [87].

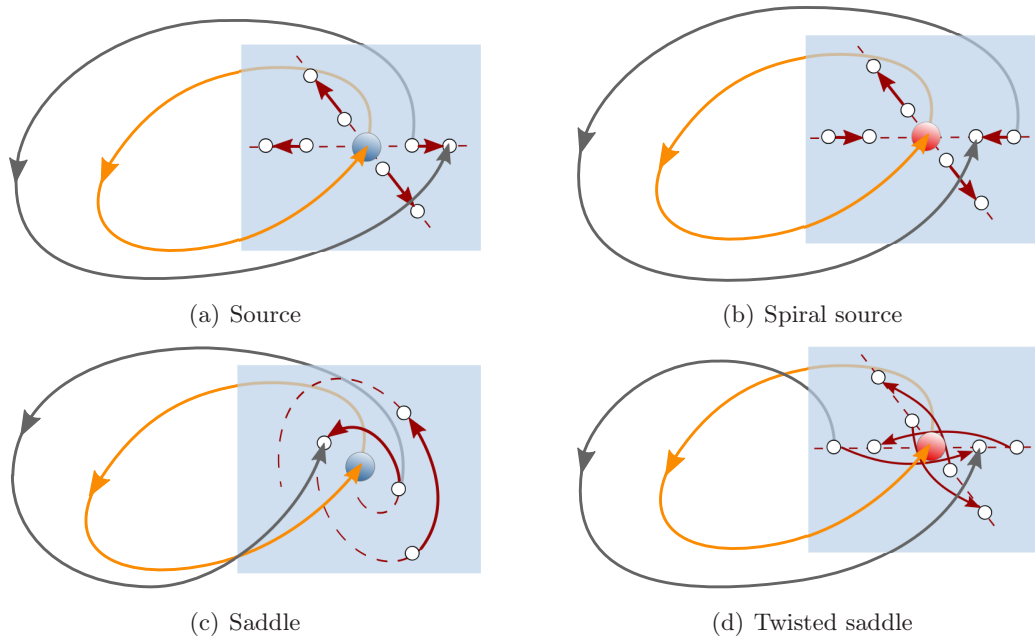


Figure 2.2: Different types of periodic orbits (orange) can be distinguished based on the behavior of neighboring stream lines (gray). This is made visible with the help of Poincaré maps (blue planes). Figure redrawn from Peikert and Sadlo [128].

Finally, a *streak surface* shows aspects of both path surfaces and time surfaces. From a seeding curve, particles are added continually to the streak surface. With that, the surface grows and evolves over time [87].

A more detailed introduction to time and streak surfaces and their efficient computation is given by Krishnan et al. [87].

2.1.3 Flow Features and Vector Field Topology

The term *flow feature* is not closely defined, but describes mainly a physically meaningful pattern in a vector field. What is considered meaningful depends on the specific research question. For hemodynamics research, interesting flow features are, e.g., areas with recirculations, abnormal vortical behaviour, or a high residence time - as these are valuable for the diagnosis or prognosis of diverse pathologies of the cardiovascular system. In aircraft design, engineers investigate flow features giving information about the flight characteristics of an airplane. Examples are wakes, vortices, and vortex breakdowns.

When compared to integral lines or surfaces, flow features provide higher-level information about the dataset. They allow the user to draw conclusions by taking a look at a very reduced variant of the dataset. Therefore, by extracting the relevant flow features from a dataset, very large amounts of data become easier to manage and analyze, too.

The extraction and visualization of flow features depends very much on the

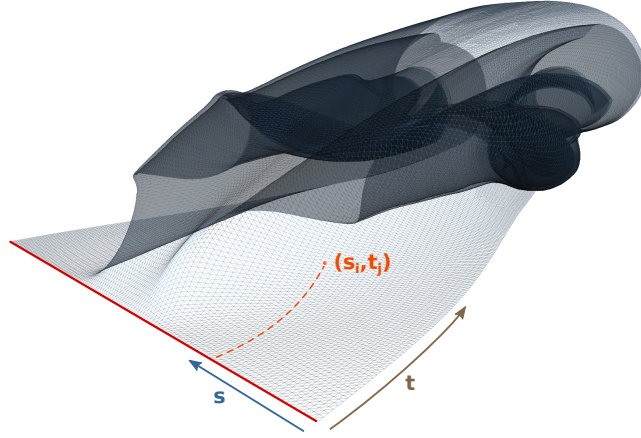


Figure 2.3: An example stream surface showing the flow around a delta wing. The seed line is depicted in red and parameterized with s . The parameter t increases along the flow direction.

specific feature and is the topic of Section 2.2.4. However, vector field topology plays an important role in this context since on one hand the topological structures may themselves be flow features and on the other hand topological algorithms can be used to extract other flow features as well. Vector field topology will be shortly introduced now along the lines of Asimov [5] and Helmann & Hessselink [58].

Vector field topology deals with the structure of the vector field, i.e., the subdivision of a vector field into areas with similar flow behavior. This subdivision is given by the critical points of a vector field and the surfaces (or lines for the 2D case) connecting them. Remarkable about these surfaces is that they divert integral lines originating from or ending up in substantially different regions of the spatial domain.

Critical points describe points where the vector magnitude is zero (i.e., no flow occurs, stationary points) and where the Jacobian is regular. Critical points can be classified as sources, sinks, or saddle points depending on the flow in their surroundings. Investigating the flow around these points is done with eigenvector analysis of the Jacobian [58]. In the 2D case, the Jacobian can have either two real eigenvectors or the eigenvectors are complex conjugates of each other. A real eigenvector determines the direction to which the field flows parallel or, in other words, a real eigenvector is tangent to the integral lines in the vicinity of the critical point. The sign of the corresponding eigenvalue indicates whether the flow is parallel or anti-parallel to the real eigenvector. With that, a sink is determined by two negative eigenvalues, a source by two positive eigenvalues, and a saddle point by eigenvalues with opposite sign. The imaginary part of an eigenvalue indicates circulatory flow

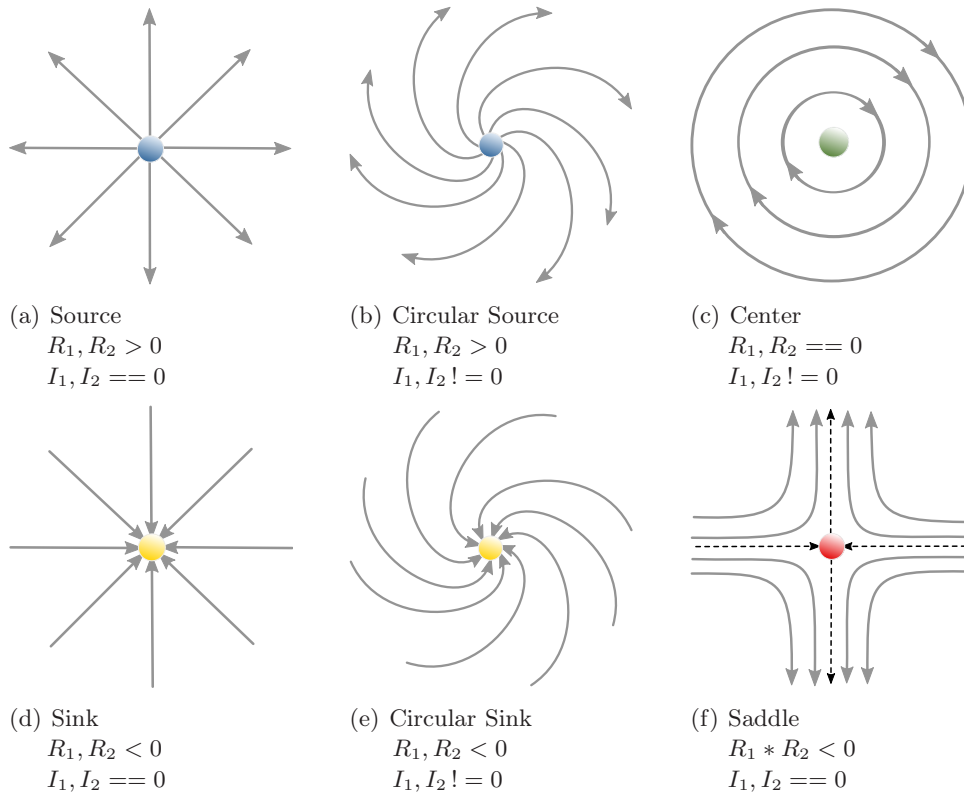


Figure 2.4: Different types of critical points and their respective combinations of real (R_1, R_2) and complex eigenvalues (I_1, I_2). Image based on [58].

around the critical point (see Fig. 2.4).

In the 3D case, the eigenvector analysis is equally straight-forward although the flow behavior around the critical points is slightly more complicated. The Jacobian can have either three real eigenvectors or one real eigenvector and one complex conjugate pair. If all eigenvectors are real, the signs of the corresponding eigenvalues classify the critical point as sources, sinks, and saddle points again. With the saddle point, the flow can either flow in two dimensions towards and in one dimension away from it, or vice versa. A circulating flow is indicated in case not all eigenvectors are real. A summary of the different combinations is given in Figure 2.5. More details about vector field topology are given by Helman and Hesselink [58] and Asimov [5].

2.2 Flow Visualization Techniques

In the previous section vector fields and their various derived representations have been presented. The latter range from simpler geometrical lines or surfaces depicting the course of one or several particles to more sophisticated, mostly physically motivated flow features. These representations differ in the type of information they contain about the original vector field and also in the level of abstraction they apply

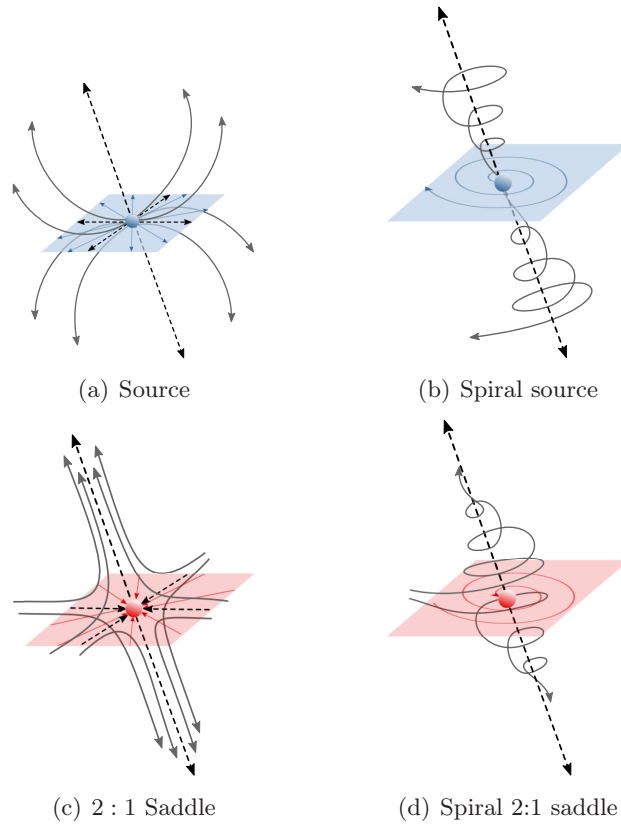


Figure 2.5: Different configurations of critical points in 3D. All other variants can be shown by inversing time. The Jacobian has either 3 real eigenvalues or a real eigenvalue R and a complex conjugate pair $K \pm Li$. (a) Source, i.e., three positive real eigenvalues. (b) Spiral source, i.e., $R > 0$ and $K > 0$. (c) 2:1 saddle, i.e., one positive and two negative real eigenvalues. (d) Spiral 2:1 saddle, i.e., $R < 0$ and $K > 0$. Redrawn from Peikert and Sadlo [128].

to it. With that, they can be thought of as the different levels of a data abstraction hierarchy ranging from the raw data at the bottom to flow features or more advanced findings on the top. On the way upwards, specific characteristics or aspects of the vector fields are extracted while neglecting the remaining data. The amount of data is therefore reduced from bottom to top and the information becomes more concrete.

Visualizing a vector field means to select data at the appropriate abstraction level and to display it such that the users can answer their questions or gain insight. Related to this approach, Post et al. suggested to categorize visualization methods based on the data abstraction level they apply [132]. This lead to four main types of flow visualization techniques: direct, texture-based, geometric and feature-based. In the following, these four groups are further explained. The level of data abstraction increases with each group.

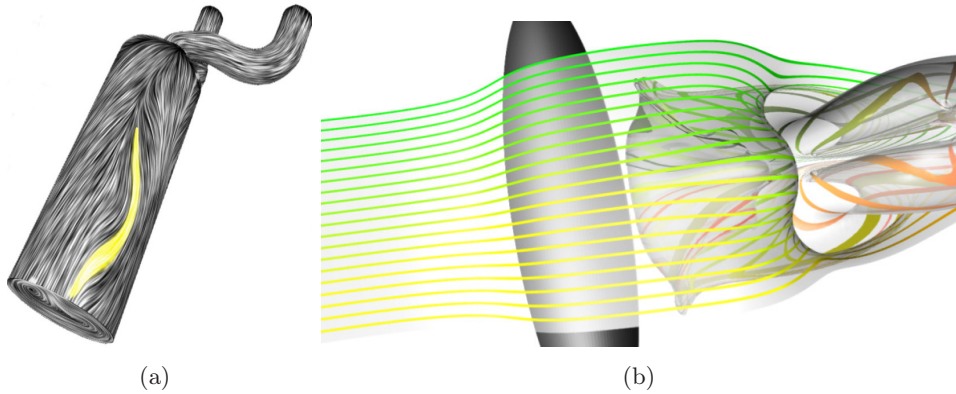


Figure 2.6: Examples for texture-based and geometric flow visualizations. (a) GPU-FLIC on the curved surface of an engine cylinder by Li et al. [97], © 2008 IEEE. (b) Streak surface with streak ribbon texture by Krishnan et al. [87], © 2009 IEEE.

2.2.1 Direct Flow Visualization

The most straight-forward way of visualizing flow is to display the vector information directly and without extensive preprocessing. Examples are volume rendering methods or hedgehog techniques where the single vectors of the field are drawn as arrows or glyphs. These techniques are rather easy to compute and provide users with an immediate and detailed depiction of the data. In the two-dimensional case, this usually leads to comprehensible visualizations. However, in the three-dimensional case, visual clutter and insufficient depth perception hinder overview in many cases.

2.2.2 Dense and Texture-Based Flow Visualization

This category's methods compute dense flow visualizations which provide lots of detail, especially in areas with complex flow patterns (e.g., critical points). For this, textures are computed that resemble the flow direction of the local vector field at every point. Laramée et al. published a state-of-the-art report on dense and texture-based flow visualization [92]. In this report, they divided these methods into spot noise, line integral convolution (LIC), and texture advection techniques [92] – depending on the primitives that are used for texture generation.

Spot noise was introduced to the flow visualization literature by van Wijk [183]. These techniques randomly distribute simple primitives (like, e.g., ellipses) on the texture and deform them according to the underlying vector field. With that, the so-called spots turn into short streaks and resemble the local flow field direction. Further, the degree of smearing gives a hint on the velocity magnitude.

Line integral convolution (LIC) methods operate on noise textures that are convolved with a kernel filter representing the input vector field. The result is a texture smeared along the direction the vector field's stream lines. In comparison to spot noise methods, LIC reflects only velocity direction not magnitude, but is superior when it comes to showing local high curvature and critical points of the vector field.

The idea of LIC was introduced by Cabral and Leedom in 1993 [21].

Texture advection methods use moving texels or texture-mapped polygons as their primitives. These primitives are advected in image space to resemble flow direction. Image-based flow visualization (IBFV) by van Wijk is the pioneer method in this area [184].

All these methods were originally designed for steady, two-dimensional vector fields where they have been proven to be very useful [90]. However, a generalization to three-dimensional vector fields is not trivial and has mainly been done by applying dense, texture-based visualizations to (boundary) surfaces in the 3D domain [94, 97, 185] (see Fig. 2.6a).

A direct application of LIC and texture advection to 3D vector fields has been done, e.g., by Interrante and Grosch [64], Rezk-Salama et al. [138], and Telea and van Wijk [171]. To overcome perceptual problems of visual clutter and low depth perception, they applied halos [64], clipping planes [138], and linked high velocity magnitudes to opacity [64, 138].

More complex and advanced methods and further methods that belong to neither of these categories can be found in [92].

2.2.3 Geometric Flow Visualization

Geometric algorithms represent a vector field with discrete, geometric objects whose course, position, and orientation resemble the vector field's underlying flow properties. These geometric objects are particles, feature curves (stream lines, streak lines, path lines, time lines; see Sec. 2.1.1) and integral surfaces (see Sec. 2.1.2) that are all derived by integration of the velocity vector field. Therefore, these methods are also called *integration-based*. McLoughlin discussed the state-of-the-art of this research area in 2010 [114].

Particle traces, feature curves, and integral surfaces are per se very comprehensible visualization tools. Still, geometric visualization, especially for 3D vector fields, is not trivial as several challenges arise, which are presented in the following (based on [114]).

Line Integration and Surface Construction One challenge is the efficient computation of these geometric objects and a large amount of literature concentrates on the acceleration of numerical line integration on complex grids [178] and by utilizing graphics hardware [82, 88].

Stream surface construction was introduced to the visualization literature by Hultquist [62]. He presented an algorithm that approximates stream surfaces with a triangulation of a certain number of stream lines. This approximation can be refined or coarsened at converging or diverging stream lines by adaptively removing or inserting stream lines respectively. Since the introduction of the algorithm more than twenty years ago, a manifold body of research on this subject has achieved substantial improvements concerning adaptivity [43, 44], accuracy [43, 152, 155],

topological correctness [122, 128, 154, 163], performance [150], and adaption to time-varying datasets, i.e. time, streak, and path surfaces [43, 87, 150] (see Fig. 2.6b).

Seeding and Placement Another problem is to pursue an *appropriate seeding strategy*, i.e., where to place the geometric objects and how many. The seeding strategy determines the object distribution and controls which flow patterns are visualized and which are neglected. Simply showing a large number of objects decreases the risk of missing interesting flow areas, but also entails higher visual complexity - especially in three-dimensional and unsteady flow. So, a large number of literature in this area deals with the clever object placement to achieve maximal insight. For 2D vector fields, the most established line seeding methods were introduced by Turks and Banks [177], and Jobard and Lefer [73] respectively. Turks and Banks introduced a seeding method that distributes stream lines evenly in image-space [177]. This method was enhanced with respect to computation time by Jobard and Lefer [73]. These ideas build the foundation for further developments in 2D (steady and unsteady flow) as well as 3D line seeding [114]. With respect to feature curves in 3D vector fields, strategies range from homogeneous seeding [112] and importance-driven seeding in the vicinity of critical points [186, 202] to topology-aware seeding [201] and line density control in physical [25] or image-space [98]. Also, stream line placement can be coupled with the selection of an adequate view-point [105, 170]. Finally, providing the user with adequate tools to interact with the integral lines or surfaces is a vivid research topic [26, 66, 161].

Placement and seeding of integral surfaces is not as thoroughly researched as the handling of integral lines. Still, there exist methods to compute topologically relevant stream surfaces [38, 128] and to automatically generate seeding curves at the domain boundaries [34].

Perceptual Issues But even with an appropriate and fast seeding approach, perceptual issues may occur that hinder insight. The user might misinterpret the spatial relations of the geometric objects because of occlusion or insufficient depth cues. With respect to line renditions, perception can be enhanced with improved illumination [104, 112, 203] or halos [37]. Advancing integral lines to three-dimensional tubes or ribbons also improves spatial perception [178] and allows to encode additional information by texturing [72, 96].

The challenge with integral surface rendering is the problem of self-occlusion, especially with complex surfaces. This problem has been tackled with surface cut-outs [101], advanced transparency [22, 43, 63], and texturing [43, 99].

2.2.4 Feature-Based Flow Visualization

Feature-based methods focus the visualization on flow features, i.e., areas of specific interest (see Sec. 2.1.3). While the original vector field contains all measured or simulated data, the consideration of specific flow features allows enlightening visualizations since the data is reduced to the aspects of interest.

How a flow feature is extracted depends on the exact feature definition as well as the available data. Post et al. [132] divide feature extraction methods in three main groups. The first group contains extraction methods that are based on *image processing* techniques. Flow fields given on a regular grid have much in common with image data. Therefore, filtering (edge detection, smoothing) or basic segmentation methods (thresholding, region-growing) can be utilized for flow feature detection. Challenges arise, however, when methods are not easily transferred from scalar image data to vector flow data. Also, the application is mainly limited to flow data given on regular grids, as methods are oftentimes not efficient or doable for complex grids used in CFD simulations. The extraction methods in the second group utilize *vector field topology* that mainly deals with critical points of a vector field as described in Section 2.1.3. Finally, the third group consists of methods detecting *physically meaningful* flow features. These physical properties, such as vorticity, low pressure etc. are used directly to formulate detection algorithms. Most flow feature extraction techniques belong to this third group, but are oftentimes combined with topology analysis or image processing methods.

A full coverage of the extraction and visualization of various flow feature types will not be provided here. Instead, we focus on different selected techniques to detect vortices as this will be of importance in the remainder of this thesis. For a detailed survey on vortex detection, we refer to Jiang et al. [71]. An overview about flow feature extraction approaches in general can be found in Post et al. [132] and Laramée et al. [93].

Vortex Detection A vortex is swirling flow around a central axis. One can observe this phenomenon frequently in everyday and natural life and it is of interest in various research and engineering settings. Still, this visual impression of a vortex has not been successfully transferred into a generally accepted, formal definition yet. As a consequence, there exist many vortex extraction approaches in literature, which differ in the vortex definition they utilize. Other differences are whether the methods work locally or globally and whether they extract the complete vortex regions or focus on the vortex core lines (i.e., the vortex' central axis). Generally speaking, extraction of vortex regions is rather inexpensive and easy to implement. However, the border between neighboring vortices is sometimes difficult to determine. On the contrary, the extraction of vortex core lines is more complicated and expensive. An advantage is the compact vortex representation by only a line, which also implicates that different vortices can be distinguished more easily [71].

Vortex Region Detection Early vortex extraction methods work with properties such as vorticity magnitude [187], vorticity lines [116], or low pressure [139]. But soon it was found that these criteria are not both necessary and sufficient indicators for a vortex [140].

Jeong and Hussain also identified the sole consideration of pressure minima as inadequate, since this only holds for turbulent flow at high Reynolds numbers [68].

To overcome this limitation, they suggested a vortex definition based on the eigenvalues of the symmetric tensor $T = S^2 + \Omega^2$. The symmetric tensor is determined by decomposing the Jacobian J into its symmetric part $S = \frac{1}{2}(J + J^T)$ and its antisymmetric part $\Omega = \frac{1}{2}(J - J^T)$. The symmetric tensor T has three real eigenvalues $\lambda_1 \leq \lambda_2 \leq \lambda_3$. Jeong and Hussain defined vortices as connected areas where the symmetric tensor has two negative eigenvalues, i.e. where $\lambda_2 < 0$. With this approach they still capture vortices at high Reynolds numbers, but also extract vortices at low Reynolds numbers where the pressure minimum definition does not hold. A drawback of this technique is that using zero as threshold for λ_2 oftentimes overestimates the vortex region. Also, determining the border between vortex regions located close to each other is difficult. Still, their λ_2 -method has become an established method for vortex detection.

Later, Sadarjoen et al. argued that considering local field properties (as, e.g., λ_2) is not sufficient for the detection of regional phenomena (like vortices) [143, 144]. Therefore, they suggested two regional geometric methods. First, the winding-angle method that selects vortical stream lines based on the sum of angles between successive line segments. Second, the center of curvature method, which is based on the idea that different stream lines of the same vortex would show the same center of curvature. Sadarjoen et al. also combined their geometric method with local methods [143]. They determined vortex candidates with local vortex detection (e.g., λ_2) and confirmed or discarded these candidates with regional geometric tests. Pagendarm et al. extended the center of curvature method to the 3D case [123].

In 2002, Jiang et al. used combinatorial topology to detect the center of spiraled flow. They applied Sperner's Lemma to this problem by using a labeling scheme based on vector directions [70].

Vortex Core Line Detection Extracting vortex core lines instead of vortex regions is the second group of vortex detection methods. This idea was initiated by Levy et al. who thought of vortex core lines as specific stream lines [95]. As seed points they chose positions where the angle between velocity and vorticity vectors are minimized (i.e., where helicity is at a local maximum). From there, the core line is integrated backwards in time without further helicity or vorticity tests. This bears the risk that the detected line wanders off the vortex.

Banks and Singer suggested their predictor-corrector method to solve this problem [7, 8]. Similar to Levy et al., they determine seed points based on the field vorticity. From there, they integrate vorticity lines and correct their paths such that they always runs through pressure minima of the vortex tube cross-sections (since lowest pressure is expected at the center of a vortex). Altogether, they predict the course of a core line based on vorticity and correct it with pressure information. Stegmaier et al. adopted this method but used minimal λ_2 for seed point selection and the correction step. As a result, the predictor-corrector method also works for low Reynold numbers, which was not the case with the low pressure criterium [164].

One of the standard techniques for the extraction of vortex core lines, is the

topological algorithm by Sujudi and Haines [167]. The basic idea is to detect singularities with spiraled behavior in their vicinity (see Sec. 2.1.3). In a tetrahedral grid, which they choose to be able to use an efficient linear velocity interpolation, an eigenvector analysis is performed for every point. A real eigenvalue and a complex-conjugate eigenvalue pair are necessary for a vortex core line point. The direction of the real eigenvector represents the direction of the core line, the complex eigenvectors represent spiraled flow perpendicular to the core line (see Fig. 2.5b,d). A second requirement is that the point is also a singularity (i.e., the velocity is zero) or has zero reduced velocity. The so-called reduced velocity is zero when the eigenvector of the real eigenvalue is parallel to the velocity at this point. The detected core line points in neighboring grid cells are connected to form the final core line.

Roth and Peikert presented a higher-order extraction method, designed to also work for bent vortices. It is based on the idea that vortex core lines consists of points showing zero torsion [142]. Since this technique utilizes higher-order derivatives, good data quality and resolution is required.

Feature Tracking The previously described methods focus on vortex detection in steady vector fields. Another challenge is to capture the evolution of flow features in time-dependent vector fields. Over time features may move, reshape, split, or fuse with other features. Two basic ways to approach feature tracking exist. First, features can be extracted separately for the distinct timesteps and it is investigated later on which of the detected objects represent the same feature in the different timesteps. This temporal correspondence is, for example, found based on spatial proximity or attribute similarity [45, 78, 136, 137, 160, 172]. Second, the detection is conducted directly in the spatial-temporal domain, which avoids the correspondence problem [11, 117, 176, 195].

Vortex Visualization Finally, the detected flow features are visualized for further inspection. Again, we concentrate on the visualization of vortices, as one flow feature example. Other examples are discussed, e.g., in Post et al. [132]. The most straight-forward way to display vortices in 3D is an isosurface rendering of the vortex cores or regions [8, 56]. These isosurfaces can be colormapped to provide more attribute information [164] or can be enhanced with LIC to give an impression of flow rotation [151] (see Sec. 2.2.2). Alternatively, vortices can be represented as icons. Here, vortices are not depicted directly, but as a symbolic object, which can be parametrized according to the vortex attributes [133, 191]. An example are three-dimensional elliptical vortex models whose colors and shapes are determined by vorticity and rotation direction of the vortices [144] (see also Sec. 2.3). The additional rendering of stream lines or stream tubes in the vortex vicinity also adds additional flow information and improves the clarity of the vortex rendition [191].

The temporal evolution of vortices can be either displayed in separate renditions [117] (see Fig. 2.7a) or by adding time as an additional dimension [197] ((see Fig. 2.7a)). The latter is a very compact visualization of vortices in 2D flow over

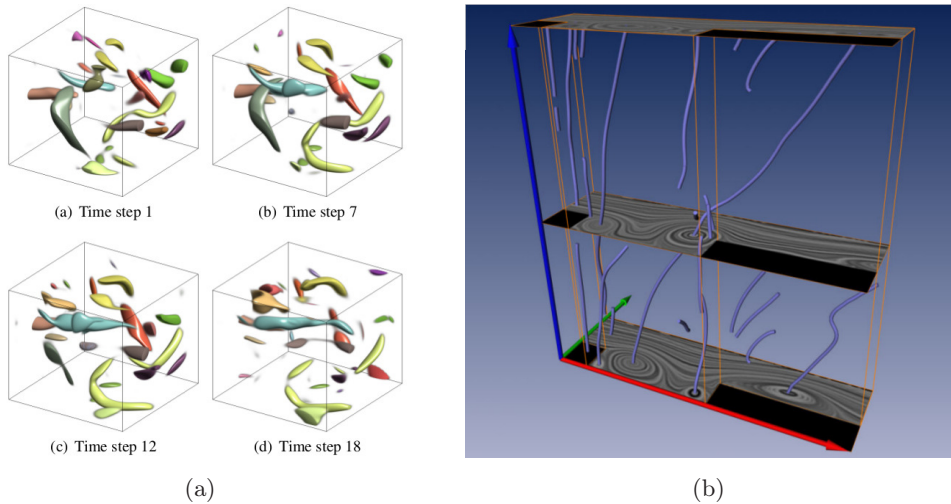


Figure 2.7: Examples of visualizations of time-dependent flow features. (a) Separate vortex renditions for different timesteps by Muelder et al. [117], © 2009 IEEE, (b) Visualization of the evolution of swirling stream line cores over time (denoted by the blue axis) by Weinkauff et al. [197], © 2007 IEEE.

time, but hard to apply to 3D. Reinders et al. show vortex evolutions with animated vortex icons and additionally provide vortex correspondences by means of an event graph [137].

2.3 Illustrative Techniques in Flow Visualization

Scientific visualization has the goal to display data in a correct, expressive, and informative way. Usually not all data is shown, but specific features or details are highlighted to emphasize an interesting data aspect. A related field to scientific visualization is scientific illustration. The task of illustrators is the effective and comprehensible depiction of (usually) complex information by means of different kinds of artwork, such as drawings, paintings, or photographs. The main difference to scientific visualization is that some general information or knowledge is displayed instead of real data. Due to this, illustrators have more artistic freedom with respect to the representation of their motif. They may idealize it, leave away uninteresting details, and emphasize important features. The message that shall be transferred is more important than the specific object or data. As an example, illustrations of a human heart, as it can be found in anatomy textbooks, do not depict all realistic details of a specific heart, but the features of a typical heart that are of interest in the particular context (emphasis on the muscles or vessels etc.). Thus, the goal is not an overly detailed or photorealistic depiction, but to communicate knowledge by means of *abstraction*, i.e., data generalization and simplification.

In the last years, the scientific visualization community has picked up different techniques effectively used by illustrators and started to apply them to real data

visualizations. This procedure is often referred to as *illustrative paradigm*. As a result, *illustrative visualization* has emerged as a new branch of visualization, which was defined by Rautek et al. as the “computer-supported interactive and expressive visual abstraction motivated by traditional illustrations” ([135], p.4:1).

Illustrative methods have already been applied in various research domains to support researchers in processing their data and gaining new insights. Striking examples can be found in the fields of biochemistry [194], geology [125], meteorology [75], and medicine [145, 162, 169, 174]. In the last years, illustrative methods have also been applied to problems in computational flow visualization and a good amount of literature has been produced in this regard. In the remainder of this section, the different types of illustrative methods are introduced and exemplified with applications to flow visualization. The methods of illustrative visualization work on different data abstraction levels and are categorized according to these levels [135].

2.3.1 Perceptual Enhancement

At the lowest data abstraction level, illustrative drawing styles are mimicked to enhance perception of the visualized data objects without having a particular knowledge about these objects. Examples are sketching styles like silhouettes, hatching, and stippling. Since the goal is not to make the objects appear realistic, this category of methods is also called *non-photorealistic rendering*. For surveys about non-photorealistic rendering methods, we refer to Strothotte and Schlechtweg [166], Gooch and Gooch [50], and Sayeed and Howard [149].

In the context of flow visualization, enhancing perception relates to the rendering of flow data so that their shape, depth, and relevant flow parameters (direction, velocity) are appropriately communicated. These methods can be applied to the raw vector field data, integration-based structures, or flow features alike. For example, Kirby et al. emulate the procedure of oil painting to display 2D vector field data. Just like oil painters build up their artwork layer by layer, the authors displayed various discrete or continuous visual elements in multiple layers to represent diverse flow characteristics [83] (see Fig. 2.8a).

A related method has been introduced recently by Auer et al [6]. Here, 2D flow data is also visualized in two layers. The background contains large-scale trends, whereas the foreground focusses on local features. All flow information is depicted by expressive visual elements, which were inspired by icons used in sketches of domain experts.

For three-dimensional raw flow data, Svakhine et al. developed a volume visualization method emphasizing structural information in the flow. For this, they mimicked hand-drawn illustrations of flow as well as photographs taken during experiments (Schlieren method) [168].

Several illustrative methods have been suggested to improve the expressiveness of integral line renderings. For example, Everts et al. used halos - known from traditional line illustrations - to improve depth perception in dense line renderings [37]. Stoll et al. presented stylized integral lines where, in addition to different

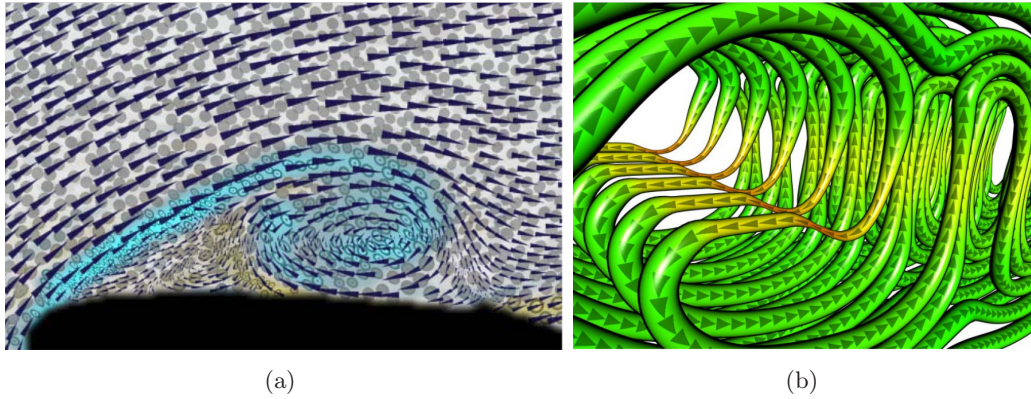


Figure 2.8: Examples of illustrative methods for perceptual enhancement. (a) Kirby et al.'s multi-layer vector field visualization inspired by oil paintings [83], © 1999 IEEE, (b) Stylized line primitives by Stoll et al. [165], © 2005 IEEE..

types of color-codings, stripe-like and arrow textures inform about flow rotation and velocity [165] (see Fig. 2.8b).

Integral surfaces suffer particularly from perception problems due to self-occlusion. A method that not only enhances shape and depth perception of these surfaces, but also gives insight about flow direction, was introduced by Löffelmann et al. [101]. Here, so-called stream arrows have been cut out of complex stream surfaces to communicate flow direction and at the same time reveal previously occluded parts of the surface. Hummel et al. created expressive integral surface renderings by using view-dependent transparency and hatch textures representing flow on the surface [63].

2.3.2 Visibility Management

On the next abstraction level, illustrative visualization is concerned with the arrangement of data in (image) space. Especially for dense data or data consisting of many pieces, occlusion needs to be decreased so that the viewer can grasp the data well. Visibility management methods (also referred to as *smart visibility*) tackle this problem by changing the layout of the data by deformation (*peel-aways*, *exploded views*) or by removing some data entirely (*cut-aways*) or partially (*ghosted views*). These techniques are very established in scientific illustration and have been successfully applied in interactive visualization software.

Flow visualization strategies for appropriate seeding and placement strategies of integral lines (see Sec. 2.2.3) are per se dealing with visibility management. In fact, some previously mentioned methods by Turks and Banks [177] as well as Jobard and Lefer [73] fall into the category of illustrative visibility management, since their goal was to achieve a line distribution and appearance like in hand-drawn sketches (such as the sparse flow illustrations by the physicist Richard Feynman).

Chen et al. visualize a 3D vector field by means of few stream lines representing

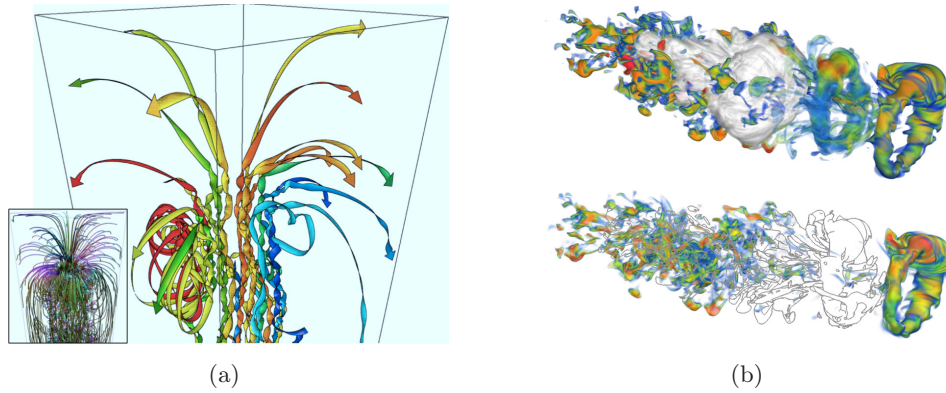


Figure 2.9: Examples of illustrative methods for visibility management. (a) Clustered and illustratively rendered stream lines by Chen et al. [24], used with permission by John Wiley & Sons, (b) Temporal visualization of several timesteps of a flow ring structure by Hsu et al. [61], used with kind permission by Springer Science and Business Media.

the typical features of the dataset. For that, a larger number of stream lines covering all important areas is clustered into groups with similar behavior. For each group, a representative is picked and rendered in an arrow-like shape [24] (see Fig. 2.9a). This work is related to the abstract visualization presented in Chapter 6.

Finally, the temporal evolution of unsteady flow data is an important topic of visibility management. Hsu et al. visualize several timesteps in one image. For this, several timesteps are either rendered on top of each other or in a vertical arrangement such that the time advances from top to bottom. In either case, the timesteps are rendered in various styles (e.g., traditional direct volume rendering, line renderings, variation in transparency or color saturation) to be able to distinguish the different timesteps and give an impression of temporal evolution [61] (see Fig. 2.9b). Joshi et al. suggest an alternative to visualizing single timesteps and animations. They use cartoon-like visual elements like strobe silhouettes or speedlines to give the impression of movement and flow direction [76].

More details about smart visibility, not restricted to applications in flow visualization, is given by Viola et al. [188].

2.3.3 Focus Emphasis

Focus emphasis is another way of managing visibility, which highlights particular features and renders them in the context of the remaining data (*focus & context methods*). This procedure requires some sort of data importance measure, which is either calculated from the data or given by the user. In contrast to this, the previously described smart visibility is mainly concerned with a general improvement of visibility without emphasizing specific data aspects. A well-known example for a focus & context method is the magic lens. A magic lens is a planar object (e.g., circle or rectangle), which the user positions in the dataset at arbitrary loca-

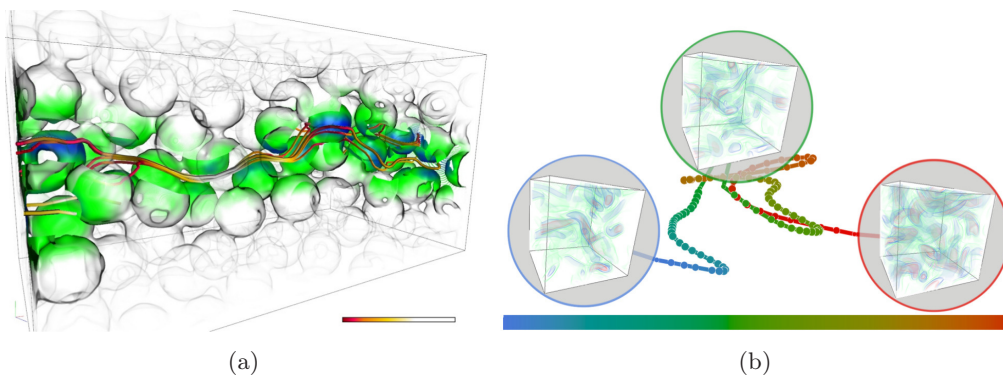


Figure 2.10: Examples of illustrative methods for focus emphasis and visual explanation. (a) Focus on a user-selected particle trajectory by Jones et al. [74], © 2010 IEEE, (b) Storyboard for a vortex dataset by Lu et al. [103], © 2008 IEEE.

tions. Within the lens, a different, usually more detailed, enlarged, or annotated, version of the data is shown. Thus, the data shown by the lens is in focus, the remaining dataset provides context information. Fuhrmann et al. [41] and Mattausch et al. [112] applied the magic lens paradigm to integral lines in 3D flow data and extended it to three-dimensional focus areas (magic boxes and magic volumes respectively). Within the focus region, a denser representation of the stream lines is shown or the region is used as seeding area for stream lines.

An alternative to magic lenses and volumes is provided by a method of Jones et al. [74]. Here, the user can put the focus on single particle trajectories. The vicinity of the selected trajectory is then highlighted by a distance-dependent color-coding whereas the rest of the dataset is rendered in an unobtrusive way (see Fig. 2.10a).

A method where the focus is not determined by the user, but objectively calculated has been suggested by Ferstl et al. [38]. The authors aim at the visualization of flow separation by detecting pairs of streak surfaces that illustrate this separating behavior. For this, the streak surfaces are seeded at either side of a ridge in the finite time Lyapunov exponent (FTLE).

2.3.4 Visual Explanation

Methods falling into this final category use advanced knowledge about the data and employ a domain-specific semantic to fit the needs of a particular research area. The knowledge is provided either by the visualization tool to guide a user through the data or by the users who use the tool to create explanatory visualizations to inform others (e.g., colleagues or the public) about their findings. One example for the first possibility (exploratory visualizations) are the interactive storyboards presented by Lu et al. [103]. They aim at a facilitated and quicker exploration of time-varying datasets with many timesteps. Representative timesteps containing essential features are determined and presented to the users as previews along a time line. This gives the users a good starting point for a strategic dataset exploration

(see Fig. 2.10a).

AniViz is a tool for the creation of explanatory visualizations (second possibility) presented by Akiba et al. [2]. This tool allows for an advanced way of building animations inspired by the paradigm used in computer animation software. There, predefined primitives (walk, run, jump) can be parametrized and assembled to build movies. Along these lines, Akiba et al. designed templates for visualization primitives (such as spatial or temporal exploration, annotations, or highlighting), which are interactively selected, parameterized, and combined by the users. Since AniViz is also a classical data exploration tool, the animations can be created as a side product during data analysis.

In summary, the low-level categories are primarily concerned with *how* to render the data. With an increasing abstraction level, the categories deal more with *what* (part of the) data to render. For further reading and a more extensive overview about illustrative flow visualization, we refer to the state-of-the-art report by Brambilla et al. [18]. They organized the literature according to the degree the visualized flow data has been processed (raw data, geometric structures, flow features) and according to the level of abstraction that is applied. The latter corresponds to the previously presented illustrative visualization categories (perception enhancement, visibility management, focus emphasis, and visual explanation).

The work presented in the next part of this thesis deals with the perception enhancement of stream surfaces and employs focus & context techniques to give the users the opportunity to interactively explore formerly hidden surface structures (see Chap. 3). Focus & context is also used in the visual analysis method for blood flow analysis where the user selects relevant flow patterns (see Chapt. 5). The abstract depiction of these patterns reduces clutter and is also practical for explanatory visualizations (see Chap. 6).

Part I

Illustrative Visualization of Stream Surfaces

Illustrative Stream Surfaces

The traditional application fields of flow visualization are the natural sciences and engineering. Here, flow visualization plays an important role, e.g., in the design process of numerous objects, such as turbines, air planes, cars, motors, and buildings. Although these objects are very diverse, their durability and usability is influenced by the flow behavior through or around them. The visualization of 3D vector fields can help to uncover the flow features affecting these objects, such as flow attachments, separation areas, and vortices.

A widely used and intuitive flow visualization method are stream surfaces. An ideal stream surface is a two-dimensional continuum of stream lines starting from a well-defined space curve. The surface represents the area through which virtual particles pass that were released from the seed curve into steady flow (see Sec. 2.1.1 and 2.1.2). Stream surfaces are an established tool for flow visualization mainly because users can easily understand their concept and the flow they represent. Unfortunately, in complex flow the shape of stream surfaces may become very complex as well and the occurring self-occlusion makes them difficult to perceive and understand. This perceptual problem renders stream surfaces useless when they depict the most interesting phenomena. Despite the fact that naïve rendering of the surface as a triangulated mesh does not solve this occlusion problem, only a small fraction of available literature is concerned with improving the visualization of these surfaces.

Our goal is to improve the perception of stream surfaces even in complex cases. Therefore, we present an approach for the illustrative rendering of stream surfaces. The field of illustrative rendering mimics techniques used in traditional illustrations with methods of modern computer graphics. The idea is to combine the best of both worlds to improve the comprehensibility of the displayed objects (see also Sec. 2.3).

Illustrations have an age-long tradition in the communication of complex knowledge of many different scientific fields, among them flow research. Here, one of the oldest examples are Leonardo da Vinci's sketches of water flow dating back to around 1500. More recent illustrations can be found in two textbooks published in the early 1980s - Dallmann's thesis [28] and *Dynamics* by Abraham and Shaw [1]. Constrained by the lack of appropriate computer graphics methods at that time, they managed to break down highly-complex facts into hand-drawn illustrations of, e.g., vector fields and stream surfaces (see Fig. 3.1). Their images are good examples for the main accomplishment of illustrators in general: simplification of complex contexts, concentration on relevant features, and neglect of details that obstruct understanding. Due to their abstract nature, the resulting images are both intuitive and esthetically pleasing. Up to now, many flow specialists have the images

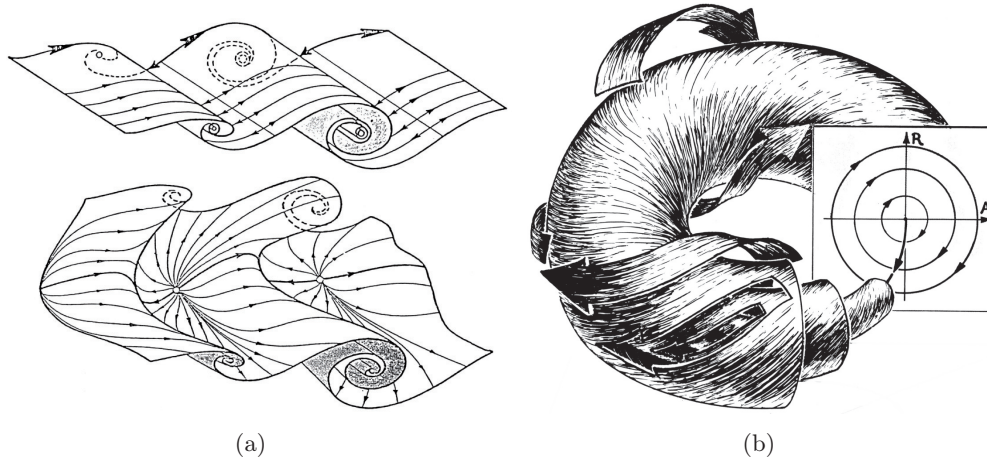


Figure 3.1: Examples for hand-drawn illustrations: Dallmann’s illustration of (a) the deformation of a two-dimensional vortical motion (top) into a three-dimensional flow (bottom) [28] and (b) a section of Abraham & Shaw’s phase portrait of an undamped pendulum [1], used with permission by R. Abraham.

by Dallmann or Abraham & Shaw in mind when discussing flow issues.

Our main contribution is a novel way of illustratively visualizing stream surfaces, so that their overall shape is well-communicated and the flow characteristics on the surface are clearly depicted. We found the inspiration for our stream surface visualizations in the already mentioned publications by Dallmann and Abraham & Shaw. We mimic their sketching and visibility techniques, such as contour lines, halftoning, and different cutting elements, and combine them for a more comprehensible visualization of arbitrary stream surface meshes. And to go further than merely static textbook images, we use modern computer graphics technology to enhance these methods and allow interactive exploration of the stream surfaces. The user can, e.g., interactively move cuts and slabs along geometrical cutting planes or along time lines or stream lines to examine the surface shape and the flow it represents.

The remaining sections have the following structure. An overview of related work concerning stream surface rendering as well as the illustrative rendering of complex objects is given in Section 3.1. The design decisions and the applied methods are presented in Sections 3.2 and 3.3. After that, we present examples of our illustrative stream surfaces (see Sec. 3.4) from various flow simulations and discuss and conclude this work in Section 3.5. This chapter is mainly based on [17].

3.1 Related Work

In this section, we give an overview of previous publications related to the focal point of our work, which is the rendering of stream surfaces and the illustrative rendering of complex objects. Foundations and related work with respect to stream surfaces and their construction as well as illustrative rendering are given in Chapter 2.

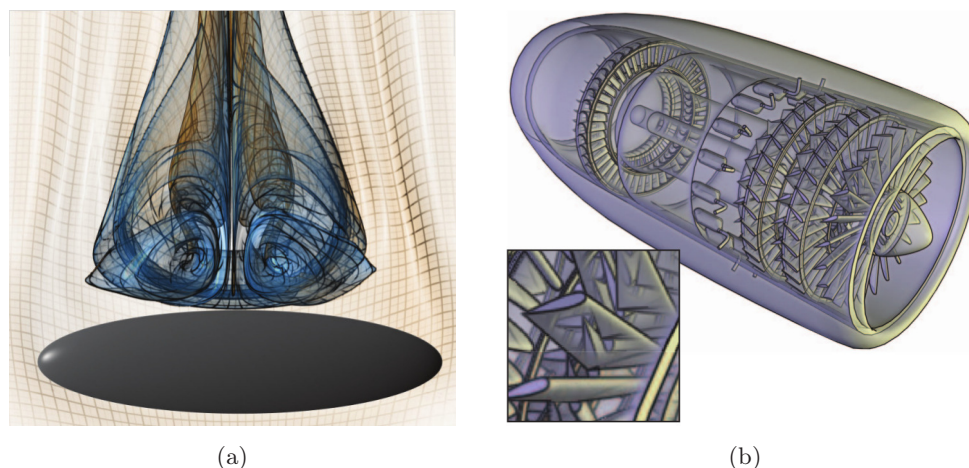


Figure 3.2: (a) View-dependent transparency and flow texture by Hummel et al. [63], © 2010 IEEE, (b) Smart transparency by Carnecky et al. [22], © 2013 IEEE.

Stream Surface Rendering Despite the large body of research in flow visualization and although stream surfaces tend to produce visual clutter through self-occlusion, the rendering of stream surfaces has received relatively little attention so far. One of the first attempts to improve the expressiveness of stream surfaces were the stream arrows presented by Löffelmann et al. [99, 101]. Up to the publication of our work, these papers have been the only illustration-inspired approach for the rendering of flow surfaces. Stream arrows cut out of the surface emphasize the direction and behavior of the flow represented by the surface. Subjacent structures become visible through the resulting holes. Still, the understanding of the flow on inner surface parts is limited. We tackle this problem with interactive cutting elements for an improved view on formerly occluded structures and the flow they represent.

Later on, Laramée et al. presented a combination of stream surfaces and texture-based flow visualization, where the surfaces were rendered with flow-depicting texture (LIC) [91]. This gives an impression of flow direction on the surface but is not overly useful for complex surfaces. More recent papers on the topic by Garth et al. [43] and Krishnan et al. [86] address the occlusion problems and perception issues by using transparency combined with time lines on the surfaces (see Fig. 2.6b). These approaches give high-quality rendering results, but are not interactive in the way that users can adapt the rendering to their needs. An earlier paper by Theisel et al. [173] also dealt with occlusion by reducing the displayed geometry to the intersection of certain meaningful surfaces (saddle connectors). However, their approach can not improve the rendering of a single surface.

After submission of this work, two notable illustrative surface visualizations were published. Hummel et al. used a view-dependent transparency to overcome self-occlusion and make underlying surface layers visible [63]. Also, a texture motivated by the hatching technique of illustrators depicts flow direction on the flow surfaces (see Fig. 3.2a). A difference to our approach is the lack of surface exploration tools.

Carnecky et al. presented a smart transparency approach for flow surfaces [22]. This approach is motivated by illustrations of overlapping structures and is supported by findings in the cognitive sciences on the perception of transparency. The results are convincing, but the method focusses on the shape and depth information and neglects information about the flow on the surface (see Fig. 3.2b).

Finally, the only paper picking up Poincaré maps has been published by Löffelmann et al. [100]. In contrast to our work, it does not relate to stream surfaces.

Illustrative Rendering of Complex Objects As already stated above, in stream surface rendering one main challenge is self-occlusion and the visualization of several layers. The same is the case in the field of technical illustration, since its motifs (such as, e.g., motor blocks, cars) are equally complex. Here, line renderings (depicting several layers) with some degree of surface shading are the most widely used technique, which we apply as well. Fundamental work concerning line renderings was presented by Dooley et al. who set down illustration rules for line renderings: line width, transparency, and style (e.g., dashing) depict different line characteristics, such as importance, layering, or silhouette type [31]. Appel discussed haloed line visualizations to improve depth perception in line renderings [4]. In our method, we distinguish layers by attenuating the line color with increasing depth.

Examples of line renderings in technical illustrative visualizations include the work by Nienhaus and Döllner [120], Kaplan et al. [79], and Fischer et al. [39], who use several silhouette layers to convey the inner construction of technical models. Surface shading is either omitted [79] or unobtrusive methods like transparent rendering [30, 120] or halftoning [39] are applied. We use halftoning, which is also recommended by Dooley et al. [32]. They discuss techniques to efficiently render nested surfaces and prefer hatching-like surface style to transparent renderings because of the improved depth representation of the layering [30, 120].

Apart from cleverly combining line and surface renderings, occluded structures can be displayed with various illustrative compositing techniques (see *visibility management* in Sec. 2.3). These smart visibility methods differ mainly in the way they represent occluding structures, i.e., either by deforming the model (peel-away, exploded view) or by removing them completely (cut-away view) or to some extent (ghosted view). Only variants of cut-away and ghosted views are applicable to stream surfaces. Exploded views are inapplicable here because they are designed for the display of complex models consisting of various units [19] and stream surfaces usually consist of only a single part. Applying deformations, as used in peel-away views [27], to stream surfaces would also modify the flow information, which may lead to misinterpretations. Cut-aways and ghosted views, however, reveal subjacent layers and still provide sufficient context information [98]. We modified the cut-away and ghosted views concept by cutting out sections of the surface and visualizing them in the context of the whole surface. By using cuts defined by geometrical properties, inner shape is made visible. Cutting according to flow characteristics allows to depict and track flow on the surfaces.

3.2 Visualization Design

Requirements. When aiming at an intuitive stream surface visualization, it is essential to identify and capture stream surface characteristics the viewer needs to understand the specific flow situation: First, the *overall shape* of the stream surface is a main clue for understanding the flow. However, stream surfaces are usually non-trivial surfaces with a high degree of self-occlusion. Therefore, rendering and interaction techniques must be provided for the enhancement of shape and depth perception, especially in the case of usually hidden parts of the surface, such as, e.g., inner windings or vortex breakdown bubbles. Second, *flow features on the stream surface* are of importance, because shape by itself cannot communicate the flow situation completely. Flow direction, courses of time- and stream lines as well as the location and characteristics of singularities, such as sources and sinks, are of interest (see Sec. 2.1.3). Here, methods must be taken into account that can depict flow directions on all parts of the surfaces without leading to visual clutter and thereby impairing the overall shape perception.

How these requirements are met in traditional illustrations. Both, Dallmann’s thesis as well as Abraham & Shaw’s textbook, have the status of a standard work in the field of flow research. Their images set the benchmark for our visualizations and their applied illustration methods serve as the starting point for our rendering techniques. In Figure 3.1a, examples of Dallmann’s illustrations show the *overall surface shape* by silhouettes and feature lines. Hidden lines are also drawn at specific positions. Their dashed appearance indicates that they are not visible from the current point of view. Different textures or colors are used to help differentiate between the front and backside of the stream surface. *Flow direction* is mainly represented by stream lines with arrow heads drawn onto the surface and the contour lines. These are especially useful when complex flow features appear, such as the sources in Figure 3.1a (bottom). Note that stream lines not only give information about the flow directions, but also about the surface curvature and thereby improve shape perception. The same is the case with the hatching technique applied in Abraham & Shaw’s illustration in Figure 3.1b: strokes not only depict the flow direction, but also the torus shape. Cutting away different parts of the stream surface is often applied to allow insight into inner surface parts. In Figure 3.1b, e.g., the different layers of the torus are represented by successively peeling away the outer layers. Further, Poincaré sections (see Sec. 2.1.1) are used by Abraham & Shaw to show the inner layers and the respective flow direction.

Design decisions based on requirements. Our goal is to enhance the perception of stream surfaces and support flow researchers in their investigations by meeting the requirements discussed above. Since stream surfaces are very diverse concerning shape, size, and complexity, we do not propose a single visualization technique applicable to all stream surfaces in general. Instead, we aim at providing the flow researcher with a toolbox of different illustrative rendering and interaction

techniques. With that, the users can explore and interact with stream surfaces without further preprocessing and adapt the visualization depending on the currently observed surface and the flow aspects they are interested in. In contrast to the hand-drawn illustrations presented above, we do not only provide static images, but exploit the advantages of computer graphics and allow real-time interaction in 3D. We adapted all techniques to be GPU-based and thereby allow for a real-time switch between different rendering techniques and a comfortable interaction without delays even for large stream surfaces.

In accordance with the demands discussed above, we provide visualization techniques representing the *overall shape* and *flow features*: The basic shape is provided by silhouettes, features lines, and halftoning without introducing too much visual clutter and occluding inner structures. Flow information on the surface is depicted by illustrative surface stream lines. The problem of visualizing shape and flow characteristics on inner and usually occluded parts of the stream surface is solved by introducing movable cuts and slabs. By moving them interactively, the user can explore the surface shape and the flow. In the following, the implemented methods are explained in more detail:

- *Silhouettes and Features Lines* depict the main shape of the stream surface. In addition to the shape information of the front layer, the silhouettes of the second and rear surface layer give shape information of otherwise occluded surfaces. With increasing depth, the silhouette colors become less saturated and by that more similar to the white background color. This depth cue relates to *proximity luminance covariance* [193], mimics atmospheric depth, and hence allows a distinction between the different silhouette layers. In contrast to rendering nested surfaces with varying transparencies, this method gives unobtrusive context information, which can be easily combined with the techniques described below. The display of the individual layers can be controlled by the user. Only a low number of layers is shown to avoid visual clutter. For a closer inspection of occluded layers, interactive tools are provided (see below).
- *Halftoning* is a non-photorealistic shading technique, that we apply as a discreet way of giving further shape and depth information, especially in more complex surfaces. Different colors can be used for the back and front faces of the surface, which allows an easier understanding of, e.g., narrow windings. Its advantage is that it minimizes distraction because colors of subjacent surfaces remain unchanged and the depth ordering of nested structures remains clear. Both can be problematic with transparent surface renderings [32].
- *Illustrative Surface Stream Lines* are ribbon-like structures with arrow heads depicting flow direction and surface shape (similar to the stream lines in Fig. 3.1a). Their appearance (e.g., width) and the number of arrows per stream line can be adapted by the user. The illustrative surface stream lines

are constructed with a GPU-based method using a geometry shader. In contrast to dense texture-based visualization techniques [91], this method allows the depiction of flow direction on the surface without occluding inner structures.

- *Cuts* are interactively movable lines, which are useful for the exploration of the surface shape and the flow it represents. The cuts are either defined based on geometry or on the flow parameters s and t given for each surface vertex. Geometry-based cuts correspond to the intersection line of a user-defined cutting plane with the stream surface. This provides insight into the inner shape of the surface. Also, Poincaré sections can be visualized for periodic orbits (see Fig. 3.12c). Flow parameter-based cuts on the other hand represent single stream lines (defined by an s -isovalue) and time lines (defined by a t -isovalue). By moving these parameter-based cuts, either time lines on the surface can be tracked over time or stream lines can be explored for different starting points. The flow direction on the cuts can be depicted by additional arrow heads.
- *Slabs* are an extension of the cuts in the sense that - instead of lines - stream surface sections of user-defined width are displayed. Like with the cuts, a geometry-based slab results in a straight section through the stream surface (see Fig. 3.10b), whereas a flow parameter-based definition produces a strip of stream lines or time lines respectively (see Fig. 3.10a). By moving the parameter-based slabs, the user can track the flow over the stream surface (either by following time line strips over time or by observing stream line strips for different starting points). An exploration like this is not possible with existing methods so far. The rendering techniques described above (silhouettes, feature lines, halftoning, illustrative surface stream lines) can be applied to these slabs as well, which enhances shape perception in comparison to cuts.

3.3 Method Description and Implementation

To provide the proposed real-time interactivity, the previously presented visualization design is implemented on the GPU. An overview of our algorithm is given in Figure 3.3. It consists of several preparation steps (indicated by the gray boxes) and the final composition (blue boxes). The user can combine the desired visualization features during runtime. According to these user specifications, the preparation steps are executed. The intermediate results, which are, e.g., silhouettes, surface, or slab style, are rendered into textures and made available to the shader performing the final composition in image space. The composition shader assembles the different features step by step from back to front.

Some techniques are based on the geometrical features of the stream surface whereas others (e.g., illustrative surface stream lines or the flow variants of the cuts and slabs) need flow parameter information. These s - and t -parameters are

delivered to the shaders as the stream surface’s texture coordinates. In the following, the different visualization features and their implementations are explained in more detail.

3.3.1 Silhouettes

Silhouettes are the main cue for shape recognition of 3D objects and are well-suited to give context information when used with other methods, like shading techniques. According to Gooch and Gooch [50], silhouettes are generally defined as points on a (smooth) surface with normals perpendicular to the viewing direction. With polygonal models, silhouettes can be approximated as edges of a front- and a back-facing polygon. With this definition, the contour lines separating the model from the background as well as internal silhouettes can be described. Further feature lines necessary for shape recognition are creases (discontinuities on the surface, such as the edges of a cube) and border lines (edges belonging to only one polygon). The latter occur in non-closed meshes and, thus, in stream surfaces as well.

Silhouette Detection. Isenberg et al. gives an overview over silhouette detection algorithms and distinguishes *image-space*, *object-space*, and *hybrid* algorithms [65]. They differ in performance, precision, and the resulting appearance of the silhouette. Since we aim at interactive frame rates, but do not require stylized silhouettes or subpixel precision, we detect silhouette with the hybrid algorithm of Decaudin [29]. This algorithm has become a standard method, because it detects all silhouettes and feature lines and solves the hidden line problem implicitly. Here, silhouettes are determined based on depth changes (mainly for outer contours) and normal discontinuities (mainly for internal silhouettes) in the scene. We implemented the algorithm with two rendering passes (*Rear Silhouette* and *Front Silhouette* in Fig. 3.3). In the first pass, the stream surface’s depth values are stored in a depth texture whereas the transformed normals are stored in a normals texture. The latter is an RGB-texture containing a pixel’s normal vector as color value. In the second pass, silhouettes are detected in these textures. For this, a Sobel edge detection filter is applied to the depth texture calculating the depth gradient for each pixel. If this gradient exceeds a certain threshold, the pixel is classified as silhouette. In the normals texture, silhouettes are detected by checking for normal discontinuities, which occur when normals of neighboring pixels are not parallel. Thus, for every pixel the dot product of its normal with the normals of its four direct neighbors is computed. If the sum of these four dot products drops below 4.0, the normals are not completely parallel. We used a threshold of 3.95 to detect a surface discontinuity and by that a silhouette. The final silhouette result is stored in a texture, which is initialized with white and set to black for all pixels classified as silhouette either in the depth or in the normals texture.

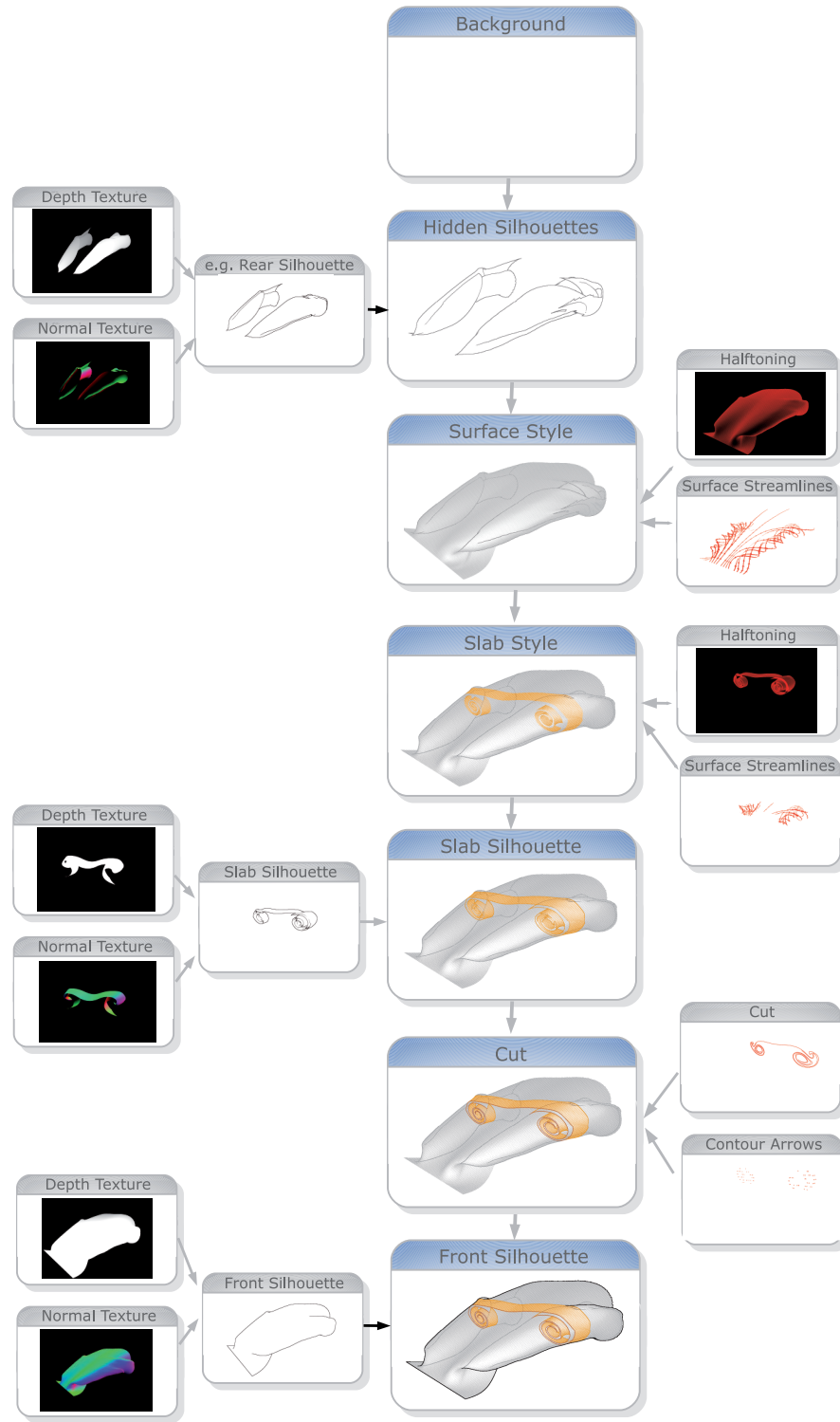


Figure 3.3: Overview over the required rendering steps. The gray boxes resemble the intermediate results of the preparation steps for the different illustrative features. The results are stored in textures and delivered to the composition stage (blue boxes). This final shader assembles the scene from back to front. © 2010 IEEE.

Layer Management. The silhouette detection presented in the last paragraph can be used to capture silhouettes from different surface layers (front, second, and rear; see Fig. 3.4). For that, the normals and depth values of the designated layer are stored in a texture in the first rendering pass. This is controlled by appropriate depth tests. For the *front* layer, the standard z-buffer test is used. The *rear* surface is captured by rendering only fragments with the highest depth values by changing the standard z-buffer depth test from `GL_LESS` to `GL_GREATER`. For the *second* surface layer *depth peeling* [39, 120] is applied. For that, each fragment of the stream surface has to pass two depth tests. It has to be at a greater depth than the front layer. This peels away all fragments of the front layer. After that, the remaining fragments have to pass the z-buffer test. Together, this results in the silhouette detection in the second layer.

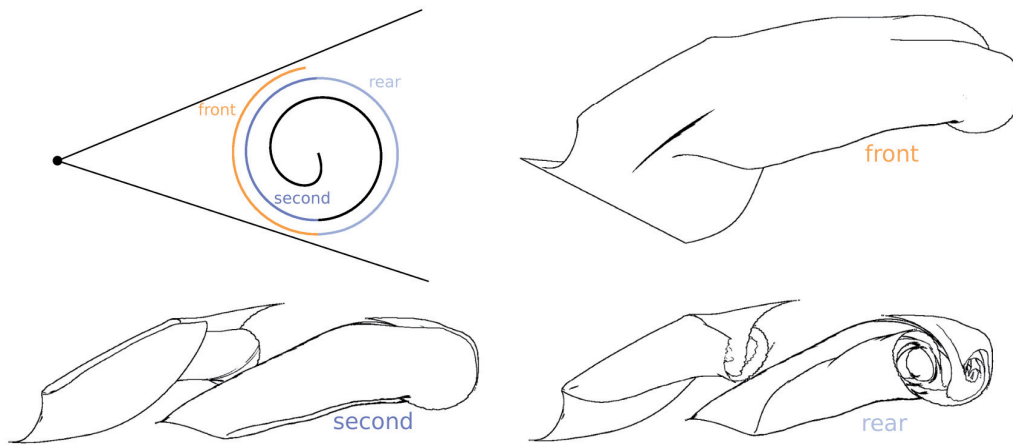


Figure 3.4: The different silhouette layers. Upper left shows the scene layout and the computed silhouette layers: front (orange), second (dark blue), and rear layer (light blue). The other three images show the intermediate results of front, second, and rear surface layer of a stream surface of the delta wing dataset. © 2010 IEEE.

3.3.2 Halftoning

Halftoning traditionally denotes the preparation of newspaper images for black-and-white printing. Continuous tone images are reproduced by filling small resolution units with black dots of varying size. For modern graphical output devices, these resolution units are composed of pixel areas of varying size ($n \times n$). Different tones are achieved by controlling the number of pixels set in this unit according to a specific pattern (*ordered dithering*). Similarly, halftoning can be used for surface shading. Here, the pixels of predefined areas are set depending on the current lighting. One advantage of this shading technique is that only a portion of the surface pixels are set so that subjacent structures are still visible and their colors are not modified.

We apply the *procedural screening* technique described by Strothotte and

Schlecht-weg [166]. In contrast to, e.g., ordered dithering, there is no need to provide the pixel pattern (called *dither screen*) as input texture. Instead, the dither screen is generated by a dither kernel $\tau(u, v)$ and a mapping function $M(x, y)$. The dither kernel controls the appearance of the halftoning texture, whereas the mapping function defines its scale and orientation in image space. The technique is implemented as follows. First, a pixel's image space coordinates (x, y) are transferred to dither texture coordinates (u, v) by applying the mapping function M .

$$(u, v) = M(x, y) = \left(\frac{x' \bmod n}{n}, \frac{y' \bmod n}{n} \right) \quad (3.1)$$

with $(x', y') = R_\alpha \cdot (x, y)$

where R_α is a rotation matrix that we use to achieve a slightly tilted appearance of the halftoning pattern. To obtain valid dither texture coordinates (u, v) , the result is clamped to $[0, 1]$ by taking into account the user-defined texture side length n (values from two to four lead to good effects). The dither kernel $\tau(u, v)$ assigns a halftoning threshold to each dither texture coordinate.

$$\tau(u, v) = \begin{cases} I_{cross} \cdot v, & \text{if } u \leq I_{cross} \\ (1 - I_{cross})u + I_{cross}, & \text{else} \end{cases} \quad (3.2)$$

For an intensity value higher than I_{cross} , crossed lines are used instead of parallel lines. Finally, for each pixel the current lighting intensity is compared with the threshold at the corresponding dither texture position. The pixel is set, if the lighting falls below $\tau(u, v)$:

$$pixel(x, y) = \begin{cases} 1, & \text{if } I_{pixel}(x, y) < \tau(u, v) \\ 0, & \text{else} \end{cases} \quad (3.3)$$

For the lighting, we use the non-linear shading model of Krüger et al. [89]:

$$I = a_l a_m + \left(\frac{L \cdot N + 1}{2} \right)^\alpha \cdot d_l d_m \quad (3.4)$$

L is the lighting direction at the current surface point, N is the surface normal. The terms a_l , a_m and d_l , d_m describe the ambient and diffuse lighting and material intensities respectively. Similar to a gamma correction, the accentuation of different tones can be controlled with α . For $\alpha < 1$, the middle tones are emphasized and the light and dark tones are compressed, leading to a rather uniform result. For $\alpha > 1$, more contrast is achieved since light and dark tones are accentuated. With the shading model, the appearance of the halftoning can be adjusted without changing the dither kernel characteristics. The result of this rendering step is stored in a texture (*Halftoning* in Fig. 3.3) and provided to the final composition shader.

3.3.3 Illustrative Surface Stream Lines

Illustrative surface stream lines are ribbon-like stream lines with arrow heads, which indicate flow direction and singularities on a stream surface (see Fig. 3.1a). As

stated previously, a stream surface is a continuum of stream lines starting from a seed curve and extending over time. The flow direction is represented by the surface parameterization (additional scalar values for each vertex). The parameter t depicts time and increases along the stream lines. The parameter s increases along the seed curve (see Fig. 3.5 and Fig. 2.3). We use a geometry shader to detect isolines on the stream surface and create the ribbon-like geometries along these lines. A geometry shader is executed in the rendering pipeline between vertex and fragment shader and emits new geometry derived from the input primitives. In our case, as in put the shader receives the stream surface triangles, the flow parameters s and t for every vertex (transferred as texture coordinates), and user specifications concerning the number and appearance of the stream lines (width, number of arrow heads). The shader output are triangles representing the illustrative surface stream lines evenly distributed over the stream surface. For further use in the final composition stage, a fragment shader renders the newly created geometry into a texture (*Surface Stream Lines* in Fig. 3.3).

The stream line construction itself is implemented in the geometry shader as follows (see Fig. 3.5, middle part). A stream line is an isoline defined by a specific s -parameter value s_{ribbon} (resulting from the user-defined number of stream lines and the s -parameter range of the surface). Each triangle (red in Fig. 3.5) is tested whether its edges cross one of the stream lines to be constructed. If a crossing edge is detected, the position and t -parameter of its cutting point c with the stream line is computed. If two cutting points per triangle are detected, their connecting edge represents a section of the ribbon's centerline (green line). For every such centerline section, the respective ribbon segment is generated by forming a trapezoid such that the triangles (green) border the current triangle and are parallel to the centerline. Two special cases have to be handled. First, the centerline might coincide with triangle edges (see Fig. 3.6a). In this case every centerline segment is visited twice because of its two adjacent triangles and the surface stream line section is generated twice. To avoid that, only the segment inside the current triangle is generated. Gaps occurring at triangles with only one cutting point with the surface stream line need to be closed. The second special case is the handling of diverging stream lines (see Fig. 3.6b). This occurs at positions where the mesh resolution changes and a triangle has three cutting points. The cutting points must be transformed correctly into a branching or otherwise the two neighboring stream lines are connected. This leads to an unwanted loop and an interruption of the correct stream line course.

The arrow heads for the illustrative surface stream lines are generated analogously to the ribbons (see Fig. 3.5, bottom). The position t_{arrow} of an arrow head results from the user-defined number of arrows and the known t -parameter range. For each centerline segment, it is checked whether t_{arrow} lies between t_i and t_{i+1} . If this is the case, the arrow's center point on the stream line is derived by interpolation. The arrow head itself consists of two triangles defined by the center point and the given arrow width.

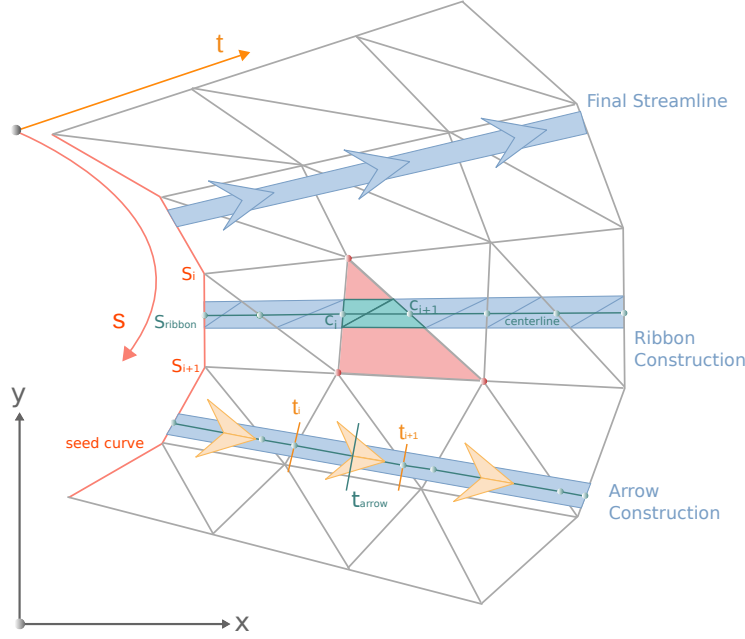


Figure 3.5: Construction of illustrative surface stream lines. The light gray grid represents a parametrized stream surface (parameter t increases along the stream lines, parameter s increases along the seed curve). A complete illustrative surface stream line is shown in the top part. In the middle, the stream line construction is shown. The edges of a triangle (red) are checked for crossings with a stream line (defined by s_{ribbon}). The cutting points (c_i and c_{i+1}) define the centerline (green) of the stream line and two new triangles (green) form the stream line segment. Then, arrow heads are added (bottom). The interpolated t -values t_i , t_{i+1} of the cutting points are checked for whether an arrow head (at position t_{arrow}) needs to be added between them. If yes, the exact position of the arrow head on the stream line is determined by interpolation and two triangles (yellow) forming the arrow are created.

3.3.4 Movable Cuts and Slabs

Cut-aways or cutting planes are used in illustrations to give insight into hidden parts of a stream surface. We apply and enhance this visualization technique by making it interactive, expanding it to slabs, and providing two variants, a *geometry-based* one to explore the shape and a *parameter-based* one to observe the flow. The difference is depicted in Figure 3.7.

Cuts and Poincaré Sections The two-dimensional cuts are constructed with a geometry shader. Its inputs are the surface triangles, the current cutting mode (parameter-based or geometry-based) and a position on the desired main axes - both specified by the user. The cut detection itself is analogous to the surface stream line detection in Subsection 3.3.3: Each triangle is tested whether its edges

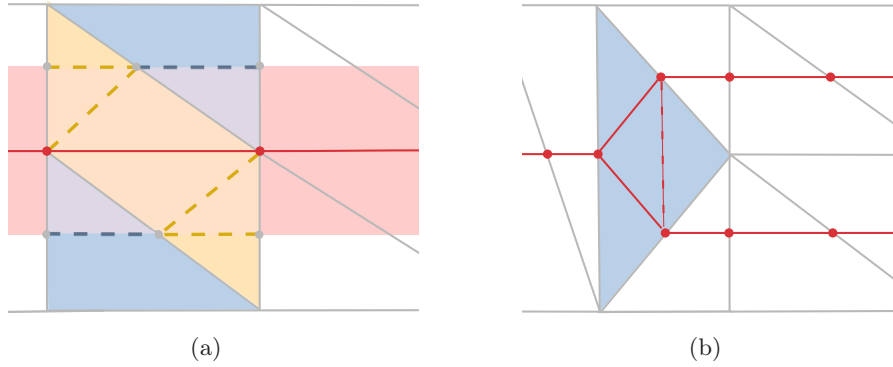


Figure 3.6: Special cases. (a) The centerline might coincide with the edges of triangles (yellow). To avoid the current segment of the ribbon (red) to be generated twice, only the part in the current triangle is drawn (where yellow and red area overlap). To avoid gaps at positions with triangles (blue) with one cutting point, appropriate triangles are inserted here (where blue and red areas overlap), (b) Diverging stream lines and changes in mesh resolution. The blue triangle has three cutting points, which need to be transformed into a branching instead of the incorrect connection of neighboring stream lines (dashed red line).

cross the predefined cutting plane. If two crossing edges are detected for a triangle, the cutting points are determined and connected. The shader output is a line strip resembling the cut (*Cut* in Fig. 3.3), which is rendered into a texture by the fragment shader. The geometry-based and parameter-based approaches differ only in the vertex information that is used for the query (either vertex coordinates or flow parameters). Our implementation of Poincaré sections are a special case of the geometry-based cuts. Here, a periodic orbit can be investigated by moving a cut around the orbit in a circular way. Therefore, the position of the cut is defined by an angle (geometry shader input) instead of a location along the main axes.

Another feature of traditional illustrations are arrows depicting flow direction on Poincaré sections (see Fig. 3.1b). We reproduced this and applied it to our cuts. The separate geometry shader generates the contour arrows and distributes them evenly on the corresponding cut according to a user-defined arrow density. The direction of the arrow heads is computed based on the t -parameter gradient at a specific position (*Contour Arrows* in Fig. 3.3).

Slabs A slab is an extension of the previously introduced cut and represents a wider, interactive surface slice. It is rendered with the same techniques as the complete stream surface (as described in the Subsections 3.3.1-3.3.3) to enhance shape perception of the slab. The computation of the slabs is accomplished in the fragment shaders of the respective techniques by discarding all fragments positioned outside the slab (given by user-defined position and width). Thus, analogous to the complete stream surface, intermediate results for the slab style (halftoning and surface stream lines), and for the slab silhouette (all depicted in Fig. 3.3) are generated

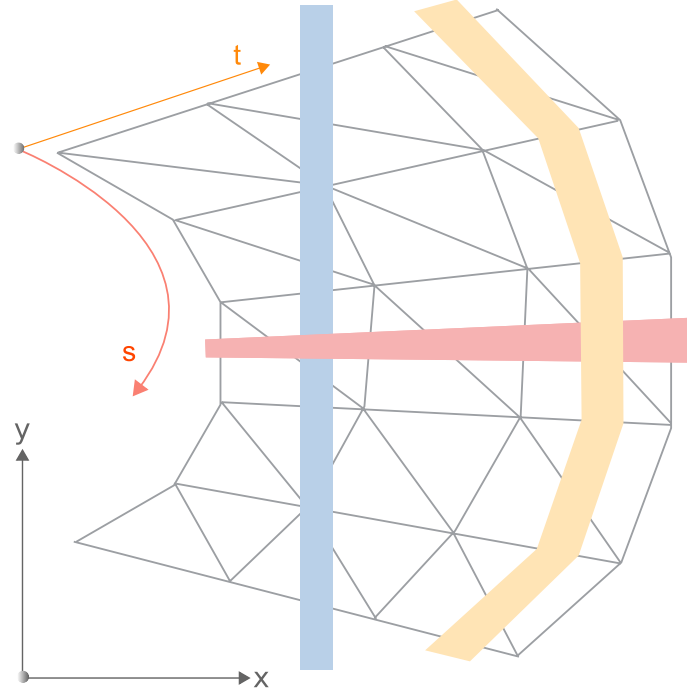


Figure 3.7: Different types of slabs. The red slab is an example for a geometry-based cutting plane moving along the x -axis (shown in 2D for simplicity). A parameter-based cut is defined based on the s - or t -parameter and is either oriented along time lines (yellow) or stream lines (blue).

and used in the final composition. Again, the parameter-based and geometry-based versions differ only in the vertex information that is used for the clipping decision, i.e., either the vertex coordinates or its flow parameters.

3.4 Results

In this section, we present illustratively rendered stream surfaces from different datasets, provide performance numbers, and report about expert feedback.

3.4.1 Delta Wing

Our first example surfaces stem from an airflow simulation around a single delta-type wing configuration. The dataset was computed in the context of the numerical research of vortex breakdown. The dataset's main features are a system of vortices and vortex breakdown bubbles with recirculation zones above the wing (see Fig. 3.8a). We process two stream surfaces, a vortex breakdown bubble and an overview surface containing both vortices. Both surfaces show a high degree of self-occlusion due to the rolling up of the vortices and the recirculating flow in the breakdown bubble. Visualizing these surfaces conventionally, i.e., with transparency and color-coding,

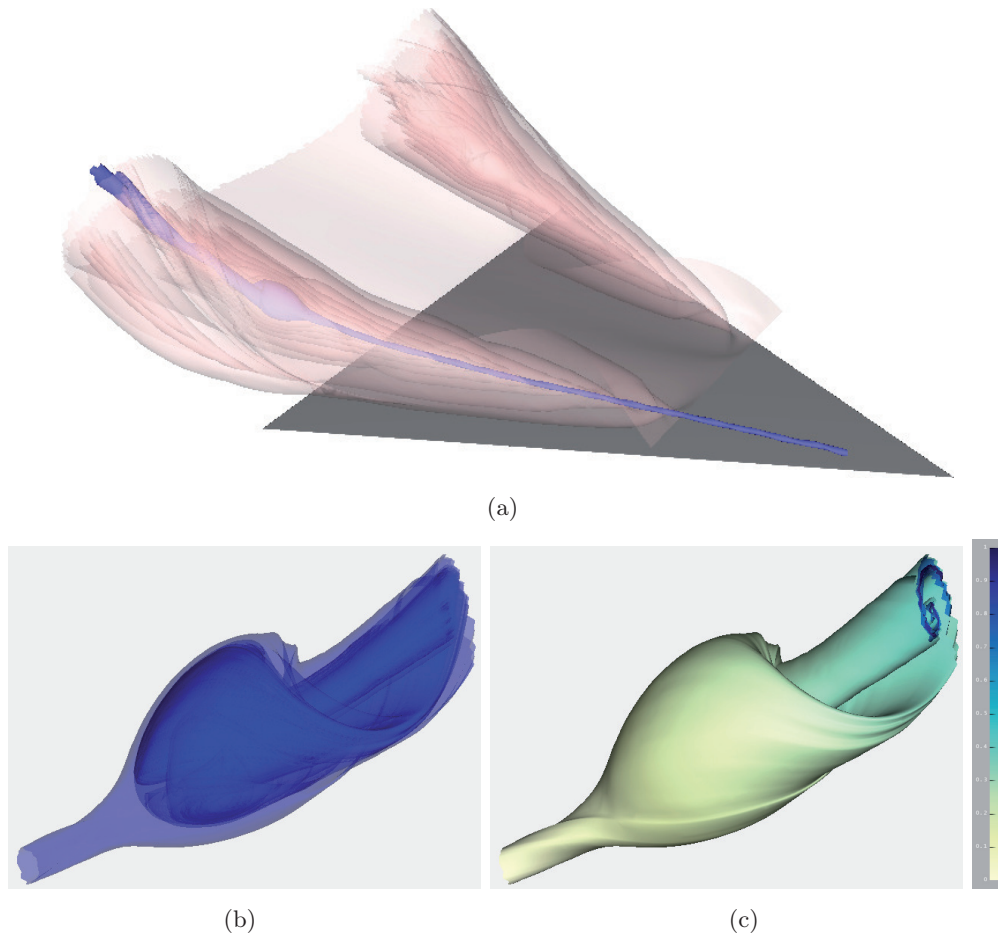


Figure 3.8: (a) An overview of the delta wing dataset with two stream surfaces: a vortex breakdown bubble (blue) and a stream surface with both vortices (transparent). (b,c) Non-illustrative renderings of the vortex breakdown bubble: Translucent (b), color-coded t -parameter (c). © 2010 IEEE; created by Alexander Wiebel.

does not lead to satisfying and comprehensible images. The transparent rendering of the breakdown bubble in Figure 3.8b only indicates that the bubble is strongly folded and built up of many layers. The coloring along the t -direction in Figure 3.8c provides information about the flow direction over time - but only on the outermost surface layer. From these depictions, it is very hard for the domain researchers to understand the surface shape and the different layers and to follow the flow.

Our method is based on the idea to mimic illustrations and to provide several useful interaction methods for stream surface exploration. As a consequence, there is not only one final result image of each stream surface. Instead, several method combinations are useful, each highlighting a different flow aspect. This is also the case for the vortex breakdown bubble. So, the first image in Figure 3.9a depicts mainly the flow through the bubble. We combine an illustrative context rendering (silhouettes and halftoning) with an interactive s -parameter-based slab. The latter

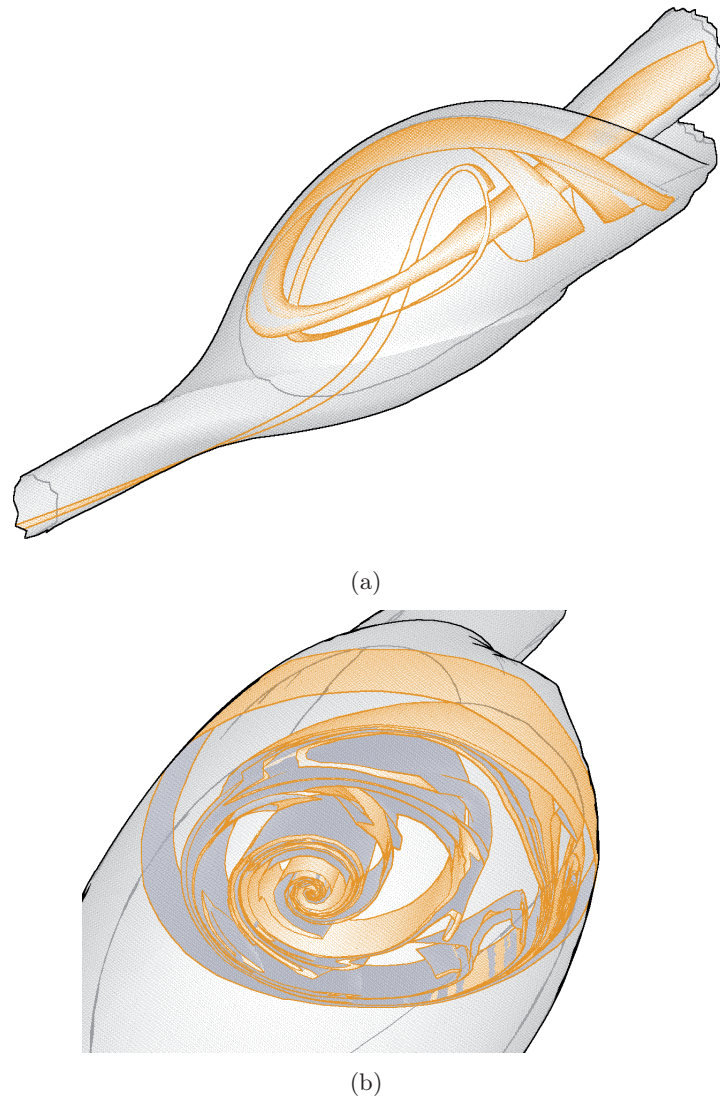
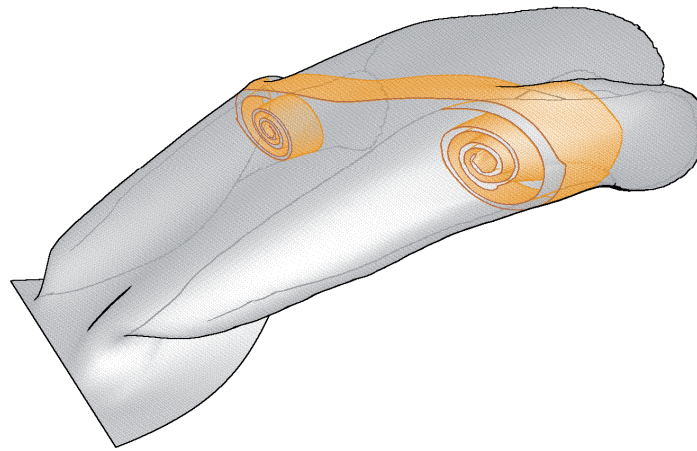
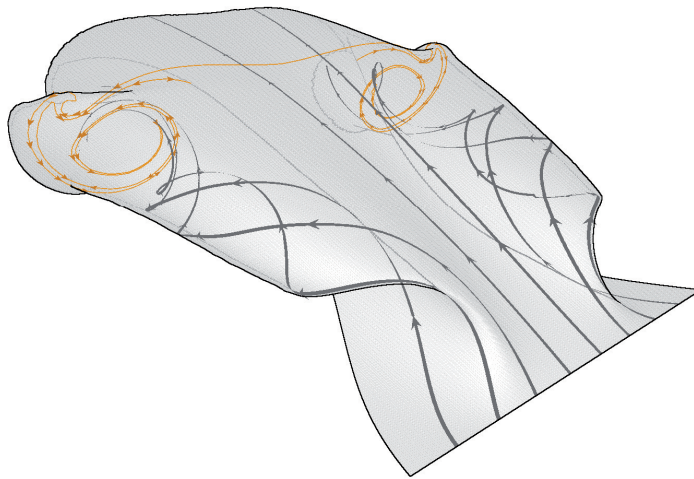


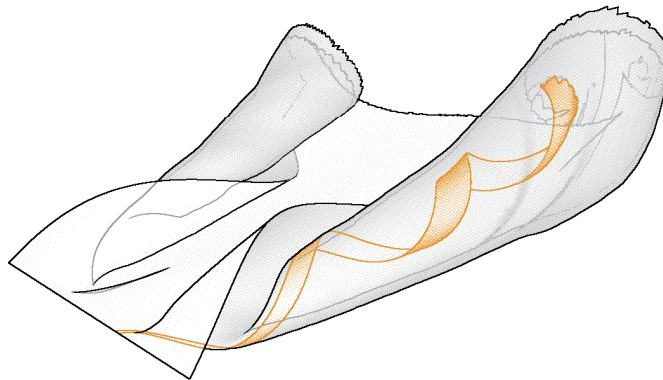
Figure 3.9: Delta wing dataset: Illustrative context rendering of the vortex breakdown bubble with (a) an s -parameter-based slab showing a flow path through the bubble and (b) a geometry-based slab showing the different folded layers of the stream surface inside the bubble. The two-colored slab allows to distinguish the front and back side of the stream surface more easily. The occasional gaps in the halftoning result from flipped normals or holes in the surface. Both images © 2010 IEEE.



(a)



(b)



(c)

Figure 3.10: Illustrative stream surface in airflow above the delta wing with (a) a geometry-based slab showing nicely the rolling up of the two contained vortices. (b) Stream surface with illustrative surface stream lines (gray) and a cut (orange) showing the rolling up of the surface and (c) a similar stream surface with a slab representing a continuum of neighboring stream lines showing the flow in the inner part of the vortex. All images © 2010 IEEE.

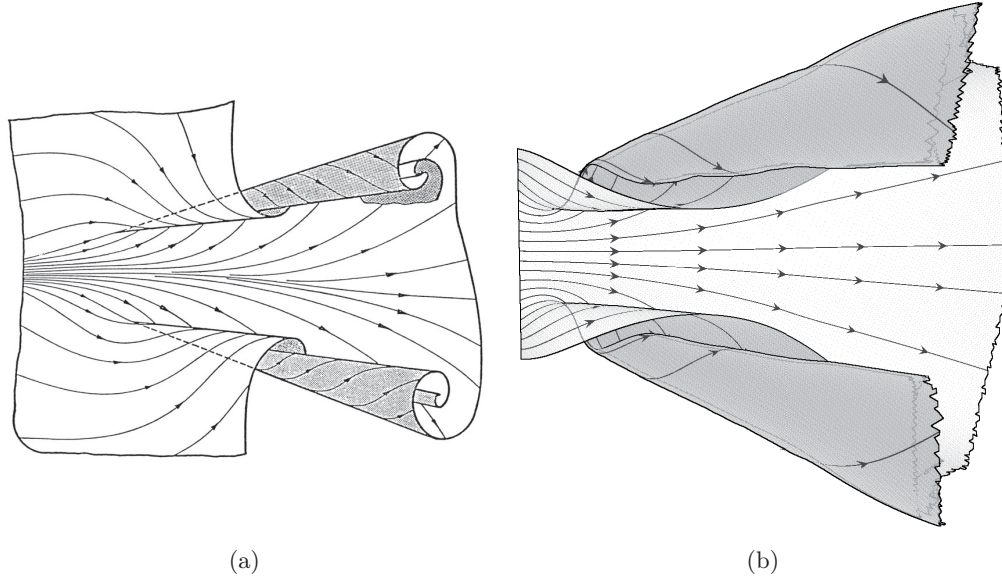


Figure 3.11: Comparison of (a) Dallmann's illustration ([28], Fig. 24) with (b) our illustrative visualization of a stream surface for the delta wing, © 2010 IEEE.

is a continuum of stream lines conveying the flow direction on several layers over time. One can see the flow entering the bubble on one side (lower left corner of the image) and curling its way through the bubble before leaving it again on the other side (upper right). This image does not show all layers as this would be confusing in this type of illustration, but focusses on the slab. A second configuration allows to inspect the different layers with an interactive geometry-based slab as shown in Figures 3.9b. Here, the layers are now clearly discernable and the bubble's inner structure is revealed.

Finally, Figure 3.10 shows exemplary illustrative renderings for the stream surface covering both wing vortices. The inner shape and windings of the vortices are the focus of the visualization with a geometry-based slab in Figure 3.10a. In Figure 3.10b, we combine surface contours with a geometry-based cut (orange) and halftoning. Illustrative surface stream lines (gray) encode flow direction on the surface. The rolling up of the vortices is more prominent in this image than in the transparent surface rendering of Figure 3.8b. Finally, Figure 3.10c provides insight into the inner flow along the streamlines of a vortex. All in all, the combination of the presented techniques yields views into the inner structure of the vortical features of this dataset that have not been possible before. We claimed in Section 3.2 that we want to mimic and enhance textbook illustrations of stream surfaces. In Figure 3.11, we demonstrate that our computer-generated visualizations are indeed very similar to these hand-drawn pictures. With silhouettes, halftoning and the illustrative surface stream lines, we apply similar methods (see Fig. 3.11b) and achieve a rendering that is as expressive as the Dallmann image. It also emphasizes the important features and nicely illustrates the vortical behavior at the rear of the surface.

3.4.2 Periodic Orbit

This section's surfaces are extracted from several periodic orbit datasets. A periodic orbit is a stream line passing the same positions in space over and over again (see Sec. 2.1.1). The periodic orbits used here, are created synthetically but it is well known that such orbits exist in real world CFD data [127]. In all our examples, the periodic orbits act like sinks, i.e., stream lines from nearby are attracted. For comparison, a non-attracting periodic orbit illustrated by Abraham & Shaw is shown in Figure 3.1b. In this figure, the flow behavior is shown with stream arrows and a Poincaré section depicting the movement of the recurring stream lines.

The stream surfaces are all seeded from circular lines. With our techniques, the specific flow characteristics of periodic orbits can be illustrated nicely. In Figures 3.12a,b parameter-based slabs along stream lines are used to show the circular flow behavior. The images nicely show the nesting, twisting, and swirl of the flow represented by the stream surface. In Figure 3.12b, illustrative surface stream line add further information about the flow direction. These illustrative visualizations unfold their value best when moved interactively by the user. Nevertheless, they already provide a good overview of the flow in still images as well.

As already mentioned, Poincaré sections are a valuable tool for the analysis of periodic orbits. Figure 3.12c-e shows our illustrative analog applied to one of the orbit datasets. The nesting of the surfaces as well as the attracting nature of a periodic orbit are clearly visible in the flow cross section. The arrows give the needed hints to the direction of the flow along the rendered contours. In the case of very tight windings, the structure is a bit clearer without the vectors. But even then, we leave this up to the users who can switch the arrows off and on (e.g., when viewing a close-up) during their exploration of the surface. The widespread use of Poincaré sections in the fluid dynamics community makes our visualization intuitive and easy to use for domain experts.

3.4.3 Combustion Chamber

The stream surfaces in this section represent the simulation result of the flow in a combustion chamber. Combustion chambers are used in central heating systems for houses. The gas enters this chamber through nine inlets distributed along the side walls (blue spots in Fig. 3.13). Exhaust gas exits on the right end (see Fig. 3.13). This construction leads to a high degree of turbulence and many vortices and singularities in the flow vector field. The turbulence is desired to achieve a good intermixture of air and gas needed for combustion.

We computed stream surfaces from two three-dimensional spiral saddle points (see Fig. 3.14). The seed lines were chosen to lie in the unstable manifolds of the saddles. Thus, both stream surfaces move away from the saddle points in a spiraling manner. For these examples, the flow behavior on the surface is more interesting than the surface shape. In the images, this flow behavior on the surface is illustrated by augmenting them with two of the proposed techniques: illustrative surface stream

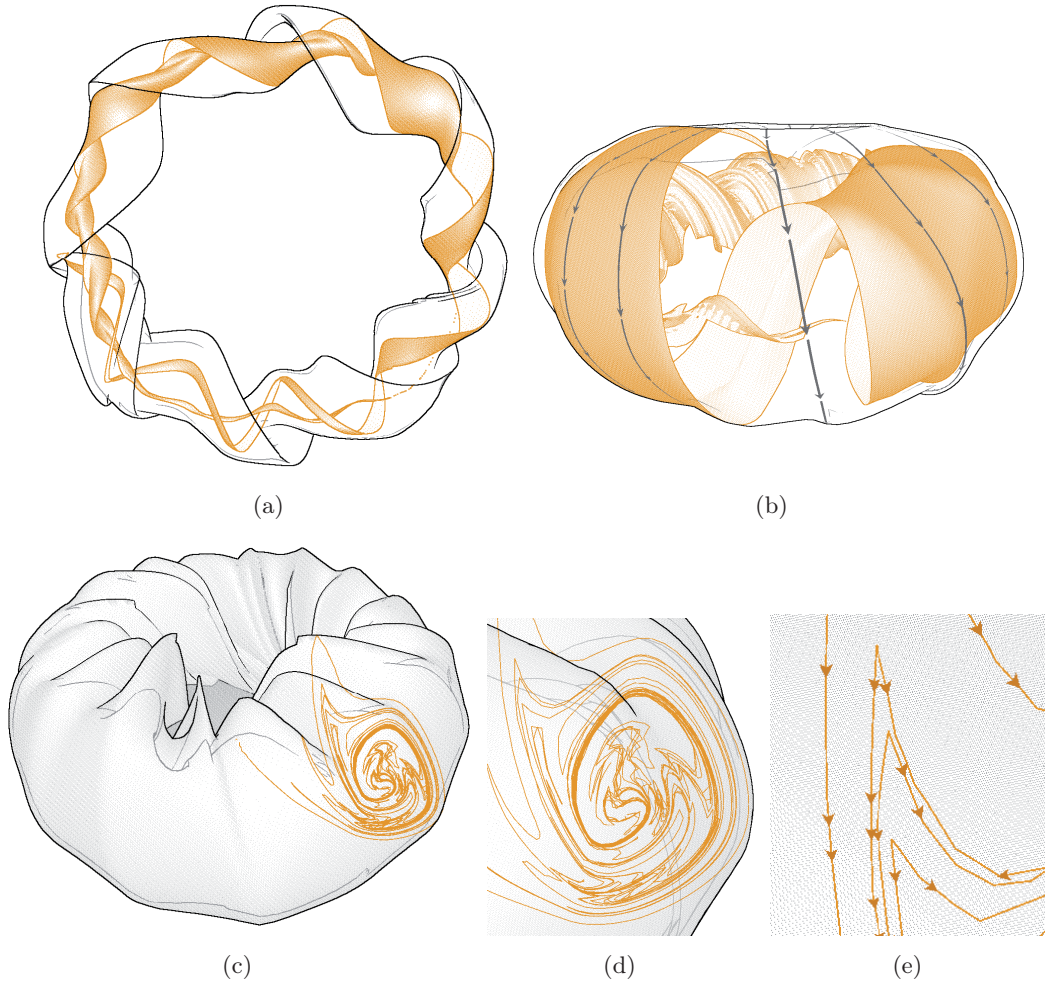


Figure 3.12: Illustrative stream surfaces in a vector field with twisted periodic orbits. (a) The parameter-based slab illustrates the flow over time. (b) Twisted periodic orbit with a parameter-based slabs along stream lines. The flow direction is depicted by additional illustrative surface stream lines. (c) Poincaré section-like flow lines in a slice orthogonal to the main flow direction of another twisted periodic orbit example. (d,e) The close-ups show the same position in the orbit and a detail with contour arrows. All images © 2010 IEEE.

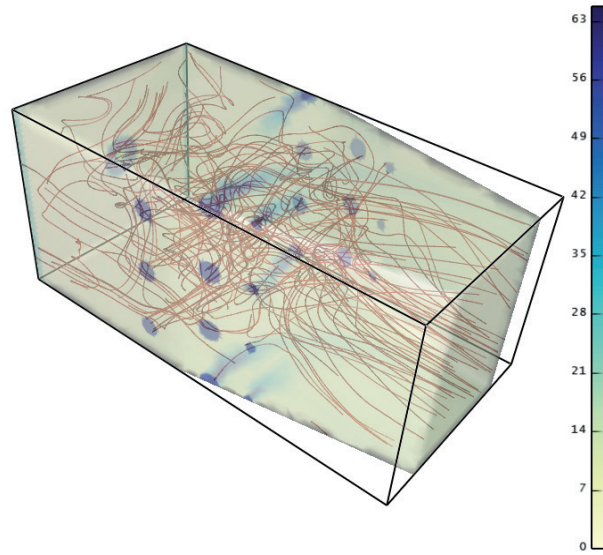


Figure 3.13: An overview of the combustion chamber with stream lines and color coded velocity magnitude on the boundary (white to blue). Be aware that the colors are paler in the image than in the color bar due to the translucency. © 2010 IEEE, created by Alexander Wiebel.

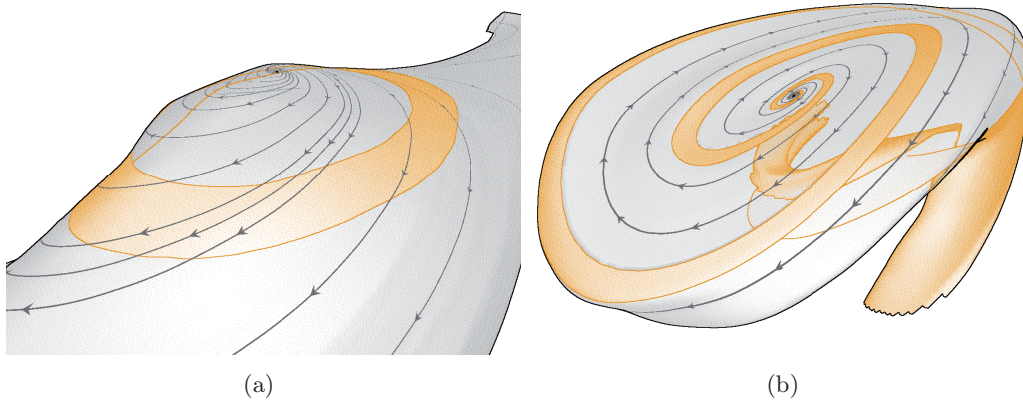


Figure 3.14: Two stream surfaces starting at spiral saddles in the combustion chamber dataset. The surfaces are rendered with illustrative surface stream lines and parameter-based slabs along (a) the time lines and (b) the stream lines respectively. Both images © 2010 IEEE.

lines and slabs. Both techniques nicely illustrate the flow divergence and rotation and, with that, the nature of the saddle point. The slab additionally allows the user to follow the flow even into hidden layers of the surface. Especially in Figure 3.14b, one clearly sees that the stream surface returns to an area close to the saddle point. A t-parameter-based slab is applied in Figure 3.14a displaying a continuum of time lines that shows where simultaneously seeded particles reside at a certain point in time. Note that the slab is interactively adjustable in width and position, allowing for an intuitive exploration of the movement of the initial seed line over the complete surface over time.

3.4.4 Performance

Interactivity is an important prerequisite for a comfortable exploration of stream surfaces. We measured performance of our illustrative rendering approach for the previously presented datasets in low (800x600) and high resolution (1800x1000) on a computer with an Intel Core 2 Quad Q9450 CPU and an Nvidia GeForce GTX 260 graphics card. The frame rates are listed in Table 3.1 and show that interactivity is achieved for all examples. Naturally the performance decreases with higher resolution and larger models (such as the periodic orbits), but still allows comfortable exploration.

Table 3.1: Frame rates (in frames per second) in low (800x600) and high resolution (1800x1000) for the datasets presented in Sections 3.4.1 to 3.4.3.

Dataset (Figure)	#Vertices	FPS (low res.)	FPS (high res.)
<i>Delta wing</i>			
3.10d	33k	168	89
3.11b	37k	116	71
3.10e	37k	125	70
3.10f	37k	153	85
<i>Breakdown Bubble</i>			
3.10a-c	133k	88	61
<i>Periodic Orbit</i>			
3.12a	888k	42	26
3.12b	396k	37	20
3.12c	396k	58	44
<i>Combustion Chamber</i>			
3.14a	124k	69	36
3.14b	47k	85	57

3.4.5 Expert Feedback

We presented our results to a fluid mechanics expert whose special subject is vortex breakdown. He regards all the visualizations suitable. He specifically considers the vortex breakdown bubbles to be a significant improvement over existing visualizations. With the inner layers of the bubble, the visualization demonstrates the

proportions of rolled-up structures, rapid changes, and its curvature and twist. He stated that he has never seen the inner details of his own data like this before.

3.4.6 Limitations

One limitation is that illustrative stream surfaces do not convey quantitative flow information, such as velocity or the like. The common way of encoding further flow parameters by color mapping is not particularly useful here. As our method uses a large variety of different visual cues for the communication of the surface structure, we believe this would make the visualization overly complex. However, because our tool is completely interactive we can simply switch between the presented and standard rendering techniques. We therefore recommend to use illustrative stream surfaces for the exploration of the surface structure and to switch to standard rendering with color coding for inspecting additional quantities at certain locations of the surface.

3.5 Discussion and Future Work

The presented combination of stream surfaces and illustrative rendering techniques proved to be a very expressive tool for the visualization of complex flow structures. It closely resembles hand-drawn flow images from renowned textbooks whose values are undisputed. The application of the methods to real world datasets showed the usefulness and applicability for the analysis of data produced by modern computational fluid dynamics simulations. Silhouettes and illustrative surface stream lines provide overview of the surfaces with their flow directions, slabs give a good orientation for surfaces with multiple layers, and Poincaré section-like slices reveal the complete folded structure of complex flow patterns. Furthermore, all techniques are completely interactive and support explorative inspection by domain scientists. We believe that the resulting images have the same inherent beauty as their hand-drawn counterparts.

In the previous section, we have already compared our methods to previous approaches. Still, we would like to emphasize two points. First, the only techniques providing a similar depth of insight into the structure of vortex breakdown bubbles is the work by Tricoche et al. [175]. They, however, used a combination of direct volume rendering and vector field topology in moving section planes, i.e. no stream surfaces, and their approach is not interactive. Second, the earlier mentioned approaches by Garth et al. [43], Krishnan et al. [86], and Hummel et al. [63] yield expressive visualizations, but they do not serve as interactive exploration tools.

As, in principle, the presented methods can be applied to any kind of surface parameterized by two variables, we will explore the usefulness for other surface types in the future. The focus will lie on streak surfaces and path surfaces, as their characteristics fit the prerequisites of our methods.

Part II

Visualization of 4D PC-MRI Blood Flow Data

Background

In this chapter, additional basics and background information for our work about blood flow analysis and illustration is provided. First, medical foundations about the cardiovascular system and some of its common diseases are given. Then, magnetic resonance imaging (MRI) and the flow-sensitive MRI in particular are explained. The latter is the source of the data used in the remaining thesis.

4.1 Medical Background

4.1.1 Human Cardiovascular System

The main function of the cardiovascular system are material transportation, protection, and regulation. It supplies the different body parts with essential nutrients and oxygen and removes carbondioxide and waste products. It protects the body by closing small wounds, cleaning up cellular debris, and fighting pathogens that have entered the body. Finally, it regulates the body's homeostasis, which means that it keeps the concentration of solvents in the blood and the body's temperature and pH value stable [153].

The cardiovascular system consists of the heart and blood vessels. The heart is a four-chambered muscular pump, which is located along the midline of the human thorax. The bottom tip is called apex and is rotated towards the left side of the chest. The top of the heart is called heart base and connects the heart to the body's large venous and arterial vessels. Veins are the blood vessels transporting blood towards the heart, and arteries are transporting blood away from the heart.

The heart is a double pump and divided into the left and the right heart (see Fig. 4.1a). It supplies two circulatory loops with blood: the pulmonary and the systemic circulation. The pulmonary circulation delivers deoxygenated blood from the right heart to the lungs for oxygenation and returns it to the left heart. From there, the blood enters the systemic circulation. The systemic circulation supplies the body with oxygenated blood from the left heart and returns the deoxygenated blood to the right heart. Therefore, the right heart always transports deoxygenated blood and the left heart transports oxygenated blood. Also, the systemic circulation shows a higher blood pressure than the pulmonary circulation, because the blood needs to be distributed in the whole body.

The two circulation loops are not separated, but the blood flows through them consecutively. However, in the heart, their flow paths are separated. More specifically, each heart side is again divided into two chambers – the right and left atrium

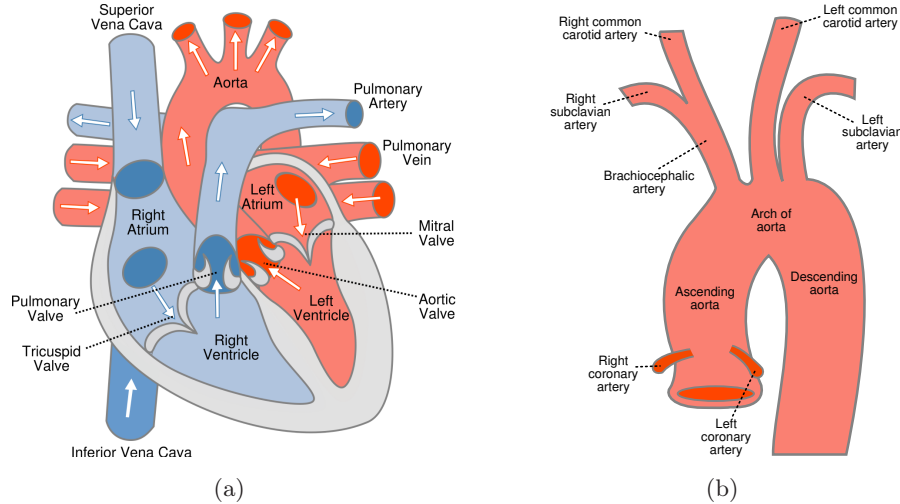


Figure 4.1: Anatomy of (a) the human heart (based on [200]) and (b) the human aorta (based on [199]).

and the right and left ventricle. The blood enters the heart through the atria (via the large veins) and leaves it through the ventricles (into the large arteries). The pumping mechanism of the heart is controlled electrically and relies on the rhythmic contraction (systole) and relaxation (diastole) of the heart chambers and the accompanying opening and closing of the heart valves. A heart beat starts with the contraction of the atria initiated by the electrical signal given by a natural pacemaker (sinoatrial node). At this time, blood already flows from the atria into the ventricles via the opened valves. The atrial contraction increases blood pressure and pushes some extra blood into the ventricles (see Fig. 4.2, middle). The electrical signal is delayed for about 100ms by the atrioventricular node and then initiates ventricle contraction. The atria relax, the valves between atria and ventricles close and the valves towards the arteries open. Due to the contraction the blood is rapidly propelled from the left ventricle into the aorta and from the right ventricle into the pulmonary arteries (see Fig. 4.2, right). Almost simultaneously, the atria are filled with blood again from the vena cava and the pulmonary veins (see Fig. 4.2, left).

The terms systole and diastole relate to different phases of the heart cycle. Systole describes the atrial filling and the ventricular contraction, which ejects the blood into the arteries. Diastole refers to the ventricular filling.

The aorta is the body's largest artery. It receives blood from the left ventricle and distributes it in the body. The aorta consists of the ascending aorta, the aortic arch, and the descending aorta. In the aortic arch, three main branchings leave the aorta: the brachiocephalic artery, the left common carotid artery, and the left subclavian artery (see Fig. 4.1b). These supply blood to the upper extremities and the head. The descending aorta carries blood through the thoracic and abdominal cavities of the body and feeds the major organs and the legs. For more information about the cardiovascular system, we refer to Schmidt & Thews [153].

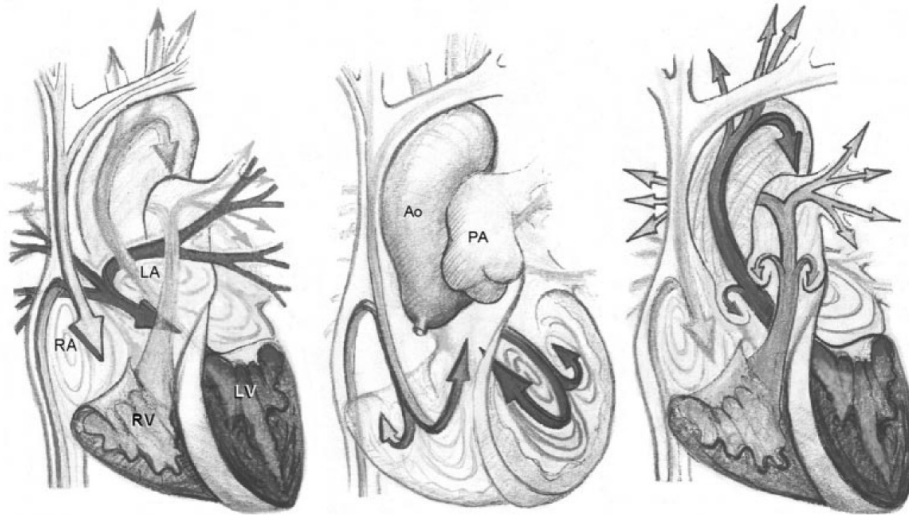


Figure 4.2: Schematic drawing of the cardiac blood flow showing the atrial filling (left), the ventricular filling (middle), and the ventricular ejection (right). The flow path through the left and right heart is depicted with dark and light arrows respectively. Abbreviations: Ao, aorta; LA, left atrium; LV, left ventricle; PA, pulmonary artery; RA, right atrium; RV, right ventricle. Reprinted by permission from Macmillan Publishers Ltd: Nature 404: Kilner et al. [81], © 2000.

In the cardiovascular system, the blood hemodynamics and the morphology of heart and vessels strongly affect each other. The asymmetric and curved geometry of the human heart leads to an efficient ejection of blood into the pulmonary arteries and the aorta [81]. On the other hand, the flowing blood exerts varying forces on the vascular morphology and causes a continuous remodeling of the heart and vessel tissue [36]. Therefore, the effective interplay of hemodynamics and morphology, where both are perfectly attuned to one another, characterizes a healthy cardiovascular system. As a consequence, this system can be destabilized by alterations of blood flow or morphology due to disease or surgery. That way, modified morphology causes changes in the blood flow behavior which may lead again to further unfavorable remodeling. In the long run, serious diseases may result from this vicious cycle [52] (see Sec. 4.1.2).

After all, information about patient-specific spatiotemporal blood flow behavior, in addition to the traditional morphological images (CT angiography, MR angiography), turns out to be valuable for diagnosis and prognosis of cardiovascular diseases. Examples for the instruments for blood flow measurement are Doppler-Ultrasound, Time-of-flight MRI or phase-contrast MRI measurements. A current overview over different cardiovascular blood flow measurements is given by Sengupta et al. [156]. However, a three-dimensional and time-resolved flow imaging modality is only given by flow-sensitive phase-contrast magnetic resonance imaging (4D PC-MRI). The basics of this imaging modality are given in Section 4.2.

4.1.2 Cardiovascular Diseases

In the following, several congenital and acquired cardiovascular diseases will be introduced, which will be of importance in the next chapters.

Fallot Tetralogy About $0.7 - 0.9\%$ of the newborns have a congenital heart defect. In 10% of these cases, the malformation is a so-called Fallot tetralogy. A Fallot tetralogy is combination of four different defects: First, a ventricular septal defect (VSD), which is a hole in the wall between the left and right heart ventricle. Given the hole is large enough, the blood pressure in the left and right heart are equal assimilate, i.e., the pressure is too high in the right and too low in the left heart. Second, a dextroposition of the aorta, which means that it receives blood from both ventricles because of the ventricular septal defect. Third, a stenosis of the right-ventricular outflow tract. Due to this, the blood flow into the lung is hindered and in combination with the VSD, not enough blood reaches the lungs to become oxygenated. Fourth, a hypertrophy (pathological enlargement) of the right ventricle. The main problem caused by this syndrome is the patient's under-supply with oxygen (cyanosis). Therefore, a Fallot tetralogy is usually surgically repaired very early in life to separate the pulmonary and the systemic loop and repair the stenosis of the right-ventricular outflow tract. Without surgical treatment, only about 25% of the patients reach the age of 10 and 5% the age of 30 [35].

Atrial Fibrillation Atrial fibrillation (AF) is the most frequent type of cardiac arrhythmia and affects about 1% of the total population and about 5% of elderly people over the age of 80. Atrial fibrillation means that the healthy sine rhythm changes to arrhythmia and the heart beats irregularly. Different severities are distinguished: (1) paroxysmal AF occurs occasionally and ends without external measures, (2) persistent AF may be stopped by medication or electrical shock, and (3) permanent AF requires intervention. Possible interventions are, for example, the implantation of a pacemaker or radiofrequency ablation. Radiofrequency ablation eliminates electrical transmission sites in the atrium and may cure atrial fibrillation completely. Atrial fibrillation is not directly harmful, but causes many patients to tolerate physical stress not very well. Further, the irregular heart beat may worsen other existent heart defects over time and may cause blood clot generation in the atria. The latter implies an increased risk for stroke [189].

Aortic Valve Diseases The aortic valve is the valve separating the left ventricle from the aorta. It consists of three pocket-like flaps (cusps) and is designed to let blood flow into the aorta during systole and to hinder backflow for the remaining time of the heart cycle. Aortic valve disease can now be related to either of these functions. Aortic stenosis is a narrowing of the aortic valve, which disrupts the blood flow into the aorta. Aortic valve insufficiency on the other hand, describes a leaky valve that allows blood to regurgitate into the left ventricle.

Aortic stenosis may appear congenitally when someone is born with only two instead of three cusps (bicuspid valve). The more frequent cause of aortic stenosis, however, is a calcification of the cusps. Consequences of aortic stenosis are a high blood pressure in the left ventricle, which causes a thickening and stiffening of the heart muscle of the left ventricle (hypertrophy). As a result, the blood pressure in the left atrium and the lungs increases as well and causes shortness of breath. Also, alterations of the aortic valves frequently cause aneurysms in the ascending aorta as secondary pathologies. These bulges develop due to the altered blood flow through the valve and a resulting high shear stress on the aortic wall [10].

Aortic regurgitation can also be caused by calcifications. This may stiffen the cusps and prevent them from closing tightly. Other causes might be a tear in the aorta or a valve infection (infectious endocarditis). The backflowing blood also causes a higher pressure in the left ventricle and has similar consequences like a stenotic valve [121]. Both defects are treated either surgically with a valve replacement or a minimally-invasive, transcatheter aortic valve implantation (TAVI) [141].

Aortic Aneurysms An aortic aneurysm describes an abnormal bulging of the aorta. Causes for aneurysms are manifold. A stenotic aortic valve may cause an aneurysm in the ascending aorta because the stenosis disturbs and changes the blood flow, which may then exert higher pressure onto the aortic wall and dilate the aorta. Also, congenital diseases, such as the Marfan syndrome often correlate with hyperelastic aortic wall tissue and favor aneurysm development. Also, smoking, high cholesterol, and high blood pressure are risk factors that correlate with aortic aneurysms. The main risk emanating from an aortic aneurysm is the rupture, which may lead to death in minutes or hours if untreated. Treatment is done surgically by either stenting the aorta to narrow the diameter or by replacing the affected aortic segment [35].

Aortic Dissection Another severe disease that may correlate with aortic aneurysms is an incomplete fissure of the aortic wall (dissection). The aortic wall consists of three main layers (intima, media, and adventitia). An aortic dissection is the rupture of the innermost layer, i.e., the intima. This allows the blood to flow into the gap between intima and media and to cause a *false lumen*. Depending on the pathological condition of the aortic wall, the blood might reenter into the real lumen further downstream. An aortic dissection implies the risk that the coronary arteries and other aortic branchings are occluded and the body (and the heart itself) are malperfused. Also the aorta might rupture. According to the Stanford classification, dissections of type A and type B are distinguished. Type A are dissections affecting the ascending aorta or the aortic arch. Type B dissections occur after the aortic arch, i.e., after the branching of the subclavian artery, and affects mainly the thoracic and abdominal aorta. Both types are serious diseases, but type A requires an immediate surgical treatment [35].

4.2 4D Phase-Contrast MRI

The development of magnetic resonance imaging (MRI) dates back to the early 1970s when Paul C. Lauterbur and Sir Peter Mansfield started to utilize the physical effect of *nuclear magnetic resonance* for the purpose of tissue imaging. By now, MRI is not only used for different kinds of morphological scans but also for functional measurements of the brain activity (functional MRI) [69], measurement of water diffusion to detect nerve fiber tracts (diffusion tensor imaging) [23, 67], elastographic measurements (MR elastography) [49], and finally the measurement of blood flow velocity (4D phase-contrast MRI) [110]. Data acquired with the latter flow-sensitive MRI will play an important role in the next chapters. In this section, we describe the basics of MRI in general and the mechanisms of 4D PC-MRI measurements in particular.

In literature a variety of introductions to MRI exists [55, 60, 69, 115]. In these textbooks, the authors mainly use a mixture of quantum and classical mechanics theory to cover MRI. For the sake of clarity and since quantum mechanics is not strictly necessary to explain MRI [53], this description will fully rely on classical mechanics theory. It will follow the lines of Hanson [54] and McRobbie et al. [115]. For more information about the quantum mechanical details of MRI, please refer to the previously mentioned textbooks.

4.2.1 Introduction to MRI

When scanning a human body with magnetic resonance imaging (MRI), different tissue types can be distinguished due to their different magnetic properties. These magnetic properties depend primarily on the tissue-specific density of hydrogen atoms and the tissue's compactness. MRI can measure these properties by utilizing *magnetic resonance*.

Magnetic Resonance Magnetic resonance is a physical phenomenon in which atomic nuclei can absorb and emit measurable electromagnetic signals. On a macroscopic scale, this effect can be observed with a simple compass needle. A compass needle is a magnetic dipole, which aligns with the earth's gravitational field. By bringing an additional magnet close to the needle, it can be deflected (i.e., rotated) from the north direction (see Fig. 4.3a). After removing the magnet, the compass needle will continue to move back and forth through the north direction and gradually reach the equilibrium position again. During this phase, the needle itself will emit an oscillating magnetic field (radio waves). The emitted radio waves contain information about the amplitude and the frequency of the needle's movements and can be measured with an antenna.

By moving the magnet back and forth with the needle movement's frequency (perpendicular to the north direction and the compass needle, see Fig. 4.3a), a strong oscillation of the needle can be achieved, i.e., needle and magnet are in magnetic resonance. The specialty of this effect is that when the needle is activated

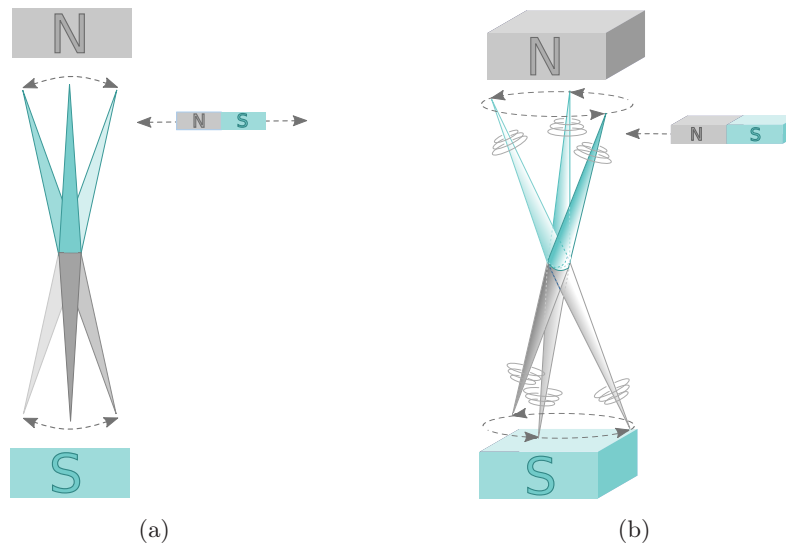


Figure 4.3: Magnetic resonance by means of a compass needle. (a) Without external influences, a compass needle aligns itself with the gravitational field. Magnetic resonance is achieved when an external magnetic field moves with the needle's inherent resonance frequency. (b) Assuming the previous needle rotates around its main axis, the activating magnetic field causes precession, i.e., instead of moving back and forth through the north direction, the needle rotates around the north direction. This behavior is similar to the behavior of spinning hydrogen protons.

with its inherent resonance frequency, a weak magnetic field and small oscillations suffice to achieve a comparably strong needle oscillation (i.e., large amplitudes). This behavior can be compared to a child on a swing that can be kept in motion with light pushes as long as they occur at the swinging frequency.

The basic principle of MRI is to bring magnetic dipoles into magnetic resonance and to measure the signals emitted when the equilibrium is rebuilt. The measured signals contain all information for imaging different tissue, flow velocities, diffusion etc. in the body. The magnetic dipoles that used here are hydrogen nuclei. A hydrogen nucleus contains one spinning proton. The proton spin is a (quantum mechanical) property, which means that the proton rotates around its central axis. This causes the proton to be a magnetic dipole directed along the spinning axis. Protons and compass needles are both magnetic dipoles and behave rather similarly in an external magnetic field, but with one slight difference. Assuming a needle that rotates around its length like a proton and again receives a single activation by a magnetic field like the compass needle earlier, it would not start to move back and forth through the north direction, but would start to precess around it with a specific resonance frequency. So, the magnetic dipole performs two different rotations (spin and precession) around two different axes now (see Fig. 4.3b).

Magnetic Resonance in the Human Body With MRI, we are not interested in single hydrogen atoms, but consider the magnetic behavior of a larger number of hydrogen atoms at once.¹ Without an external magnetic field, a population of protons is oriented randomly in space. When a body is moved into an MRI scanner, it is applied to a very strong magnetic field (which we call B_0). Against the expectations, the protons in the body's tissues do not align fully to B_0 . Instead, the influence of neighboring magnetic protons onto each other is stronger than the influence of B_0 (see Fig. 4.4a). Therefore, during equilibrium only a small number of protons² is directed along B_0 . Protons that are not fully aligned, precess around B_0 – similar to the spinning needle mentioned previously. The precession frequency is called Larmor frequency ω_L . It is proportional to the magnetic field strength B_0 and depends on the atom-specific gyromagnetic factor γ :

$$\omega_L = \gamma B_0 \quad (4.1)$$

For hydrogen protons, the Larmor frequency is $\omega_L = 42.48 \text{ MHz}$ with $B_0 = 1 \text{ T}$. This is the frequency that an activating magnetic field needs to oscillate with to achieve nuclear magnetic resonance of hydrogen protons.

For simplicity, we will examine the net magnetization M instead of single magnetic dipoles from now on. The net magnetization is the sum of the magnetic moments (vector values) of all protons. Here, the longitudinal and the transverse components of M are distinguished. The longitudinal magnetization M_z is aligned with B_0 and the transverse magnetization M_t is perpendicular to B_0 and rotates in the xy-plane of the considered reference coordinate system.

In the equilibrium state, a longitudinal magnetization with $M_{z,e} > 0$ exists, because of the small overhead of protons directed at B_0 (see Fig. 4.4b). There is no transverse magnetization $M_{xy,e}$ during equilibrium, since the protons do not precess in phase and their transverse components of the magnetic moments sum up to zero. MRI scanners detect only the oscillating signal caused by the transverse magnetization. Thus, no signal is measurable during equilibrium.

Magnetic resonance of this proton population is achieved with a radio-frequency (RF) pulse rotating with Larmor frequency perpendicular to B_0 . This RF pulse is the equivalent to the oscillating magnetic dipole that was previously used for activation in the compass needle example. In the MRI scanner, this pulse is created by means of an RF coil, which is placed around the body part of interest (head coils, knee coils etc.). In order to visualize the effects of an RF pulse more easily, we choose the perspective of the rotating reference frame. This means, we consider the coordinate system to be rotating at Larmor frequency in the same direction as the protons. The spinning protons and an eventually precessing net magnetization

¹ Advantage of this: The behavior of a larger number of hydrogen atoms can be explained with classical mechanics, whereas explaining the behavior of a single proton in a magnetic field would necessitate quantum mechanics.

² The number of protons that align exactly with B_0 depends on the magnetic field strength and the temperature and is described by the Boltzmann factor. At body temperature, 3 out of one million protons (3 ppm) are aligned with B_0 if $B_0 = 1 \text{ T}$.

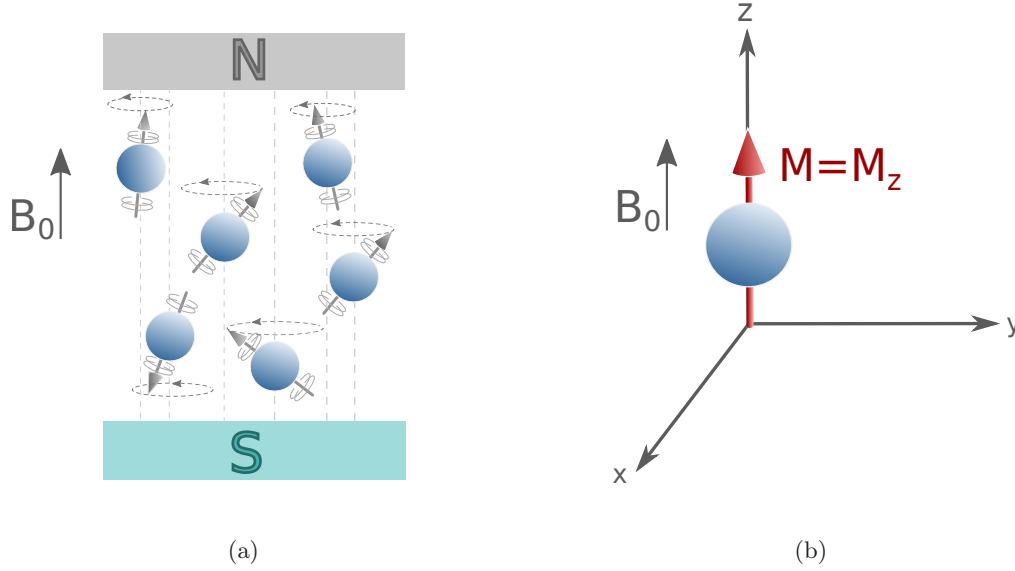


Figure 4.4: Equilibrium. (a) A population of hydrogen protons does not fully align to the strong magnetic field B_0 . They are orientated almost randomly and precess around the direction of B_0 . (b) The net magnetization M is positive, since a slight overhead of protons is aligned with B_0 . M consists only of longitudinal magnetization M_z , since the protons precess asynchronously ($M_{xy} = 0$).

or RF pulse appear stationary. In this rotating frame of reference, the RF pulse produces a static magnetic field B_1 perpendicular to B_0 (see Fig. 4.5a). As a result of the RF pulse, the net magnetization M undergoes a torque perpendicular to B_1 and is rotated away from the equilibrium position. The longer the RF pulse lasts and the stronger it is, the further the net magnetization M deflects and the stronger the decrease of M_z becomes. Another consequence of the RF pulse is that the hydrogen nuclei are synchronized and start to precess in phase. This results in an increase of M_{xy} . In the rotating frame of references M_{xy} also appears stationary. The decrease of M_z and increase of M_{xy} depends on the RF pulse's strength and duration. Most commonly, 90° RF pulses are applied. After this pulse, M_z is 0 and M_{xy} is maximal. A 180° RF pulse is either twice as long or twice as strong as a 90° pulse and results in M_z to be directed anti-parallel to B_0 .

In Figure 4.5b, the same process is shown for the rotating frame of reference. Here, the deflection of M occurs in spiraling manner because M_t precesses around the z -axis representing the synchronized protons.

After the RF pulse, the MRI scanner can detect radio waves emitted by the precessing proton population. The signal is proportional to the transverse magnetization M_{xy} , which is equivalent to the circle radius in Figure 4.5b. The longitudinal magnetization has no influence on the signal.

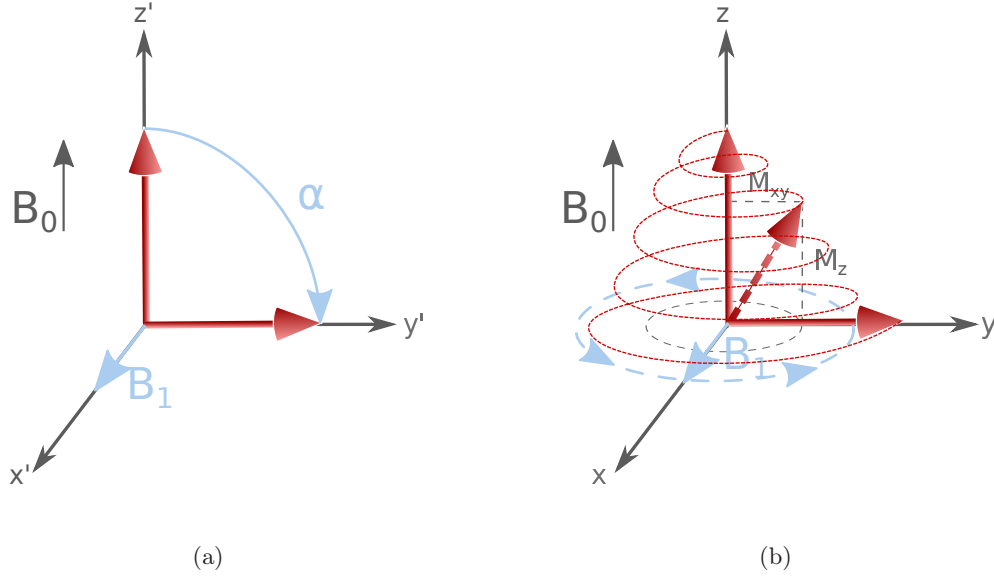


Figure 4.5: The effect of a 90° radio-frequency pulse in a stationary (a) and a rotating reference frame (b). (a) The RF pulse results in a second magnetic field B_1 that rotates in the xy -plane around B_0 . As a result the protons precess in phase, M_{xy} increases and precesses itself around B_0 . At the same time, the longitudinal magnetization decreases. Altogether M spirals away from its equilibrium position towards the xy -plane. (b) In the rotating reference frame, B_1 appears stationary directed at x' . M shows no spiraling behavior, but is rotated around the B_1 -direction by 90°.

Relaxation After the RF pulse, the previous magnetic equilibrium is rebuilt. This mechanism is called relaxation and is mainly caused by interaction and collisions between neighboring nuclei. Two independent relaxation processes occur. First, the protons lose phase coherence because neighboring magnetic protons accelerate or decelerate each other. This leads to slightly differing precession frequencies and a gradual dephasing (*spin-spin relaxation*). The dephasing causes a decrease of the transversal magnetization M_{xy} and proceeds more quickly the more compact the tissue is.

The rebuilding of the longitudinal magnetization M_z is caused by a second process, which involves random interactions between nuclei in which the resonance condition is met (*spin-lattice relaxation*). Then, a torque moves the protons' magnetic moments towards equilibrium position and therefore increases M_z . As a side effect M_{xy} decreases also. Spin-lattice relaxation is fastest for semi-firm tissue and takes longer for liquid and very compact matter.

These processes are described by two time scales: T_1 describes the increase of the longitudinal magnetization. T_2 describes the reduction of the transversal magnetization. Different tissue types have specific T_1 and T_2 values and can therefore

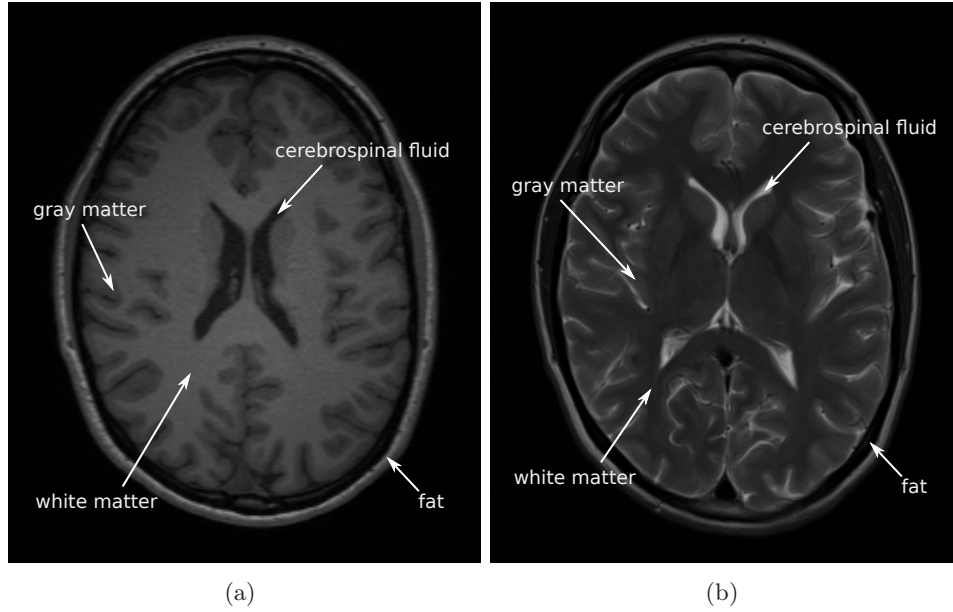


Figure 4.6: Axial MRI image slices of the human brain with T_1 -weighting (a) and T_2 -weighting (b). (a) The shorter the T_1 of a tissue, the brighter it appears in a T_1 -weighted image. The T_1 value of a tissue decreases with its compactness. Thus, rather firm tissue such as fat, gray and white brain matter appear brighter, whereas the cerebrospinal fluid in the ventricles appears dark. (b) The longer the T_2 of a tissue, the brighter it appears in a T_2 -weighted image. T_2 is short for semi-firm tissue, such as gray and white matter, which appear dark gray. T_2 is longer for fluids and very dense matter. Thus, the cerebrospinal fluid appears bright. Bone appears dark in both weightings because it contains a low number of hydrogen protons.

be distinguished in the final image. With MRI, the resulting image can be weighted to have either a higher T_1 - or a higher T_2 -contrast (see Fig. 4.6).

Spin Echo Sequence An MRI measurement is a repeated sequence of the previously explained RF pulses for proton excitation, a time period of relaxation and dephasing, and the measurement of the proton's echo signal as an AC current while the equilibrium is rebuilt. The time interval between RF pulse and the measurement is called echo time TE . The time interval from one RF pulse to the next is called repetition time TR . By adjusting TE and TR of an imaging sequence, the T_1 - and the T_2 -contrast can be controlled. For example, a T_1 -weighted image is achieved with short TE and TR , whereas for a T_2 -weighted image long TR and TE are used.

Dephasing also occurs due to inhomogeneities of B_0 . A magnetic field with an inhomogeneous strength across the scanner results in different ω_L . Due to this, protons spin at different frequencies. Some protons precess faster and some slower and they get quickly out of phase. This unwanted loss of signal occurs at time

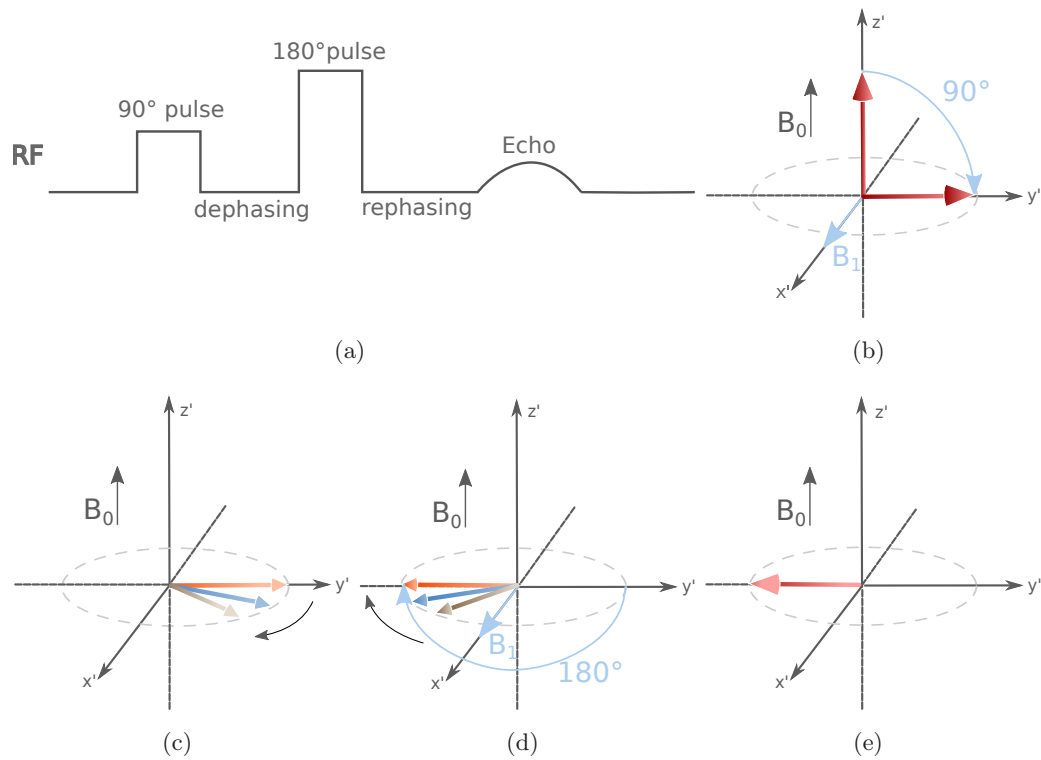


Figure 4.7: Spin echo sequence. (a) Sequence overview. (b) The protons are activated by a 90° RF pulse. (c) Field inhomogeneities cause some protons to precess more quickly (brown arrow) than others (orange arrow). (d) After the 180° RF pulse, the protons' order is reversed, i.e., faster precessing protons that were ahead are now lagging behind (brown arrow) and vice versa. (e) The protons have rephased and the signal echo is measured.

scale T_2^* . A so-called *spin-echo sequence* can solve this problem. Assuming that the inhomogeneities are constant over time, the idea of a spin echo is to revert the dephasing effects by applying a refocussing pulse. More specifically, as shown in Figure 4.7, the sequence starts with a 90° RF pulse. During relaxation, the T_2^* -processes lead to tissue-unspecific dephasing: field inhomogeneities cause varying precession frequencies. Due to this, some protons precess more quickly (brown arrow) than others (orange arrow) and gradually, the protons dephase and M_{xy} decreases. The refocussing is carried out with a 180° RF pulse, which turns the protons by 180° around the B_1 direction. As a result, the faster rotating protons are now behind and the slower ones are in front. However, the precession frequency and direction of the protons stays the same. Thus, after the same time (as between first and second RF pulse), the spins are in phase again and the echo signal can be measured. This signal is free of the unwanted T_2^* influence.

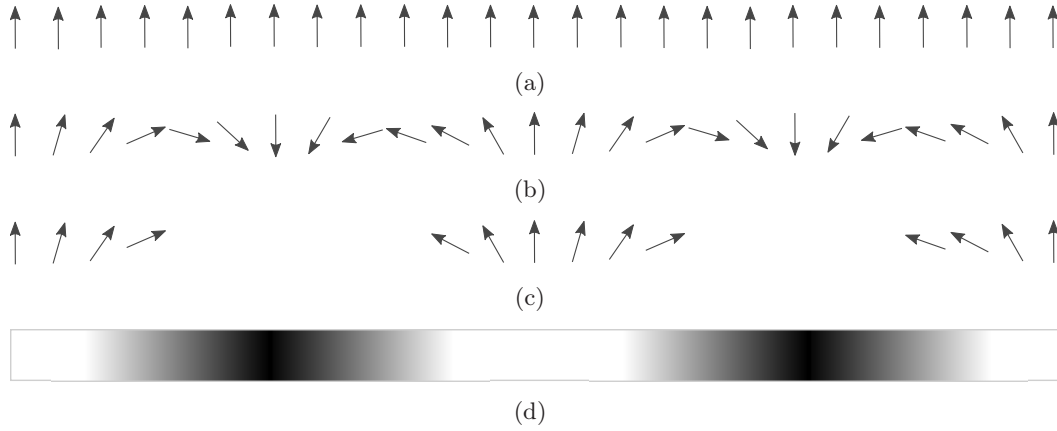


Figure 4.8: Spatial encoding within a slice. The arrows represent the protons' phases, i.e., only their rotation in the xy -plane is shown. (a) After slice selection the protons are all in phase. (b) A field gradient along the row of arrows causes a phase roll. In homogeneous tissue, the net magnetization in the xy -plane (M_{xy}) and the signal diminishes. (c) In inhomogeneous tissue, a signal can be measured. The signal magnitude depicts how well the tissue structure matches the stripe pattern (d) induced by the gradient. Redrawn based on [54].

Spatial Encoding and Image Reconstruction The MRI signal that is detected as an AC current with a receiver coil, is caused by an oscillating transverse magnetization. It contains information about the strength of the magnetization and its phase, but is emitted by the complete population of activated protons. For an imaging technique, the signals at specific locations need to be distinguishable.

For a three-dimensional localization of the signal source, additional weaker gradient magnetic fields are used. A first gradient causes a slight change of the magnetic field strength along B_0 . Consequently, the Larmor frequency differs along B_0 as well and the RF pulse excites only a narrow slice of the body. This limits the source of the measured signal to this area.

A spatial localization within the slice necessitates two further gradients. After excitation, all protons precess in phase. Applying gradients in x - and y -direction within the slice makes the protons precess with slightly increasing frequencies in these directions and hence lets them lose phase coherency over time. This means that the gradients cause a *phase roll* (see Fig. 4.8). In homogeneous tissue, the phase roll eliminates the signal because the net magnetization in the xy -plane (M_{xy}) is nullified. If, however, the imaged structure is composed of tissue that shows a lower transverse magnetization in areas with protons precessing in about same phase, the net magnetization does not diminish. A signal can be measured and its magnitude relates to how well the tissue structure matches the stripe pattern induced by the gradient. In Figure 4.8, the case for a one-dimensional gradient (in either x - or y -direction) is shown. With MRI, two gradients with differing strengths and durations are used to create 2D stripe patterns of varying frequencies and tilts.

In inhomogeneous tissue, however, the phase roll might match the tissue structure in the sense that some areas send no signal and the remaining protons do emit signals since they precess in phase (see Fig. 4.8). In other words, the signal contains information about how well the tissue matches a stripe pattern with a specific width and angle. By varying the gradients, stripe patterns of different spatial frequencies and tilts can be tested for. The results are stored in *k-space*. Every point in *k-space* corresponds to a specific stripe pattern. The final image is generated by a Fourier transform of the *k-space*. This is equivalent to combining the different stripe patterns and weighting them by the corresponding signal strength.

The sequence in which the *k-space* is filled, is determined by the used measurement sequence. With the classical spin-echo sequence, described earlier, one line in *k-space* is measured with one RF excitation pulse. With echo planar imaging (EPI), one RF pulse is sufficient to fill the complete *k-space* and acquire a complete MR image.

4.2.2 Velocity Measurement with 4D PC-MRI

MRI is traditionally used to image tissue morphology. The measuring sequences are based on the assumption that the hydrogen protons are stationary. However, MRI is sensitive to tissue motion caused by body movement, breathing, or fluid flow. Moving protons cause phase differences that often appear as unwanted blurring or ghosting artifacts in the final MRI image. During blood flow measurement, however, one can utilize this inherent motion sensitivity of MRI.

The MRI signal not only contains magnitude, but also phase information. Similar to the magnitude images that are usually reconstructed from *k-space*, a phase image can be generated. Therefore, the basic idea of velocity measurement is to apply an MRI sequence that encodes the velocity of tissue, blood or the like in the signal's phase information. The following explanations are based on Markl et al. [106] and Lotz et al. [102].

Velocity Encoding The Larmor frequency ω_L depends on the gyromagnetic ratio γ and the applied magnetic field strength (see Eq. 4.1). During a real-world MRI scan, not only B_0 contributes to the field strength, but also local field inhomogeneities and gradients. Therefore, the Larmor frequency is proportional to the combined field strength caused by the static magnetic field B_0 , local field inhomogeneities ΔB_0 , and the spatial encoding gradient \vec{G} . Thus, at a spatial position \vec{r} and time t , the Larmor frequency is:

$$\omega_L(\vec{r}, t) = \gamma B_0 + \gamma \Delta B_0 + \gamma \vec{r}(t) \vec{G}(t) \quad (4.2)$$

The phase ϕ at location \vec{r} and time TE is the time integral of ω_L , as the phase depends on the frequency and the time period the rotation lasted. Without loss of

generality, the time period $[0; TE]$ is considered:

$$\phi(\vec{r}, TE) = \int_0^{TE} \omega_L(\vec{r}, t) dt = \gamma \int_0^{TE} (B_0 + \Delta B_0 + \vec{r}(t) \vec{G}(t)) dt \quad (4.3)$$

The signal is transformed into a reference frame rotating at $\omega_{0,L}$ (the Larmor frequency caused by B_0). Due to this, the influence of B_0 on the signal can be neglected and omitted in the remaining equations.

For velocity imaging, the movement of protons needs to be modelled and handled. The proton position \vec{r} over time can be described with a Taylor series with initial time t_0 :

$$\vec{r}(t) = \vec{r}(t_0) + \vec{v}(t_0)(t - t_0) + 0.5\vec{a}(t_0)(t - t_0)^2 + \dots \quad (4.4)$$

Further, we assume that the measurement time TE is short enough, so that we can consider tissue motion to be constant (i.e., $\vec{a} = 0$ and constant \vec{v}). This simplifies the proton movement over time in Equation 4.4 to

$$\vec{r}(t) = \vec{r}_0 + \vec{v}_0 t \quad (4.5)$$

with initial position \vec{r}_0 and constant velocity \vec{v}_0 , and $t_0 = 0$. Using this expression in Equation 4.3 gives the following:

$$\phi(\vec{r}, TE) = \gamma \int_0^{TE} \Delta B_0 dt + \gamma \int_0^{TE} (r_0 + \vec{v}_0 t) \vec{G}(t) dt \quad (4.6)$$

$$= \phi_0(\vec{r}) + \underbrace{\gamma \vec{r}_0 \int_0^{TE} \vec{G}(t) dt}_{\vec{M}_0: \text{ static spins}} + \underbrace{\gamma \vec{v}_0 \int_0^{TE} \vec{G}(t) t dt}_{\vec{M}_1: \text{ moving spins}} \quad (4.7)$$

This resulting equation now demonstrates that three components influence the phase shift: the initial background phase ϕ_0 , the phase shift due to stationary and the phase shift due to moving protons. The two integrals M_0 and M_1 specify the field gradients' contributions to the phase shift during the respective time period. For flow imaging M_1 is of interest, which is the first gradient moment and describes the signal phase shift induced by the protons' velocities. Therefore, an MRI sequence needs to be applied that eliminates the other two effects in the MRI signal and encodes the velocities of the moving protons. The effect of the stationary spins on the phase shift can be neutralized with a bipolar gradient. Using a bipolar gradient means that two gradients with the same strength and duration, but opposite amplitude are used after one another. As shown in Figure 4.9, a balanced bipolar gradient results in no phase shift for the stationary protons, as the second fully reverses the phase shift of the first gradient. The moving protons, however, experience different gradient field strengths over time and accumulate a phase shift that is proportional to their velocity.

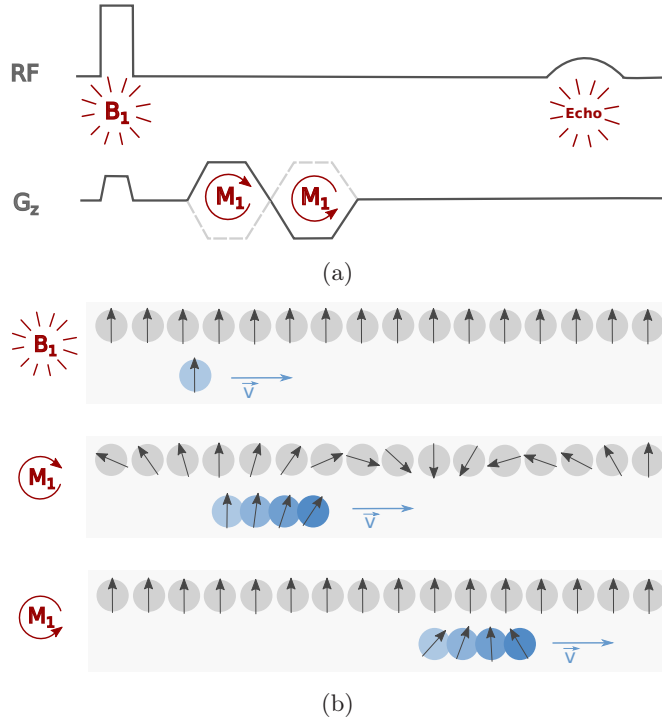


Figure 4.9: Bipolar gradients for velocity encoding. (a) Sequence of RF pulse and gradient application and (b) the corresponding effects on stationary (gray) and moving protons (blue). The arrows represent the proton phases in the xy-plane. After B_1 , all protons are in phase. The application of the first gradient M_1 causes a phase roll in the protons, which is reversed only for the stationary protons by the second gradient M_1 (reversed polarity). The moving protons experience a phase shift, which correlates with their velocity along the gradient direction. Figure based on [102].

The phase shift at position \vec{r} after the application of the bipolar gradient is

$$\phi(\vec{r}, TE) = 2\phi_0(\vec{r}) + \gamma\vec{v}_0\vec{M}_1 \quad (4.8)$$

with $M_1 = -G\frac{T^2}{2}$. However, the influence of the unknown background phase ϕ_0 is not eliminated yet with bipolar gradients because they are not appropriate to refocus phase shifts caused by field inhomogeneities (see Sec. 4.2.1). This problem is solved by measuring two velocity-encoded phase images with two different first gradient moments $\vec{M}_1^{(1)}$ and $\vec{M}_1^{(2)}$ and subtracting these images. This results in the following phase difference:

$$\Delta\phi(\vec{r}, TE) = \left[2\phi_0(\vec{r}) + \gamma\vec{v}_0\vec{M}_1^{(1)} \right] - \left[2\phi_0(\vec{r}) + \gamma\vec{v}_0\vec{M}_1^{(2)} \right] \quad (4.9)$$

$$= \gamma\vec{v}_0(\vec{M}_1^{(1)} - \vec{M}_1^{(2)}) \quad (4.10)$$

$$= \gamma\vec{v}_0\Delta\vec{M}_1 \quad (4.11)$$

From this subtraction image the velocity magnitude can be determined by simply dividing the pixel value by $\gamma\Delta M_1$. These velocities correspond to the motion component along the direction of the currently used first gradient moment. Thus, to gain information about the three-dimensional velocity, three subtraction images need to be produced – one for each spatial dimension. The straight-forward measurement would result in six independent measurements with different bipolar gradient combinations. However, it is possible to acquire one reference image with a flow compensation sequence, which refocusses all stationary and moving spins after TE and thus only contains background phase information. Since this reference image can be used to generate the subtraction images for all three separate directional velocity measurements, the total number of imaging cycles is reduced to four. This also results in a shorter acquisition time.

Equation 4.11 shows that the moving protons' phase shifts are proportional to the gradient moment M_1 , i.e., to the duration and strength of the gradient. Therefore, the phase signal strength can be controlled with these two parameters. As previously mentioned, the velocity signal strength can be controlled by gradient strength and duration. This allows to adjust the acquisition parameters to the occurrent velocities in the imaged structure. The largest phase differences that can be measured are $-\pi$ or π respectively. They represent the maximal measurable velocity magnitude (v_{enc} , velocity encoding, see Fig. 4.10a). For a blood flow measurement in the heart a v_{enc} of $100 - 150 \frac{cm}{s}$ is commonly used [9].

The described procedure captures the velocity information for one point in time and needs to be repeated several times to capture a complete heart cycle. The phase-contrast measurement takes too long to image the beating heart in a useful temporal resolution. Therefore, several heart beats are captured at different stages and from the resulting measurements the velocities of an averaged heart cycle are derived. To synchronize these measurements, the image acquisition is ECG-gated. Additionally, time-resolved morphological scans are recorded.

Finally, the result are a morphology sequence and three velocity sequences (one for each spatial dimension) covering one averaged cardiac cycle. An example is shown in Figure 4.11. Typically, the temporal resolution is around 40ms which leads to 16 – 25 time steps for one heart beat (which takes about 800 ms). The spatial resolution in a slice is usually in the range of 1.6 – 2.0 mm. The slice distance is usually larger and around 2.5 – 3.5 mm. The acquisition takes about 10 – 20 min with $TE = 2 - 4 ms$ and $TR = 5 - 7 ms$. The final acquisition time depends on the patient's heart and respiration rate. For more information on data characteristics associated with the imaging of different structures, we refer to Barker et al. [9].

Limitations and Data Artifacts Just like every MRI sequence, 4D PC-MRI is prone to various sources of inaccuracies and limitations. Oftentimes one needs to find a satisfactory trade-off between several drawbacks when choosing the measurement parameters or apply appropriate postprocessing methods.

One such crucial parameter is the previously described velocity encoding v_{enc} . It

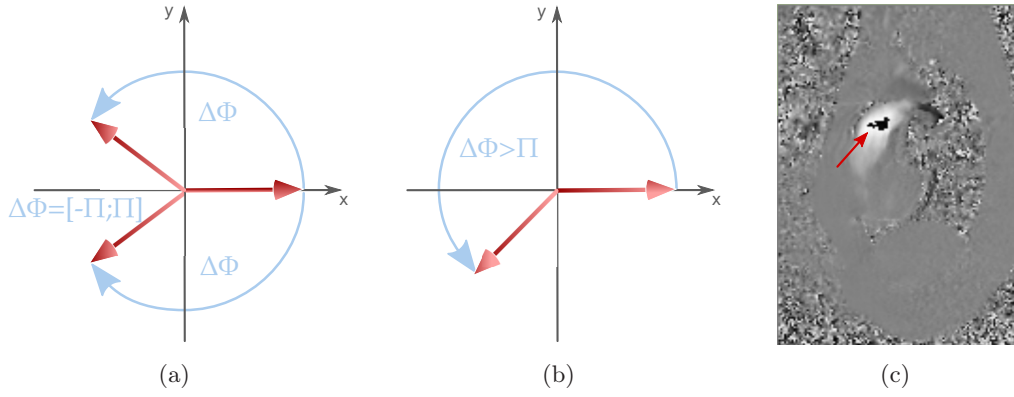


Figure 4.10: Phase shift and velocity encoding (v_{enc}). (a) The measurable phase shift is between 0 and $-\pi$ for velocity against gradient direction and between 0 and $+\pi$ for velocities in gradient direction. The v_{enc} defines the maximal measurable velocity magnitude which corresponds to $-\pi$ and π respectively. (b) If the v_{enc} is chosen too low, higher velocity magnitudes result in larger phase shifts and cause a phase wrapping. (c) These large velocities appear in the flow image as velocities in the opposite direction (red arrow). Figure based on [106]

should be chosen according to the highest expected velocity. If higher velocities do occur, this leads to higher phase differences, which cannot be represented correctly. For example, with $v_{enc} = 100 \frac{cm}{s}$ a phase difference of $-\frac{\pi}{2}$ corresponds to $v = -50 \frac{cm}{s}$. However, if a higher velocity magnitude of, for example, $150 \frac{cm}{s}$ occurs, this would cause a phase difference of $\frac{3\pi}{2}$, which cannot be distinguished from the previous value (see Fig. 4.10b). The result is phase aliasing, which means that the high velocity appears as a lower velocity in the opposite direction in the final image (see Fig. 4.10c).

Choosing v_{enc} too high in order to prevent phase aliasing may increase noise, however. Like every MRI sequence, 4D PC-MRI contains noise as well. The noise in these images is given by the standard deviation σ_ϕ of the phase differences in areas where no flow takes place (e.g., in the background). σ_ϕ is inversely dependent on the signal-to-noise-ratio (SNR) in the respective magnitude images of the morphology. This results in a noise of the derived velocity images of

$$\sigma_v = \frac{\sqrt{2}}{\pi} \frac{v_{enc}}{SNR} \quad (4.12)$$

Therefore, a high v_{enc} implies higher noise and decreases the sensitivity to the real velocity signals. Altogether, the choice of v_{enc} strongly influences the imaging quality of 4D PC-MRI [106, 129].

Another potential source of inaccuracy is that the phase-contrast sequence is based on the assumption of constant velocity. However, this assumption is not always fulfilled in the cardiovascular system where blood does accelerate, e.g., when it is pumped from the left heart ventricle into the aorta. Pelc et al. [129] argues

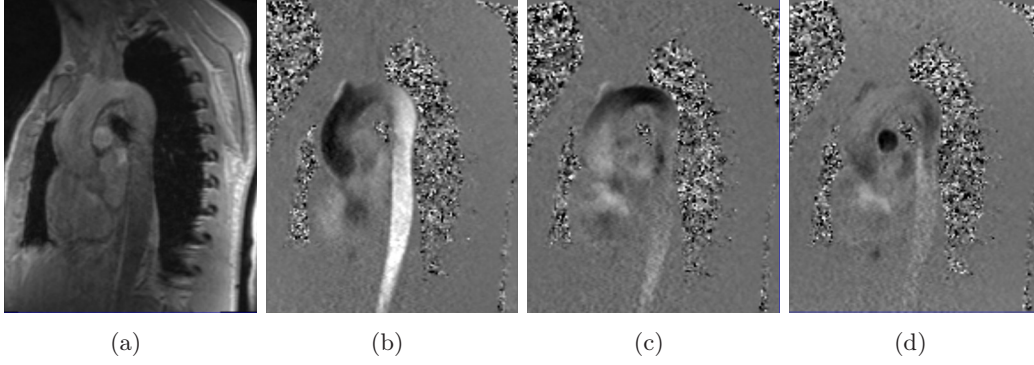


Figure 4.11: 4D PC-MRI measurement of the heart of a healthy volunteer. A timestep during systole is shown. A 4D PC-MRI scan delivers a morphological (a) and three phase-contrast images (b-d) for each timestep. Each phase-contrast image represents the flow velocity in one spatial dimension. The gray values range from black ($-v_{enc}$) to white ($+v_{enc}$). Light gray represents areas with no flow. In (b) velocity in inferior-superior direction is shown. Black and white represent high velocity of blood flowing upwards and downwards respectively. The other images depict velocities in the directions (c) anterior-posterior (front and back) and (d) left-right.

that the errors introduced during spatial encoding (spatial mismapping between slice selection and in-slice encoding) is more severe than the effects of acceleration. Still, by using higher gradient amplitudes, this effect can be lessened. More details are given in [129].

When a voxel contains both stationary and moving spins, partial volume effects emerge. The resulting velocity should represent the velocity of the moving spins weighted by their volume fraction in the voxel. However, at vessel walls moving spins often experience a flow-related enhancement and the flow velocity is overestimated. A higher spatial resolution or the use of smaller flip angles are measures to alleviate these inaccuracies. Again, these measures have drawbacks. Higher spatial resolution is only achieved with longer acquisition times and lower flip angles increase noise. An experienced radiologist is needed to find the best trade-off.

More details on the accuracy and data artifacts of MRI and 4D PC-MRI can be found in Pelc et al. [129] and McRobbie et al. [115]. With 4D PC-MRI one quality measure is the plausibility of integral lines. A low-quality flow measurement, therefore, stands out by a especially large number of integral lines leaving the blood-carrying structures [110].

Visual Analysis of 4D PC-MRI Blood Flow with Line Predicates

As already mentioned in the previous chapter, a healthy cardiovascular system depends on the effective interplay of blood hemodynamics and the morphology of heart and vessels: On the one hand, the particular shape of the human heart allows the efficient supply of blood to the lungs and the body [81]. On the other hand, the flowing blood cause varying shear stress on the vascular morphology. These forces causes a necessary and continuous remodeling of the heart and vessel tissue [36]. However, alterations in either morphology (e.g., due to surgery) or hemodynamics (e.g., due to high blood pressure) may destabilize the cardiovascular system and may lead to serious diseases. As an example, patients with a damaged mitral valve suffer from a backflow of blood from the left heart ventricle into the left atrium. This backflow decreases the heart's pumping efficiency and causes an increased blood pressure and volume in the left atrium and the lung veins. Over time, the body compensates for this altered blood flow by a dilatation of the left atrium, the veins, and finally also of the left ventricle. By then, a vicious cycle has started: The changed heart chamber morphologies again impair the valve that becomes even more insufficient and further decreases the heart efficiency until surgery is unavoidable [35].

After all, it becomes obvious that cardiovascular diseases cannot be reduced to morphological aspects. In most cases blood flow aspects are also important for diagnosis and treatment. While three-dimensional imaging of morphology has a long tradition (CT angiography, MR angiography), less regard is paid to patient-specific blood flow information. In clinical routine, flow examinations are usually, if at all, carried out in 2D with either Doppler-Ultrasound or 2D phase-contrast MRI measurements [156]. Flow-sensitive phase-contrast magnetic resonance imaging (4D PC-MRI) now allows 3D flow imaging and closes this gap. 4D PC-MRI allows to acquire time-resolved blood flow velocities in 3D over the complete cardiac cycle with a rather good spatial resolution (see Sec. 4.2). This 4D nature allows for a flow analysis at every desired location. This decreases the risk of missing crucial information as it may occur with 2D flow imaging techniques because of an adversely positioned 2D measurement planes.

However, to exploit the full potential of 4D PC-MRI, appropriate analysis methods are essential. Currently, medical doctors analyze 4D PC-MRI data mainly by inspecting the flow field or derived flow parameters on manually defined 2D planes (see Fig. 5.1). For a 3D impression of the flow dynamics, they examine the course of stream lines or path lines started from these 2D planes [9]. Valuable insight in car-

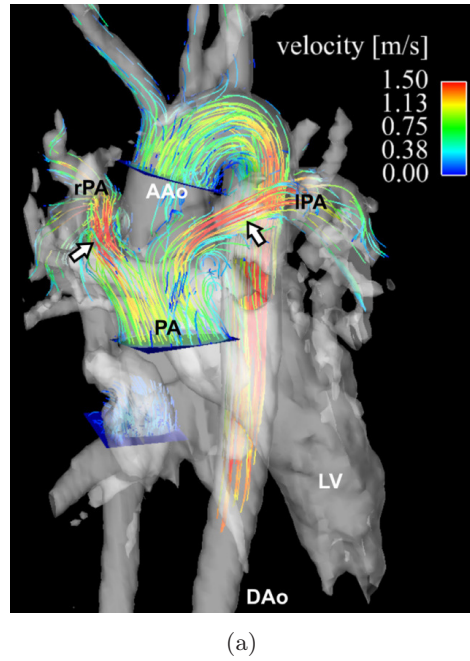


Figure 5.1: Analysis with current software tools mainly consists of the examination of flow parameters on 2D cross-sections and stream lines and particles traces seeded from these planes. Figure from Markl et al. [110], licensed under CC BY 2.0

diovascular physiology, pathologies, and associated hemodynamics have been gained with these approaches [110]. However, for larger patient studies or the clinical routine, these approaches do not suffice. Their drawback is the high degree of user interaction that makes the analysis very time-consuming and the results strongly user-dependent. Further, the current methods fail to provide a good overview of the flow situation. The medical experts are not always interested in a confined anatomical area, but want to detect irregular flow behavior at arbitrary locations in the heart or vessels. To gain proper overview, numerous 2D planes need to be selected and examined. This is tedious and the risk of missing important aspects is high – especially with complex flow situations as they occur in the human heart.

In this chapter, we start to attend to the problem of lacking overview. Our aim is to provide an analysis method for 4D PC-MRI data, which captures general flow structures instead of local flow patterns. The flow structure is the division of a flow field into areas with similar characteristics. A flexible approach of determining this flow structure are *line predicates* which have been introduced before for the analysis of non-medical CFD simulation data [147, 148]. We utilize and adapt this method to blood flow analysis in measured and therefore noisier 4D PC-MRI data.

The basic idea is that the entity of integral lines represents the spatiotemporal behavior of a flow field. However, the need to inspect thousands of lines is not helpful when trying to understand the blood flow. Instead, we present a method that allows the user to carve out interesting flow patterns by interactively selecting

and displaying lines according to relevant flow properties. For the selection process, we adapted the previously mentioned line predicate approach. Based on expert input, we defined several line predicates for steady and unsteady blood flow analysis to sort the lines into groups with a certain flow behavior related to flow paths, vortices, velocity, or residence time.

In blood flow analysis, usually more than one or combinations of these flow properties are of interest. By choosing the appropriate predicates, the users can bring in his medical knowledge and query the specific data according to their current interest. With our method, it is possible to explore the flow by answering questions like: “Where are the main vortices located and which areas are influenced by them?”, “Are there any high-velocity flow jets occurring during the cardiac cycle?”, or “What part of the blood flows from the left ventricle into the left subclavian artery?”.

We both consider the analysis of blood flow in the aorta [16] and the human heart [15]. Whereas the aorta can be handled comparably easily because of a simpler blood flow and a more rigid shape, the heart holds several challenges. Its shape changes severely during contraction and the flow is more complex (see Sec. 4.1.1).

The establishment of 4D PC-MRI in the clinics is the goal of current research. Still, before this goal can be reached, more fundamental research has to be done. For example, in order to find irregular flow behavior in patient data, it has to be clear what regular behavior is. Further, it has to be clear which flow structures indicate specific pathologies. Hence, at this point we mainly address medical researchers who want to understand the role of cardiac hemodynamics in the development of cardiovascular diseases. Their findings from clinical studies – with the help of appropriate analysis methods – are expected to have a strong impact on the clinical use of 4D PC-MRI.

In summary, these are our main contributions:

- We introduce a line predicate method for the visual analysis of 4D PC-MRI data of the human aorta. This approach allows to capture the main blood flow structures according to various properties, such as vortices, flow paths, velocity, residence time. With that, it provides a quicker and more comprehensive insight into the main blood flow characteristics than other methods presented in literature so far.
- The method is capable of processing cardiac 4D PC-MRI data. For this, the heart contractions are taken into account during segmentation and additional line predicates are introduced.
- The method allows the user to set queries to identify areas with interesting or abnormal flow characteristics, but it is also suited for the design of automatic overviews.
- We show that vortex detection techniques developed for the analysis of CFD simulation data are applicable to measured flow data.

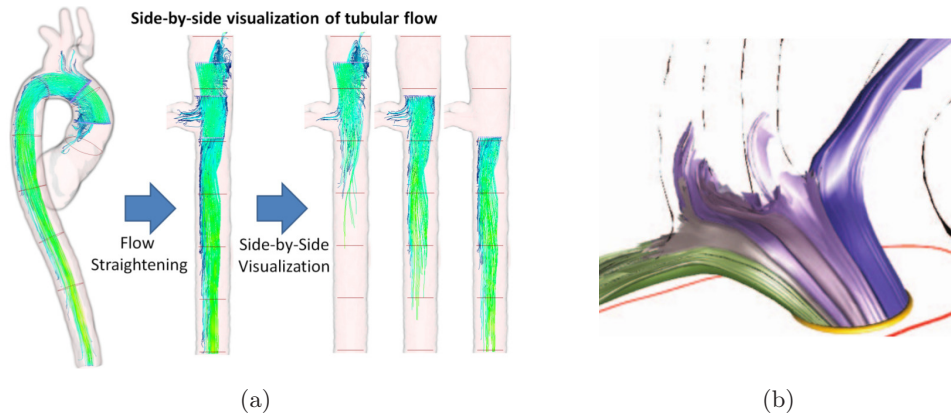


Figure 5.2: Examples for related work of blood flow visualization. (a) Straightened and side-by-side visualization of aortic blood flow by Angelelli et al. [3], © 2011 IEEE, (b) Blood flow segmented according to target regions by Krishnan et al. [85], © 2011 IEEE.

- We evaluate our method by presenting feedback of three 4D PC-MRI experts and showing aspects of flow through healthy and pathological human hearts and aortae that so far can only be displayed with a lot of effort – if at all.

The remainder of this chapter is organized as follows. At first, relevant related work about blood flow visualization as well as the basic concepts with regard to line and point predicates are introduced. Then, the idea of our method, necessary preprocessing steps, and the defined predicates are presented. Finally, we present analysis results of healthy and pathological cases of the aorta and the heart and explain feedback given by experts. We end the chapter with a discussion and suggestions for future work.

Although the work on aortic and cardiac blood flow analysis has been done and published successively, the core concepts are related. In order to avoid redundancy, we combined the presentation of this work in one chapter and point out the significant differences. Altogether, the chapter is mainly based on [15, 16].

5.1 Related Work

4D PC-MRI is a rather young imaging modality that has not found its way into the clinical routine yet. Instead, mainly 4D PC-MRI experts have been engaged with the data analysis for research purposes in the last years. Here, analysis methods have been used that were adopted from standard flow visualization techniques, such as 3D stream lines or path lines [13, 20] or the depiction of flow information on 2D cross-sections by means of color-coding, vector plots, or velocity profiles [108]. These methods were enhanced [9, 57, 110] but still demand a lot of interaction and user experience to find specific flow patterns in the data.

In recent years, the flow visualization community discovered 4D PC-MRI blood

flow as an exciting research area. There is interest in supporting users by improving the existing analysis methods in terms of usability and comprehensibility. For example, van Pelt et al. apply an easier interaction for the selection of 2D vessel cross-sections and use arrows instead of simple color-coding to depict time-dependent blood flow on these planes [181]. An approach to encode a multitude of parameters on the 2D plane that separates an aneurysm from the main vessel is presented by Neugebauer et al. [119]. Furthermore, they introduce interaction widgets that are tailored to the task of examining cerebral aneurysms. Gasteiger et al. tackle the problem of occlusion and clutter when vessel morphology and complex flow are visualized together. They introduce a ghosted view approach that displays the vessel surface but reveals the blood flow depending on the orientation between surface and viewer [47]. Also, to overcome clutter and achieve a more sparse flow representation, van Pelt et al. applied a hierarchical clustering method to unsteady 4D blood flow data [182]. A probing approach developed recently also by van Pelt et al. allows to explore flow by interactively injecting seeds into the 4D PC-MRI flow field and observing the flow behavior based on illustrative renderings and animations [179]. Angelelli and Hauser [3] developed a method to straighten tubular boundaries to provide a side-by-side visualization helping in comparing tubular flows (see Fig. 5.2a).

Thus, there exist several promising approaches to improve the depiction of 3D integral lines or flow parameters on 2D planes. A remaining problem, however, is the interpretation of flow patterns and the overall flow behavior that is still mainly left to the user. To assist the user in this task, Heiberg et al. introduced an automatic detection of swirling flow based on a vector pattern matching technique and showed its applicability to 4D PC-MRI data [56]. Krishnan et al. depict particles with similar paths by segmenting integral lines starting from a seeding plane according to their anatomical target area. The clusters are displayed on the emitter planes [85] (see Fig. 5.2b). Eriksson et al. are concerned with the quantification of blood flow in the left ventricle [36]. They group path lines based on their start and target regions into several groups (entering, exiting, passing, staying in the ventricle) and determine the volume of the different compartments from this.

Our approach is related to these methods in the sense that we also aim at detecting specific flow behavior. But while each of the aforementioned methods deals with one type of flow behavior, we apply a more flexible approach, which allows to structure the flow according to several different properties. To achieve this, we adopt the line predicate approach introduced by Salzbrunn et al. [147, 148]. The path line attributes by Shi et al. are also closely related to our line predicate approach [158]. They apply linked views to select path lines with specific parameters in non-medical data. However, our concept of line selection based on features of the underlying velocity or anatomical data is not readily possible with their technique. In the context of path line attributes, Pobitzer et al. recently presented an analysis about the minimal attribute set necessary to adequately represent features for an interactive flow analysis [130]. Interestingly, the identified attributes correlate with the characteristics that we base our line predicates on.

The semantic layers by Rautek et al. [134] are an example for the use of fuzzy-logic to control the illustrative visualization of complex data. Here, data subsets with similar properties are rendered in a similar way. This relates to our approach in the sense that we also control the visualization by selecting flow areas with specific properties and displaying them or not. Thus, we use a binary approach. It is thinkable to also use fuzzy-logic for our application to, e.g., show integral lines fulfilling a property with a certain probability. However, this leads again to a higher number of rendered lines and cluttered visualizations that need to be designed in a way that they are still comprehensible. As we feel that the medical researchers are more interested in a simple visualization, we aim at an intuitive, uncluttered visualization and prefer the binary predicate evaluation.

5.2 Concepts

A 4D PC-MRI heart dataset consists of several measurements at different time points altogether covering an averaged cardiac cycle (see Sec. 4.2). The complete dataset corresponds to an unsteady velocity vector field $v : D \times T \rightarrow \mathbb{R}^3$ with the field domain $D \subset \mathbb{R}^3$ (anatomical area covered by the scan) and timespan $T = [t_0, t_n]$ (cardiac cycle covered by $n + 1$ time steps).

A formal introduction to path lines and stream lines is given in Section 2.1.1. In the more specific context of 4D PC-MRI data, a path line $P_{a,\tau}$ represents the movement of a virtual blood particle from a starting point $a \in D$ and starting time $\tau \in T$ through the unsteady blood flow in the pumping heart. Stream lines S_{a,t_i} however are calculated only for a single timestep t_i with starting point $a \in D$. The course of a stream line is not equivalent to the course of a blood particle through the pumping heart. Instead, it gives information about the steady flow situation at that timestep t_i of the cardiac cycle.

The basis of our visual analysis approach is the idea that the infinite set of path lines or stream lines contains all necessary information about the flow dynamics of the given 4D PC-MRI dataset. So, we precalculate a sufficient number of integral lines and use line predicates to select particular lines with specific flow characteristics, i.e., we extract meaningful flow structures. In the following, we briefly recapitulate the concept of line predicates [147, 148].

5.2.1 Point and Line Predicates

In mathematical logic, predicates are Boolean functions. A *point predicate* Γ is Boolean function, which evaluates to true or false depending on whether a certain feature is valid at a point $(x, t) \in D \times T$:

$$\begin{aligned} \Gamma : D \times T &\rightarrow \{\text{true}, \text{false}\} \\ (x, t) &\mapsto \Gamma(x, t) \end{aligned} \tag{5.1}$$

Similarly, a *line predicate* Λ evaluates to true or false depending on whether a considered integral line $P_{a,\tau} \in \mathcal{P}$ meets a certain characteristic:

$$\begin{aligned} \Lambda : \mathcal{P} &\rightarrow \{\text{true}, \text{false}\}, \\ P_{a,\tau} &\mapsto \Lambda(P_{a,\tau}) \end{aligned} \tag{5.2}$$

with \mathcal{P} describing the set of all integral line equivalence classes (see Sec. 2.1.1). The characteristic set \mathcal{C}_Λ of a line predicate Λ defines all points in spacetime visited by lines fulfilling this predicate:

$$\mathcal{C}_\Lambda = \{(x, t) \in D \times T \mid \Lambda(P_{x,t}) = \text{true}\}$$

with $t = t_i$ for the characteristic sets of stream line predicates.

The evaluation of one line predicate yields two sets of integral lines that do or do not fulfill the predicate respectively. This division is the simplest example of a *flow structure* that is defined as the partitioning of the flow domain into disjoint groups:

$$\mathcal{P} = \bigcup_{k \in \mathcal{K}} \mathcal{C}_{\Lambda_k} \tag{5.3}$$

with an index set \mathcal{K} . For blood flow visualization, it is useful to combine several line predicates resulting in a more distinct line selection and more meaningful flow structures. Here, however, it must be assured that every line is assigned to one and only one characteristic set. In other words, the chosen set of line predicates \mathcal{G} needs to yield disjoint characteristic sets. Predicate sets do not yield disjoint line groups automatically. According to Salzbrunn et al. [146], this requirement can be met with the following definition of a predicate set (with predicates Λ_x and Λ_y):

$$\mathcal{G} = \{\Lambda_x \wedge \Lambda_y, \neg \Lambda_x \wedge \Lambda_y, \Lambda_x \wedge \neg \Lambda_y, \neg \Lambda_x \wedge \neg \Lambda_y\} \tag{5.4}$$

In our approach, however, we usually carve out single line bundles by concatenating several line predicates with a logical AND-operation. This means, that we only consider one element of \mathcal{G} and the described problem does not arise.

5.2.2 Line-based and Derived Predicates

Line-based and derived line predicates differ in the data that is used for their evaluation. For *line-based* predicates the examined feature depends only on characteristics of the integral line itself. *Derived* predicates deal with features of the corresponding velocity field or anatomical volume dataset. The detection of a derived feature is equivalent to the evaluation of a point predicate Γ . The result of the point predicate is then used as input for the derived line predicate. The line predicate resolves to *true*, if a path line $P_{a,\tau}$ is present at the point $(x, t) \in D \times T$ fulfilling the point predicate $\Gamma(x, t)$:

$$\Lambda(P_{a,\tau}) = \begin{cases} \text{true} & \text{if } \exists t \in T_{a,\tau}, P_{a,\tau}(t) = x \\ & \text{and } \Gamma(x, t) = \text{true}, \\ \text{false} & \text{otherwise.} \end{cases} \tag{5.5}$$

In the steady case, for stream line S_{a,t_i} :

$$\Lambda(S_{a,t_i}) = \begin{cases} true & \text{if } \exists s \in B_a, S_{a,t_i}(s) = x \\ & \text{and } \Gamma(x, t_i) = true, \\ false & \text{otherwise.} \end{cases} \quad (5.6)$$

5.3 Analysis of Aortic and Cardiac Blood Flow

We present a method using line predicates for the analysis of 4D PC-MRI blood flow data of the human aorta and heart. The basic idea of our line predicate approach is that the full set of integral lines contains all necessary information about a dataset's blood flow dynamics. Visualizing flow with stream lines and path lines is very intuitive, but displaying too many lines at a time is confusing and hinders insight. In our approach, we therefore allow the user to interactively select and display lines of interest (e.g., with a certain velocity, running through a certain anatomical area or the like). For this selection process, we provide the user with several line predicates. We formulated these predicates based on the input of 4D PC-MRI experts. Together with an automatic vortex detection, these predicates form a tool box helping users to explore the 4D PC-MRI data and easily display distinct flow structures of interest. With this approach, we give an alternative to current analysis techniques, which also use integral lines but leave the detection of interesting patterns and the flow interpretation fully to the users.

In the following, we describe the preprocessing of the 4D PC-MRI datasets, the different line predicates for the analysis of steady and unsteady blood flow, and the visualization of the extracted flow structures. We consider the use of our method for the exploration of blood flow in the aorta and the whole heart. These two anatomical areas place different demands on our line predicate approach. The aortic blood flow is rather well understood and the aortic shape does not change too much over the cardiac cycle. In contrast, the blood flow through the heart is highly complex and the heart deforms strongly due to the pumping behavior. So, in the next sections we always discuss the methodical differences for these two use cases.

5.3.1 Workflow and Data Preprocessing

With 4D PC-MRI, an anatomical dataset and three velocity datasets are acquired simultaneously (see Fig. 5.3, left). From the three velocity datasets - each representing the flow velocities in one of the three spatial directions - a velocity vector field is derived. This vector field forms the basis for all advanced analysis methods. Section 4.2 contains a more thorough introduction to 4D PC-MRI data.

Our line predicate approach is divided into two main phases. As depicted in Figure 5.3, a data preprocessing phase is required to segment the anatomical area of interest, calculate the integral lines, and detect vortices. Then, the visual analysis can take place where the users can freely combine line predicates to explore the flow data in this anatomical area. In the following, we explain the needed preprocessing steps.

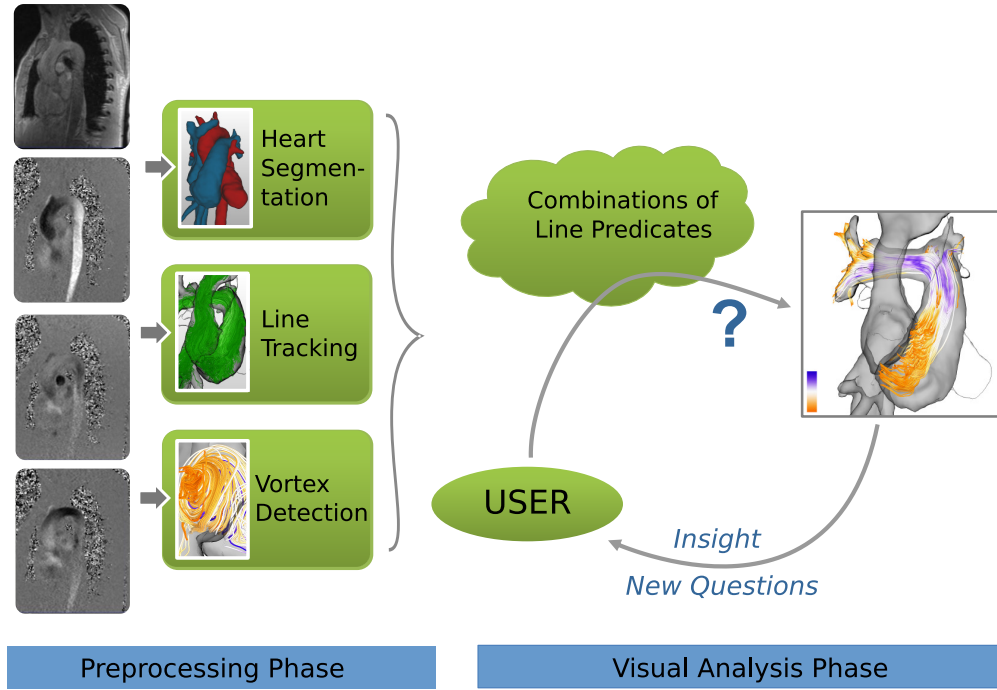


Figure 5.3: Workflow: In the data preprocessing step, the anatomical (left, top image) and the velocity datasets (left, lower three images) are used to segment the desired anatomical region, precalculate integral lines, and detect vortices. In the visual analysis phase, the user is provided with a line predicate tool box for an easy exploration of the 4D PC-MRI dataset (based on the precalculated information). © 2013 IEEE.

Segmentation

As a first step, the anatomical area of interest needs to be segmented. Anatomical regions we consider here are the aorta and the heart. With respect to the heart, this could be the left or the right heart or both, each consisting of atrium, ventricle, and adjacent vessels. The segmentation results are used later on as seed masks for the line tracking, as predefined anatomical regions of interest for several derived predicates, and as context for the line visualization.

For a correct segmentation of the anatomy, each timestep needs to be handled separately. Only then we can ensure that at every point in time the integral lines cover the complete area of interest and the predefined regions of interest (ROIs) correlate with the desired anatomical area. Since the anatomical datasets are usually not contrast-enhanced with respect to blood-carrying structures, the segmentation is done mainly on the velocity data. First, we calculate the average velocity value for each voxel over all timesteps of the 4D PC-MRI data and store these values in a scalar data volume. Blood-carrying structures, such as heart chambers and vessels, have higher average velocities than areas without blood flow. Thus, in the derived volume dataset these structures appear brighter than areas where no blood flow

takes place. We take advantage of this and use region-growing to create an initial segmentation mask. This segmentation represents the maximally expanded heart chambers or the aorta.

The aorta does not change its shape too much over the cardiac cycle. If we decide to accept a slight oversegmentation during some timesteps, we can already use this region-growing result as aorta segmentation. If necessary, the aorta is separated manually from neighboring vessels.

The heart segmentation is more demanding than the aorta segmentation due to the heart's strong shape changes over time in combination with the thin tissue layers between the different heart chambers. So, here we continue with determining the minimal size of ventricles and atria. The minimal size of the ventricles relates to the late systole when blood is pumped from the ventricles into the pulmonary artery and aorta respectively. The atria are smallest during late diastole when blood has been ejected from the atria into the ventricles. These timesteps in the 4D PC-MRI measurement are characterized by high velocity values. We determine the absolute velocity values for each timestep and segment areas with high velocities inside the initial mask again with the region-growing approach. Based on the minimal and maximal heart masks, the other timesteps are interpolated manually based on additional anatomical information from the morphology scans.

Segmentation was not the focus of this work. The chosen approach requires a high manual effort and is only suitable to show the feasibility of our line predicate method. For the processing of larger amounts of data in, e.g., research studies or the clinical routine, an advanced segmentation technique is necessary. It is thinkable to utilize a level set method as it has already been presented for the left heart [77] or a deformable model approach as it was used for the large vessels [180].

Integral Line Precalculation

The previously generated segmentations of the aorta or the heart are used as seeding masks for line integration. Both, path lines and stream lines, are generated with a DOPRI5 integration [33].

We require that every temporal flow pattern larger than twice the voxel size is captured by the path lines. This implies that at least one particle is present in each voxel at each timestep. The straight-forward approach of seeding a particle in each voxel at each timestep results in a large number of lines containing a lot of redundant flow information. To speed up the method, we reduce the number of seeded lines (based on the method in [148]). For the first timestep, particles are seeded in each voxel of this timestep's seeding mask and integrated in forward and backward direction. For the second timestep, particles are emitted only in voxels of the second timestep's seeding mask that are unvisited by previous particles at that point in time. This procedure is repeated for all timesteps until all voxels are visited in each timestep. Compared to the trivial approach, this reduces the number of lines by 81-95%.

Stream line calculation is carried out analogously providing that at least one

stream line runs through each voxel of the corresponding segmentation mask. We start a stream line in one voxel, integrate back and forth to cover the complete field domain of this timestep, and keep track of the voxels visited by this first stream line. Subsequently, stream lines are only started from voxels not yet visited by a stream line.

Both, stream lines and path lines, are then stored as line strips, together with the current velocity and the line parameter value for each vertex x_i . Further, for the evaluation of line-based predicates (see Sec. 5.3.2) several other parameters are calculated and stored with each integral line. More specifically, for each path line $P_{a,\tau}$ (with velocity $v_i(P_{a,\tau})$ at vertex $x_i \in \{0, \dots, n\}$) the following parameters are computed:

- Line length: $len(P_{a,\tau}) = \sum_{i=1}^n |x_i(P_{a,\tau}) - x_{i-1}(P_{a,\tau})|$
- Maximal velocity: $v_{max}(P_{a,\tau}) = \max(v_0(P_{a,\tau}), \dots, v_{n-1}(P_{a,\tau}))$
- Mean velocity: $\bar{v}(P_{a,\tau}) = \frac{len(P_{a,\tau})}{t_{max}(P_{a,\tau}) - t_{min}(P_{a,\tau})}$

These parameters are analogously defined for stream lines S_{a,t_i} .

Derived predicates (see Sec. 5.2.2) are evaluated by determining all lines running through a specific set of voxels. The cost for this test when implemented straightforwardly, i.e., by testing every line vertex whether it lies within a volume of interest, depends on the number of lines, line vertices, and voxels. We reduce the costs for this test significantly with a lookup table that assigns to every voxel in the field domain D all integral lines crossing them and, for path lines, at what point in time. Since a quick predicate evaluation is essential for an interactively usable visual analysis tool, we accept the resulting memory overhead.

Vortex Detection

Since our approach is meant to help identify important flow structures, we provide an automatic vortex detection. As more thoroughly described in Section 2.2.4, the vortex core line detection by Sujudi and Haimes [167] and the λ_2 vortex region detection by Jeong et al. [68] belong to the standard methods for vortex detection in CFD simulation data. Thus, we chose these two methods to research whether techniques established for CFD data might also be appropriate for measured flow data. What further supported our choice is the fact that here only first-order derivatives are used. Since 4D PC-MRI data is rather noisy and has – when compared to CFD – a low spatial resolution, we expected insufficient results from higher-order techniques (such as [142]).

The selected vortex detection methods can be restated to point predicates that determine for each voxel at a specific point in time whether a vortex or a vortex core line respectively is present at this position or not [148]. The point predicate

for the λ_2 method [68] is expressed by:

$$\Gamma_{\lambda_2}(x, t) = \begin{cases} \text{true} & \text{if } S^2 + \Omega^2 \text{ at position } x \\ & \text{and time } t \text{ has 2} \\ & \text{negative eigenvalues,} \\ \text{false} & \text{otherwise.} \end{cases} \quad (5.7)$$

$S = \frac{1}{2}(J + J^T)$ and $\Omega = \frac{1}{2}(J - J^T)$ are the symmetric and asymmetric parts of the vector field's Jacobian matrix J .

The vortex core line detection by Sujudi and Haimes [167] formulated with the parallel vectors operator by Peikert et al. [126] results in the following point predicate:

$$\Gamma_{SH}(x, t) = \begin{cases} \text{true} & \text{if at time } t, J_x \text{ has complex} \\ & \text{eigenvalues and} \\ & \mathbf{v}(x, t) \parallel (J_x \mathbf{v}(x, t)), \\ \text{false} & \text{otherwise.} \end{cases} \quad (5.8)$$

J_x is the Jacobian matrix entry for position x at time t . Since 4D PC-MRI data contains more noise than simulation data, it is most likely that a vortex detection method produces false positives. Therefore, we combine the previously described techniques: On the one hand, we detect vortex core lines and discard lines shorter than $30mm$. On the other hand, we apply the λ_2 -method. Finally, we define a vortex as detected if at a voxel a core line is present and the λ_2 point predicate returns true (see Fig. 5.4a,b).

With this, the vortex core line is detected. The vortex region can be determined based on the λ_2 -result. In practice, however, this depends on a well-chosen threshold, as 0 does not always give usable results, especially with measured data. 4D PC-MRI data has as a rather low spatial resolution. Thus, we use the original voxel representation of the vortex core line as approximation of the vortex region. A comparison of this approach with the use of the λ_2 -result shows that the results of the vortex predicates do not differ significantly (see Fig. 5.4c,d). However, further research is needed on this topic as for very small or very large vortices the regions might be overestimated or underestimated respectively. Also, we achieved robust results by discarding all vortex core lines shorter than $30mm$. A higher threshold increases the risk of missing important vortices and a lower threshold increases the risk of false positives. Still, it should be investigated whether this threshold could be more flexibly adapted to the current dataset. For the actual evaluation of vortex predicates later on, a ROI point predicate Γ_{ROI} is used to test whether a point (x, t) in spacetime is part of a vortex region I_{vor} .

5.3.2 Line Predicates for Analysis of Steady Blood Flow

The following line predicates allow the exploration of steady blood flow, i.e., they can be applied to stream lines of a single measured timestep of the 4D PC-MRI data. Thus, with these line predicates one cannot yet draw conclusions about the spatiotemporal blood flow behavior, but about the flow situation at a specific time

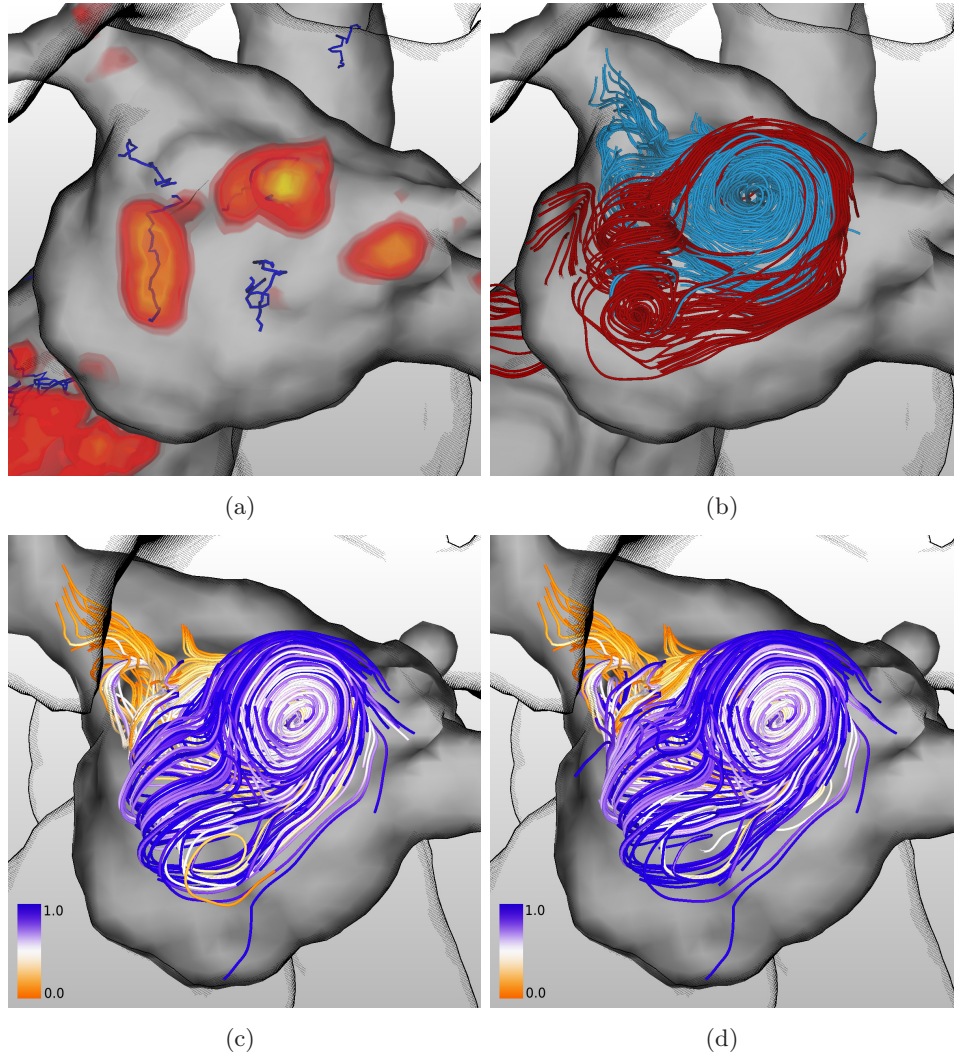


Figure 5.4: Vortex detection in the left atrium of a healthy heart during diastole ($t = 672ms$). (a,b) A vortex is detected if both the λ_2 and the Sujudi-Haimes method yield a positive result. (a) Detected vortex core lines and the λ_2 result. The volume rendering depicts λ_2 values from $-8.0 \cdot 10^{-5}$ (red) to $-2.5 \cdot 10^{-3}$ (yellow). (b) Stream lines running through two different vortex regions that were previously detected both by λ_2 and Sujudi-Haimes. (c,d) The results of a vortex predicate evaluated based on λ_2 (c) and based on a voxelized vortex core line (d) are almost identical. Color-coding represents the line parameter. All images © 2013 IEEE.

point during the cardiac cycle. The predicates were defined based on the input of 4D PC-MRI experts about what blood flow aspects they are interested in.

Region-of-Interest (ROI) Predicate

An apparent interest about the cardiac blood flow is the course the blood takes through the different chambers. With the region-of-interest line predicate Λ_{ROI} stream lines can be selected based on anatomical regions they run through. These anatomical regions change in shape and position over the cardiac cycle due to the heart contractions. Thus, a region of interest I is a subset of $D \times T$, i.e., it is part of the field domain D and can vary over time.

The ROI point predicate Γ_{ROI} determines whether a point x is part of a region of interest I at time t :

$$\Gamma_{ROI}(I, x, t) = \begin{cases} true & \text{if } (x, t) \in I, \\ false & \text{otherwise.} \end{cases} \quad (5.9)$$

The user can either interactively define a ROI or select one from the predefined regions, such as the different chambers, the areas of the valves and the adjacent vessels. The ROI line predicate Λ_{ROI} evaluates to true, if the course of a respective stream line intersects at least one voxel of this specified area I :

$$\Lambda_{ROI}(S_{a,t_i}, I) = \begin{cases} true & \text{if } \exists s \in B_a, \\ & S_{a,t_i}(s) = x \text{ and} \\ & \Gamma_{ROI}(I, x, t_i) = true, \\ false & \text{otherwise.} \end{cases} \quad (5.10)$$

By combining several ROI predicates one can select lines even more specifically and display, e.g., stream lines entering the left atrium but not continuing into the left ventricle. The latter is possible because one can also use negative predicates and select lines *not* running through a specific area. A special version of the ROI predicate determines whether the beginning of the line is inside the ROI. This helps to select lines *starting* in a specific anatomical area.

Vortex Predicate

Vortices are of special interest for blood flow analysis. Vortices emerging, e.g., in the healthy human aorta or the left ventricle are rather well-understood [42, 80]. The occurrence of abnormal vortices may therefore be an indication for a pathology and significant for disease prognosis.

With the vortex predicate, stream lines running through a specific vortex can be determined. This way, not only the vortex region itself, but the flow area influenced by the vortex, can be identified. A prerequisite for this predicate is the detection of vortices in the underlying vector field (see Sec. 5.3.1). Based on this, the vortex predicate Λ_{vor} evaluates to true, if the stream line traverses a voxel of the respective vortex area I_{vor} . Thus, Λ_{vor} is defined analogously to Equation 5.10. An example for stream lines selected with the vortex predicate is shown in Figure 5.4d.

Maximal Velocity Predicate

This predicate identifies stream lines with a maximal velocity v_{max} in a specified interval $[v_1, v_2]$. With $[v_1, \infty]$, e.g., this allows the medical expert to identify areas with high velocities that usually correspond to the main flow paths. As an example, this predicate allows to easily capture the flow during systole when blood is ejected from the ventricles. Also, these flow paths can form a significant criterion during diagnostics as high-velocity jets may also indicate pathologies like valvular stenosis and might correlate with a higher risk for aneurysm development (see Sec. 4.1.1).

The maximal velocity predicate $\Lambda_{v_{max}}$ is line-based, i.e., it is evaluated according to the precalculated parameter v_{max} (see Sec. 5.3.1) for each stream line S_{a,t_i} :

$$\Lambda_{v_{max}}(S_{a,t_i}, v_1, v_2) = \begin{cases} true & \text{if } v_{max}(S_{a,t_i}) \in [v_1, v_2], \\ false & \text{otherwise.} \end{cases} \quad (5.11)$$

Mean Velocity Predicate

Along the lines of the previous predicate, this function identifies integral lines with a mean velocity in a specific interval $[v_1, v_2]$. As a result, areas with a high average velocity can be distinguished from areas with low average speed. The respective predicate $\Lambda_{\bar{v}}$ is defined analog to Equation 5.13 and evaluated based on the precalculated mean velocity \bar{v} of each stream line (see Sec. 5.3.1).

Regional Velocity Predicate

Due to noise or partial volume effects, it occurs that integral lines cross anatomical borders, such as closed heart valves. As a result, flow areas are wrongly connected and cannot be analyzed separately. One arising problem is that the selection of integral lines with a specific velocity in a particular region gets difficult. Applying a maximal velocity predicate might select unwanted lines because, e.g., the blood flow in the neighboring heart chamber is faster and masks the desired lines. A combination with a ROI line predicate (removing the lines in the neighboring chamber) also eliminates lines of interest. To solve this problem, we define the regional velocity predicate that evaluates to true if a line reaches a velocity $v \in [v_1, v_2]$ in a defined anatomical region I :

$$\Lambda_{v_r}(S_{a,t_i}, I, v_1, v_2) = \begin{cases} true & \text{if } \exists \sigma \in B_a, \\ & \frac{dS_{a,t_i}}{ds}(\sigma) \in [v_1, v_2], \\ & S_{a,t_i}(\sigma) = x \text{ and} \\ & \Gamma_{ROI}(I, x, t_i) = true, \\ false & \text{otherwise.} \end{cases} \quad (5.12)$$

This line predicate was defined and is especially useful for cardiac 4D PC-MRI data. Here, the problem of integral lines not obeying anatomical borders arises more often. Reason for this are the heart valves, which are prone to partial volume effects.

Length Predicate

The length predicate Λ_{len} allows to structure the flow according to stream line length and allows to, e.g., filter out shorter lines that oftentimes lead to clutter. It identifies lines with a length in the interval $[l_1, l_2]$ and is defined analogously to Equation 5.13:

$$\Lambda_{len}(S_{a,t_i}, l_1, l_2) = \begin{cases} true & \text{if } len(S_{a,t_i}) \in [l_1, l_2], \\ false & \text{otherwise.} \end{cases} \quad (5.13)$$

5.3.3 Line Predicates for Analysis of Unsteady Blood Flow

While the steady analysis of a single timestep of the 4D PC-MRI data allows to explore flow patterns at a fixed point in time, the unsteady analysis allows to capture features of the flowing blood over time. As the steady flow is a special case of the unsteady flow, the majority of the previously explained predicates can be applied here as well: The line-based predicates concerned with mean velocity, maximal velocity, and line length can be adopted without change.

With respect to the derived line predicates, the regional velocity predicate and the ROI line predicate as explained in Section 5.3.2 can also be used in the unsteady case with a small change (see unsteady variants below). Additional time-dependent variants are introduced to select path lines running through an area at a *specific* point in time.

The vortex predicate cannot be used as is. A necessary time-dependent variant is explained shortly. Finally, a residence time predicate is introduced that allows to display path lines residing in a ROI for longer than a user-defined time interval.

Time-dependent ROI predicate

When inspecting the course of path lines, one might be interested in path lines that flow through a specific area at a certain time t_i in order to, e.g., examine their subsequent course by applying further predicates. As in the steady case, the region of interest is either selected from the predefined ROIs or defined manually by the user. Then, the ROI predicate Γ_{tROI} determines all path lines flowing through this area at the given time t_i .

$$\Lambda_{tROI}(P_{a,\tau}, I, t_i) = \begin{cases} true & \text{if } P_{a,\tau}(t_i) = x \text{ and} \\ & \Gamma_{ROI}(I, x, t_i) = true, \\ false & \text{otherwise.} \end{cases} \quad (5.14)$$

Unsteady Region-of-Interest (ROI) Predicate

In the unsteady case, one might also want to select all path lines running through a specific region of interest - not at a specific time (as described previously) but at any time during the cardiac cycle. Here, it is important to handle the deformation and shifting of regions while the heart pumps. Therefore, an adapted ROI needs to be used for each timestep. This relates to the use of a time-dependent point predicate for the ROI definition:

$$\Gamma_{ROI}(P_{a,\tau}, I) = \begin{cases} true & \text{if } \exists t \in T_{a,\tau}, \\ & P_{a,\tau}(t) = x \text{ and} \\ & \Lambda_{ROI}(I, x, t) = true \\ false & \text{otherwise.} \end{cases} \quad (5.15)$$

For user-defined ROIs, the user has to ensure that the correct area of interest is specified for each timestep. With predefined ROIs, the user can simply select the desired anatomical region and avoid this overhead.

Unsteady Regional Velocity Predicate

Analog to the unsteady ROI predicate, the regional velocity predicate needs to be adapted to the deforming morphology. With this unsteady regional velocity predicate, lines are selected that have a specific velocity in a region of interest that might change during the cardiac cycle. Therefore, the point predicate defining the ROI is now time-dependent:

$$\Gamma_{vreg}(P_{a,\tau}, I, v_1, v_2, t_i) = \begin{cases} true & \text{if } \exists t \in T_{a,\tau}, \\ & \frac{dP_{a,\tau}}{dt}(t) \in [v_1, v_2], \\ & P_{a,\tau}(t) = x \text{ and} \\ & \Lambda_{ROI}(I, x, t_i) = true \\ false & \text{otherwise.} \end{cases} \quad (5.16)$$

Time-dependent Vortex Predicate

In an unsteady flow field, the location and shape of vortices change over time. Therefore, the vortices are detected for each timestep and the path lines are queried for whether they flow through the respective area at a specific time t_i :

$$\Lambda_{tVor}(P_{a,\tau}, I_{vor}, t_i) = \begin{cases} true & \text{if } P_{a,\tau}(t) = x \text{ and} \\ & \Gamma_{vor}(I_{vor}, x, t_i) = true, \\ false & \text{otherwise.} \end{cases} \quad (5.17)$$

Inspecting the influence of a vortex on a path line over a longer time is accomplished by combining vortex predicates for subsequent timesteps. Tracking a vortex over several timesteps can be accomplished spatially (see Sec. 2.2.4). This approach is feasible, as the main vortices occurring in the aorta or the heart chambers are rather large compared to the structure size and do not dislocate significantly over time. Further, in case vortices are wrongly identified as being the same vortex in different timesteps, the time-dependent vortex predicate will not show meaningful results with this input. Therefore, the predicate evaluation can be seen as confirmation of a correct vortex correspondence.

Residence Time Predicate

This path line predicate determines whether a particle resides longer than a specified time span $t_r > 0$ in a predefined ROI. The residence time of particles in the heart can be used as a measure for the efficiency of the working heart. But long residence times may also indicate a higher risk for thrombogenesis and, thus, of suffering from stroke or a heart attack. Blood clots often form in (aortic) aneurysms or in the left atrium in conjunction with atrial fibrillation (see Sec. 4.1.1). So, the residence time predicate is of special interest in these structures. The residence time predicate Λ_{res} can be expressed with:

$$\Lambda_{res}(P_{a,\tau}, I, t_r) = \begin{cases} true & \text{if } \exists t_1, \exists t_2, (t_2 - t_1) \geq t_r, \\ & \forall t \in [t_1, t_2], \\ & \Lambda_{tROI}(P_{a,\tau}, I, t) = true, \\ false & \text{otherwise.} \end{cases} \quad (5.18)$$

5.3.4 Flow Structure Visualization

The lines selected by the different predicates are visualized in the context of the corresponding aorta or heart mesh and the anatomical MRI slice data (optionally).

With the aortic data, the flow information is displayed inside the aortic mesh, which was generated based on the segmentation mask. In order to provide sufficient context for the flow bundles without causing occlusion, the back-facing part of the aortic mesh is rendered opaquely, whereas the front-facing part is depicted as a simple line rendering together with an illustrative shading technique (halftoning, see Sec. 3.3.2).

With the rendering of the heart chamber models, we differentiate between focus and context rendering. The focus rendering is done similarly to the previously described aortic meshes, i.e., the front layer is rendered illustratively and the chamber background as gray isosurfaces. The chambers that are currently examined are rendered this way. The context rendering also consists of halftoned silhouette renderings but without the opaque background. The context rendering is used for the remaining chambers because it is an unobtrusive way of giving some anatomical orientation. Usually, one investigates the blood flow either through the left or the right heart. This means that oftentimes the left heart is rendered as focus and the right heart as context or vice versa.

The integral lines are rendered as color-coded tubes. The color-coding either represents the complete line's mean velocity or the changes of parameters along the course of the line. In the latter case, the colors represent the velocity, line parameter, or time at the individual line vertices. The line parameter encoding gives information about the flow direction in case this may not be clear from the context. Despite of the perceptual disadvantages of the rainbow color map, it is still the common way of depicting line parameters in medical software and publications. Not following this convention, but the advice of Silva et al. [159], we decided to use diverging maps. Due to feedback of users who suggested using different colormaps

for different parameters, we apply colormap ranging from orange to purple for the line parameter and a colormap ranging from green to pink for the velocity.

To show the course of lines over time, it is also possible not to display the complete line but only segments according to the line parameters. This allows to display, e.g., the temporal evolution of path lines, as shown for a vortex example in Figure 5.10.

It is possible to display a line bundle not as single lines, but as a mesh representing the outer border of the bundle. Here, a smooth surface of the voxelized characteristic set is displayed transparently and thus allows to overcome occlusion (see Fig. 5.5a). This visualization style is appropriate if only the outer shape of the line bundle is of interest or if a less dense representation is desired when several bundles are visualized.

We implemented the line predicate approach as a scene graph addition to our medical visualization framework. The predicates are represented by separate scene graph nodes that can be combined freely during run-time. Except for the user-defined predicate parameters, no parameter adjustment is necessary.

5.4 Results and Evaluation

We applied our line predicate method to a number of 4D PC-MRI measurements of the aorta and the heart (both pathological and healthy cases). The datasets were preprocessed as introduced in [12, 109]. After that, segmentation of aorta or heart respectively, precalculation of integral lines, as well as vortex detection was accomplished as described previously in Section 5.3.

Here, we discuss aorta and heart cases separately. So, in the following sections, we first introduce the datasets of healthy and pathological aortae and discuss data-specific insights gained from their analysis. Then, we do the same for a healthy and a pathological heart case. Finally, we evaluate our method with regard to performance numbers and a summary of general expert feedback that we received during data analysis.

5.4.1 Aorta Cases

In this section, we present our analysis results with respect to a healthy aorta and two patient cases with an aortic aneurysm and a distorted aorta respectively. We do not provide a full discussion of each dataset, but we show and discuss the usefulness of our method by means of selected flow features of each case.

5.4.1.1 Aorta of Healthy Volunteer

Using the example of the 4D PC-MRI of a healthy volunteer (spatial resolution: $1.67 \times 1.67 \times 2.2 \text{ mm}^3$; temporal resolution: 40.8 ms ; 17 timesteps; velocity encoding: $150 \frac{\text{cm}}{\text{s}}$), we examine typical flow patterns occurring in the aorta during a cardiac cycle. It is known, e.g., that a right-handed vortex forms in the healthy

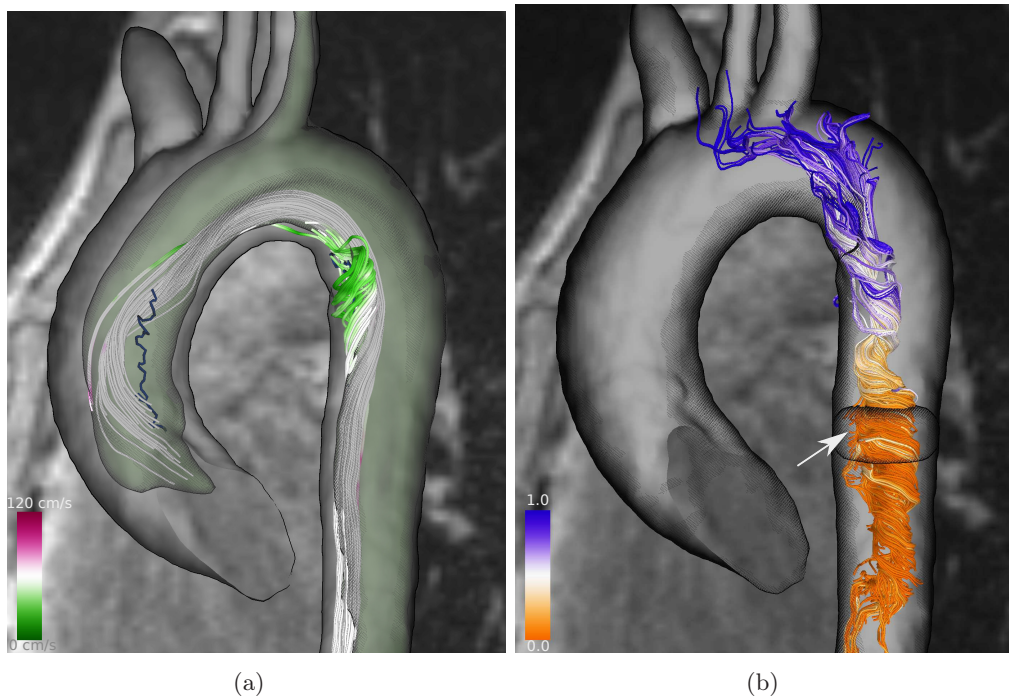


Figure 5.5: Healthy aorta. (a) Green isosurface represents the union of all stream lines running through the vortex in the ascending aorta (blue core line). White lines are a subset running also through the vortex in the descending aorta. The color-coded stream lines through the arch vortex are outside this isosurface and, thus, do not interact with the ascending vortex. (b) Stream lines starting at the ROI (white arrow) and running through a vortex in the descending aorta reveals retrograde vortical flow in late systole.

ascending aorta during systole [13, 59, 80]. This could be verified in the present dataset and two other vortices develop in the descending aorta at $t = 100ms$ (one close to the arch, one further downstream). Applying vortex predicates to stream lines reveals the interplay of these three vortices at a specific timestep. The green isosurface in Figure 5.5a represents the characteristic set of the vortex predicate for the ascending vortex, i.e., all stream lines flowing through this vortex are within this surface. The color-coded stream lines result from the vortex predicate for the arch vortex. These lines are outside the characteristic set of the ascending vortex. This indicates that these vortices do not interact at this timestep. We can confirm this finding by evaluating the combined vortex predicate for the ascending *and* the arch vortex which results in an empty characteristic set. The white stream lines display a subset of stream lines flowing through the ascending and the descending vortex. Exploring the interaction of vortices, as shown here, provides insights into the overall flow behavior, which is not possible with current methods yet.

During end systole (at around $t = 300ms$) rotating and retrograde flow evolves in the healthy aorta, especially in the descending part [13, 59, 80]. Figure 5.5b shows

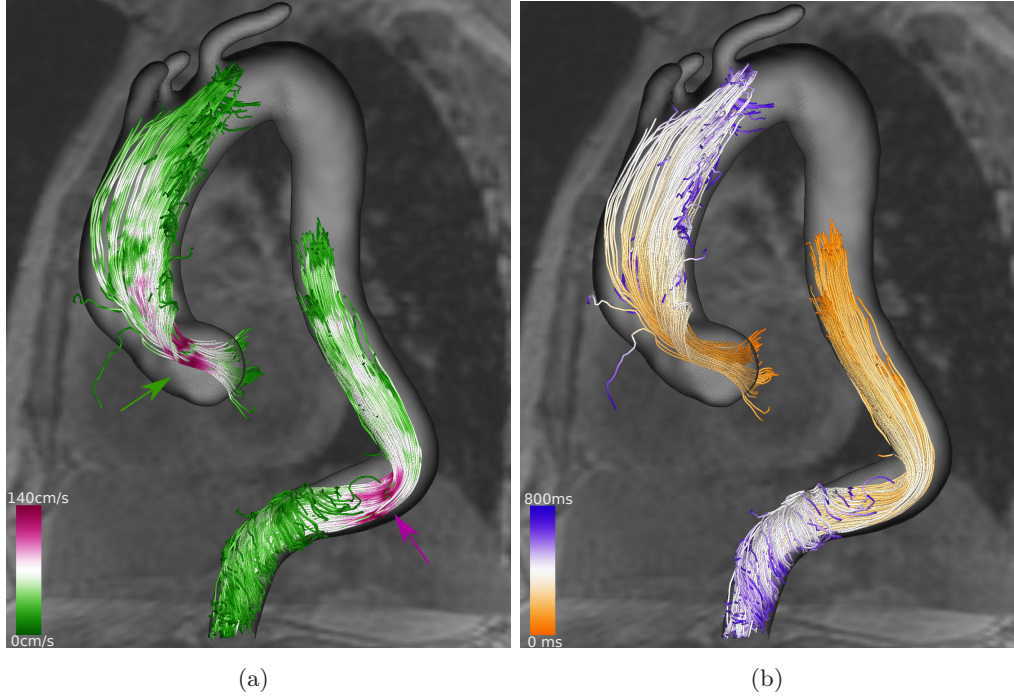


Figure 5.6: Path lines of the distorted aorta dataset representing particles which are at some point faster than $130 \frac{cm}{s}$. Velocity color mapping reveals these areas of fast flow (left). Time color mapping shows at what time during the cardiac cycle the velocity was high (right).

stream lines involved in the descending vortex at $t = 350ms$ that are passing the depicted ROI (white arrow). This reveals the course of reverse flow in the vortex. Again, we have an advantage to current methods. In addition to the possibility to inspect reverse flow by starting integral lines from a 2D cross-section, we can select and visualize the reverse flow in a specific vortex. Other subsets as the reverse flow into an anatomical region of interest or the like are also thinkable and useful.

5.4.1.2 Distorted Shape in Descending Aorta

This 4D PC-MRI (spatial resolution: $1.67 \times 1.67 \times 2.2 mm^3$; temporal resolution: $40.8ms$; 22 timesteps; velocity encoding: $150 \frac{cm}{s}$) of a female patient (72 years) was acquired after an MRI revealed an uncommonly bent shape of the descending aorta. This distortion leads to changes in the blood hemodynamics compared to a healthy aorta [107]. A consequence of the kink is, e.g., an increase in velocity. Flow with unusual high velocity occurring at any time during the cardiac cycle can be easily identified with a maximal velocity predicate applied to the complete path line set. In this dataset, this predicate reveals the expected high-speed flow at the proximal ascending aorta where the blood enters from the left ventricle (green arrow in Fig. 5.6a) and the aforementioned area of high velocity at the kink in the descending aorta (pink arrow in Fig. 5.6a). To find out in what phase of the cardiac

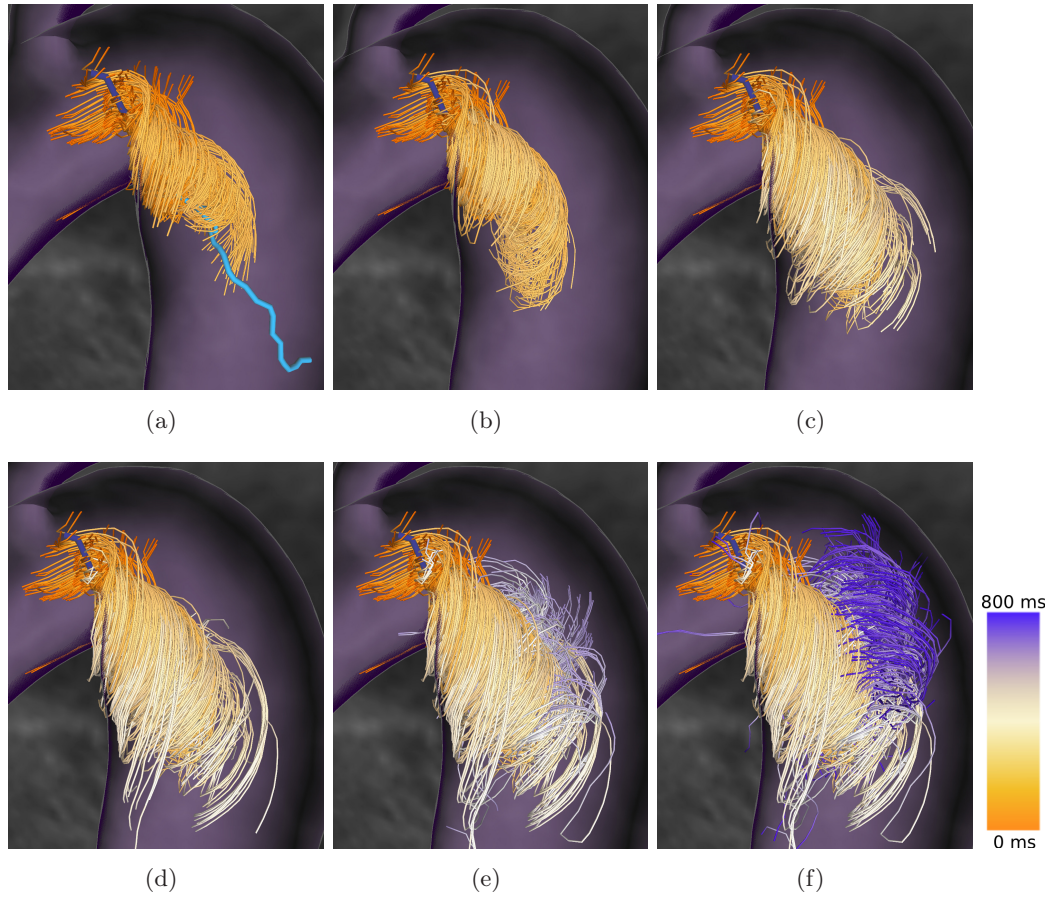


Figure 5.7: Vortex in the proximal descending aorta of the 4D PC-MRI data of the distorted aorta. Path lines run through the evolving vortex at $t = 102ms$. The dark lines represents the vortex core line at $t = 102ms$. In (a) the light blue line depicts the core at $t = 347ms$. The remaining images (b) - (f) show the course of the path lines from seeding time until $140ms$, $200ms$, $300ms$, $350ms$, $500ms$, $700ms$. All images © 2012 IEEE.

cycle the high velocities occur, the line color mapping can be switched from velocity to time (see Fig. 5.6b). Thus, we provide a method to quickly identify the flow with unusually high velocity. With this information, these areas can be further inspected with stream lines of the respective timesteps or 2D analysis planes, for example.

A further specialty of this dataset is a vortex in the proximal descending aorta [107], which is depicted in Figure 5.7. A time-dependent vortex predicate identifies path lines running through the vortex area detected in the third timestep of this dataset ($t = 102ms$). The single images show the swirling of the path lines around the core and the temporal development of the vortex.

We are certain that the structuring of the flow according to vortical flow and velocity as presented in this section, is not possible with other methods yet.

5.4.1.3 Aneurysm in Descending Aorta

This patient (19, male) underwent surgical repair of aortic coarctation several years ago. During a regular follow-up with MR angiography the development of a post-operative aneurysm in the descending aorta had been observed, which was further examined with 4D PC-MRI (spatial resolution: $1.77 \times 1.77 \times 2.6 \text{ mm}^3$; temporal resolution: 40 ms ; 20 timesteps; velocity encoding: $150 \frac{\text{cm}}{\text{s}}$) [40].

In the aneurysm a strong vortex prevails and influences the flow patterns in the whole aorta. Since the vortex takes up the whole aneurysm over large portions of a cardiac cycle, it is interesting to analyze to what extent the blood flow is interrupted by this. The residence time of the particles in this vortex is a measure for the, probably decreased, efficiency of the blood transport and is very likely to correlate with the risk of blood clot development. We can examine this aspect with the residence time predicate. We preselect all path lines of a certain minimum length (length predicate with $l_1 = 10 \text{ cm}$) crossing a region of interest in the aortic arch during the second timestep (time-dependent ROI predicate). This set of path lines is then divided into lines residing in the aneurysm for more or less than 150 ms respectively (see Fig. 5.8a). With the residence time predicate, we could also identify a large number of particles staying in the aneurysm for more than 300 ms which corresponds to the duration of an average systole (not shown). ROI definition, predicate evaluation, and visualization take no longer than 2 min altogether and allow insight into the residence times of particles and their paths. This information is of interest when assessing the risk of blood clot development or the heart's pump efficiency.

For the estimation of the risk of an aneurysm rupture, high-velocity jets are of interest. The application of the mean velocity predicate to stream lines at $t = 180 \text{ ms}$ shows the high-velocity flow faster than $80 \frac{\text{cm}}{\text{s}}$ during that timestep (see Fig. 5.8b). Here, it is especially interesting that these fast stream lines are directed at the aneurysm wall which is an indication of high forces on the vessel boundary. This flow behavior is likely to be connected to the aneurysm development.

Further, we show how line predicates can be used to inspect flow into the main branches of the aortic arch. Assuming that we want to inspect the flow of particles entering the aorta during early to mid-systole, we select a ROI at the proximal ascending aorta and apply the time-dependent ROI predicate ($t = 100 \text{ ms}$). Subsequently, these path lines can then be subdivided with one ROI predicate for each branching vessel (see Fig. 5.8c,d). It can be learned from the result that, e.g., blood entering at the ventral part of the aorta flows into the left carotid artery (blue lines), that parts of the vortical flow in the ascending aorta find the way in the brachiocephalic artery (white lines), and that a considerable vortex evolves at the entry to the left subclavian artery (yellow lines). With current methods, findings like these are only possible by observing a larger number of path lines over time and visually detecting the interesting patterns. However, it is not possible to preselect and visualize only the lines that are of interest, i.e., that flow into a certain region, as we provide it here.

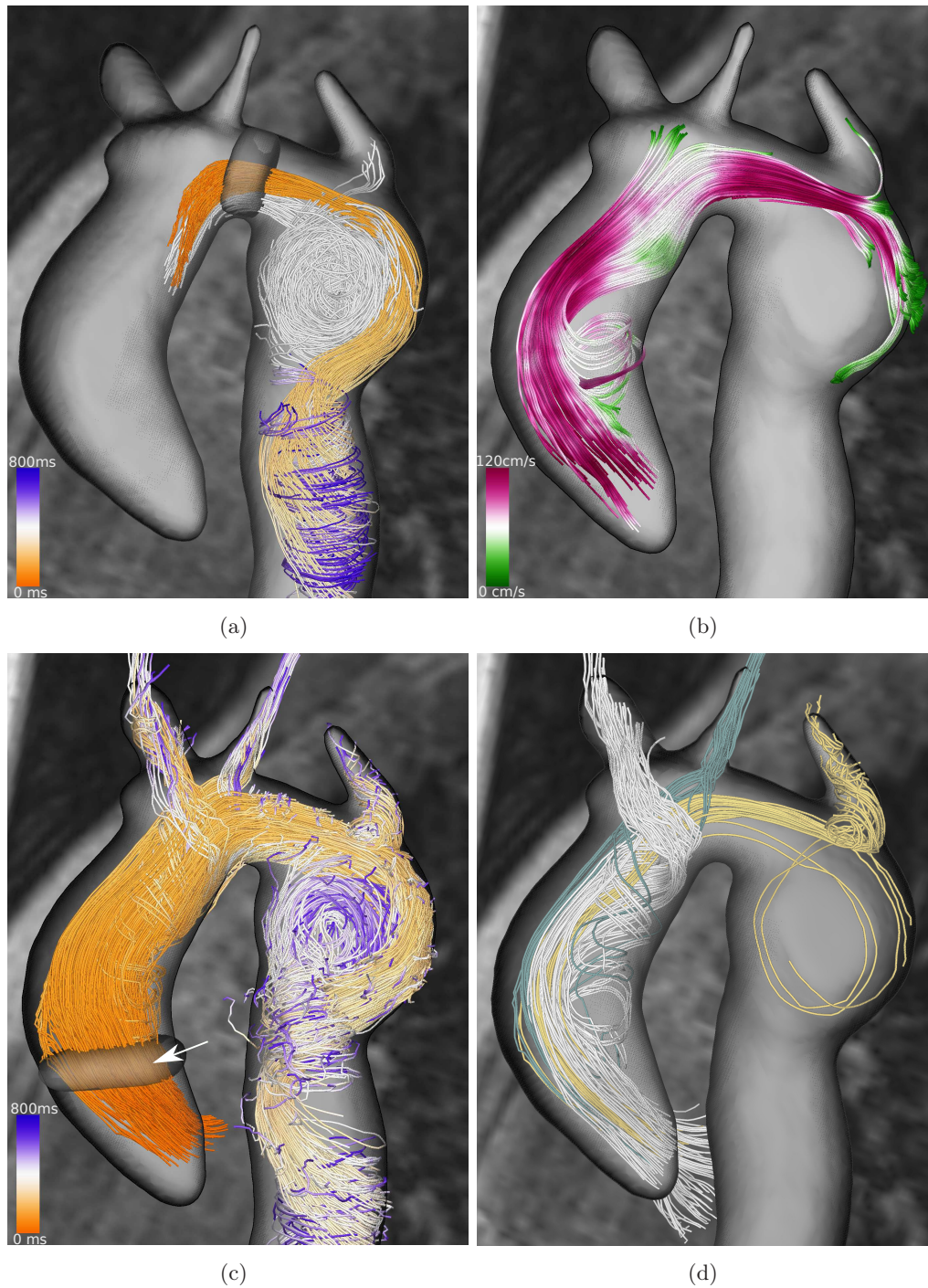


Figure 5.8: Aneurysm in the descending aorta. (a) Gray lines represent particles residing in the aneurysm for more than $150ms$. Color-coded lines pass the aneurysm in less than $1500ms$. (b) Stream lines (at $t = 180ms$) with mean velocity greater than $80 \frac{cm}{s}$ are directed at the aneurysm wall (white arrow). (c) Path lines crossing the displayed ROI (white arrow) at $t = 100ms$. (d) Subdivision of these lines into path lines entering the brachiocephalic artery (white), the left carotid artery (blue), and the left subclavian artery (yellow).

5.4.2 Heart Cases

In the previous section, we showed the usefulness of our line predicate method for the analysis of the aortic blood flow. Here, we show that our method is also very valuable for the analysis of blood flow through the human heart. For this, we analyzed two 4D PC-MRI datasets of the human heart and present the result in the following. More specifically, we discuss the different phases of a typical heart cycle in a healthy heart and show abnormal flow patterns in a heart after the repair of a Fallot tetralogy.

To handle image artifacts common for 4D PC-MRI data (noise, eddy-currents), the datasets have already been preprocessed with methods based on Walker et al. [190]. After that, heart segmentation and ROI definition, precalculation of integral lines, and vortex detection is accomplished as described previously in Section 5.3.1. For a typical 4D PC-MRI heart dataset, the preprocessing phase takes around 1.8 hours. The additional overhead for the manual segmentation depends strongly on the anatomical knowledge of the user and may last several hours. For a potential use in the clinical or research routine, this time needs to be significantly reduced by using advanced segmentation techniques and accelerating the line integration, e.g., by using the GPU.

5.4.2.1 Healthy Heart

In this section, we discuss the application of our line predicate method to a 4D PC-MRI scan of a healthy volunteer (spatial resolution: $2.97 \times 2.97 \times 3.0 \text{ mm}^3$; temporal resolution: 38.4 ms ; 24 timesteps; velocity encoding: $150 \frac{\text{cm}}{\text{s}}$; length of the covered heart cycle: 920 ms).

In Section 5.3, we briefly explained the heart anatomy and the cardiac cycle. In the following, we present different aspects of systole and diastole in the left and right heart. With current 4D PC-MRI analysis techniques, these characteristics can only be shown with experience and time – if at all.

During a cardiac cycle, the right atrium is filled with deoxygenated blood from the lower and upper body via the inferior and superior vena cava respectively (see light arrows in Fig. 4.2). At the same time the right ventricle contracts and blood is pumped into the pulmonary artery (*systole*). The tricuspid valve between the atrium and the ventricle is closed at this moment. When the ventricle relaxes again (*diastole*), the tricuspid valve opens and the ventricle is refilled with blood – supported by a light contraction of the atrium. This flow behavior can be easily followed in the 4D PC-MRI data of the healthy human heart by applying various line predicates:

Right Heart: Atrial Filling and Ventricular Ejection During atrial filling, a vortex evolves in the center of the right atrium. Figure 5.9a shows the result of the respective vortex predicate applied to stream lines at $t = 172 \text{ ms}$. From this image, it becomes clear that this vortex is mainly fed by the blood coming from

the inferior vena cava. To examine the behavior of stream lines entering the atrium through the superior vena cava at the same time, a ROI predicate is used. Figure 5.9b depicts the course of these lines. They draw near the vortex area (gray lines) but do not participate in the rotation. The two selected line bundles are disjoint.

At this point in time, the tricuspid valves between atrium and ventricle are closed. That the stream lines are traversing this area all the same, is due to the voxel side length of $3mm$ in combination with very thin valves. The 4D PC-MRI data does not cover the anatomical and flow situation in the necessary level of detail due to the partial volume effect. To still view stream lines in only one of the chambers, one can use a negative ROI predicate, i.e. a predicate that displays all lines *not* running through a specific area. This was done in Figure 5.9c; in combination with a velocity predicate selecting all lines with a maximal velocity $v_{max} > 90 \frac{cm}{s}$. Thus, the shown stream lines represent the fastest flow from the right ventricle into the pulmonary artery while avoiding the right atrium (used as ROI). From this image, one can learn, e.g., that, at this point in time, the fastest flow happens in the right branch of the pulmonary artery. Figure 5.9d shows that already one timestep later the outflow into the left and right pulmonary arteries is more evenly distributed.

Right Heart: Ventricular Filling Later in the heart cycle during diastole, the ventricular outflow slows down and the heart relaxes. Finally, the tricuspid valve opens and the inflow from atrium into the ventricle begins (see Fig. 4.2, right). This occurrence can be nicely visualized with several path line predicates: Figure 5.10a shows path lines flowing from right atrium into the right ventricle at any time during the cardiac cycle, i.e., two unsteady ROI predicates for atrium and ventricle are combined. From this image, one can get a vast impression of the unsteady blood flow. In Figure 5.10b-e, an additional vortex predicate can be used to reduce the number of lines. Here, path lines are selected with a time-dependent vortex predicate that run through a vortex detected in the right atrium at $t = 403ms$. Figure 5.10b,c show that these particles enter the atrium mainly through the inferior vena cava and start rotating in the vortex. As the valves open (see Fig. 5.10d), the vortex transforms into a helical inflow into the right ventricle that starts to form a circular vortex, assumedly as the particles hit the opposite wall of the ventricle (see Fig. 5.10e).

The left heart pumps oxygenated blood from the lungs into the body (see dark arrows in Fig. 4.2). Oxygenated blood leaves the lungs and enters the left atrium through the pulmonary veins. During the filling of the atrium, the left ventricle contracts and blood is ejected via the aorta into the body. After the left ventricle relaxes, the mitral valve between the left atrium and the left ventricle opens and the ventricle is refilled. Again, we present selected aspects of this process by applying various line predicates.

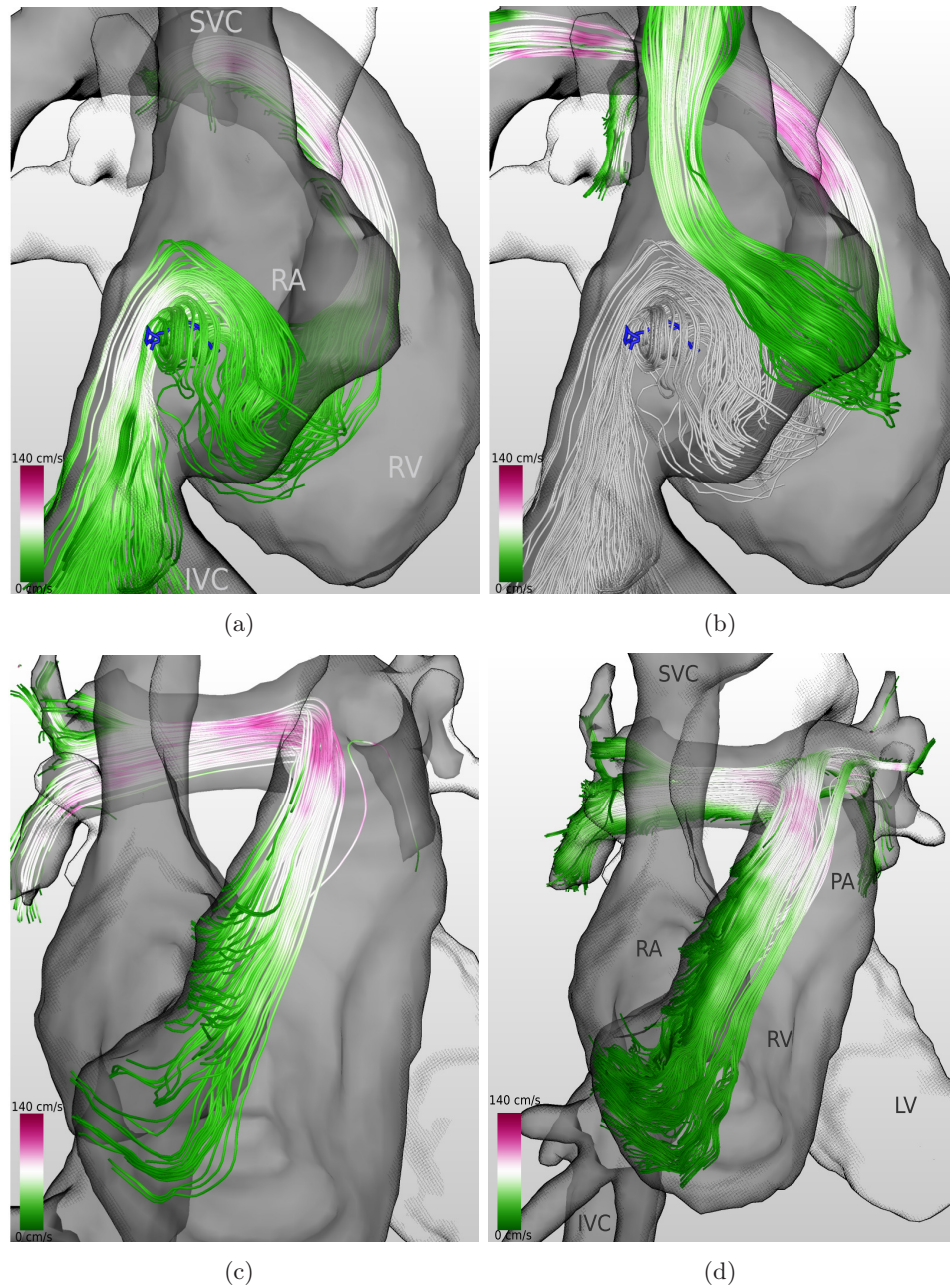


Figure 5.9: Blood flow in the right heart during systole at $t = 172ms$. (a) Stream lines running through a large vortex (blue core line). (b) Stream lines entering the right atrium through superior vena cava. The vortex is now shown as gray lines. The two bundles are disjoint. (c) Selection of fastest stream lines ($v_{max} > 90 \frac{cm}{s}$) not running through the right atrium. Fastest flow at this time is in the pulmonary artery towards its right branch. (d) One timestep later, the fast blood flow is more evenly distributed into the left and right pulmonary arteries, © 2013 IEEE. The color-coding represents velocity. Abbreviations: RA, right atrium; RV, right ventricle; SVC, superior vena cava; IVC, inferior vena cava.

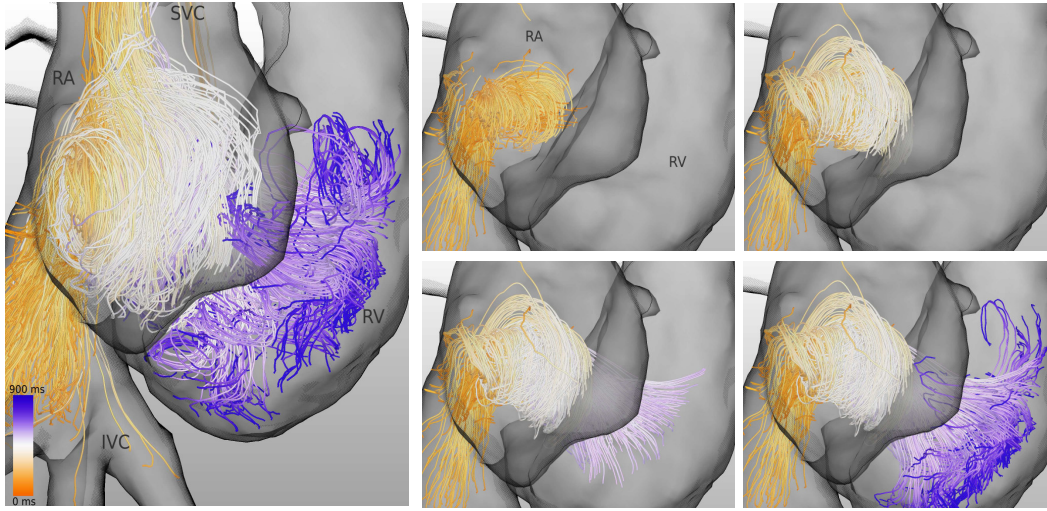


Figure 5.10: Path lines in the right heart. Color-coding represents time. Left: All path lines running through atrium and ventricle with $l > 20\text{cm}$. The images on the right show path lines running through the large vortex in the atrium at $t = 403\text{ms}$. Path lines are shown from beginning of the cardiac cycle until $t = 211\text{ms}$, $t = 326\text{ms}$, $t = 556\text{ms}$, and the end of the cycle (top left to bottom right). The pictures show the transformation of the large vortex in the right atrium into a helical inflow into the right ventricle. All images © 2013 IEEE. Abbreviations: RA, right atrium; RV, right ventricle; SVC, superior vena cava; IVC, inferior vena cava.

Left Heart: Ventricular Ejection Figure 5.11 shows the phase of blood ejection from the left ventricle into the aorta at $t = 172\text{ms}$. In Figure 5.11a stream lines are displayed that are faster than $120 \frac{\text{cm}}{\text{s}}$ and run through the left ventricle. The left heart is responsible for the distribution of blood into the whole body whereas the right heart pumps the blood into the nearby lungs. This explains why the stream lines during systole are significantly faster in the left heart than in the right heart, as shown in Figure 5.11b. Here, the fastest stream lines of the right ventricle are displayed as well. These lines are selected with a maximal velocity predicate $v_{\max} > 90 \frac{\text{cm}}{\text{s}}$. The application of the same velocity predicate as for the left heart ($v_{\max} > 120 \frac{\text{cm}}{\text{s}}$) shows no result.

The residence time predicate can be used to identify particles staying in an anatomical area for longer than a specified time interval. This is of medical interest since blood pools which reside somewhere for too long might develop blood clots. Blood clots are the major cause for stroke and heart attack and the recognition of a high risk of blood clot development is of great importance during diagnostics. In Figure 5.12a, path lines are shown that run through the left ventricle at $t = 96\text{ms}$, i.e., at the beginning of the cardiac cycle. In Figure 5.12b, only those path lines are selected from the bundle which stay in the ventricle for the complete cycle.

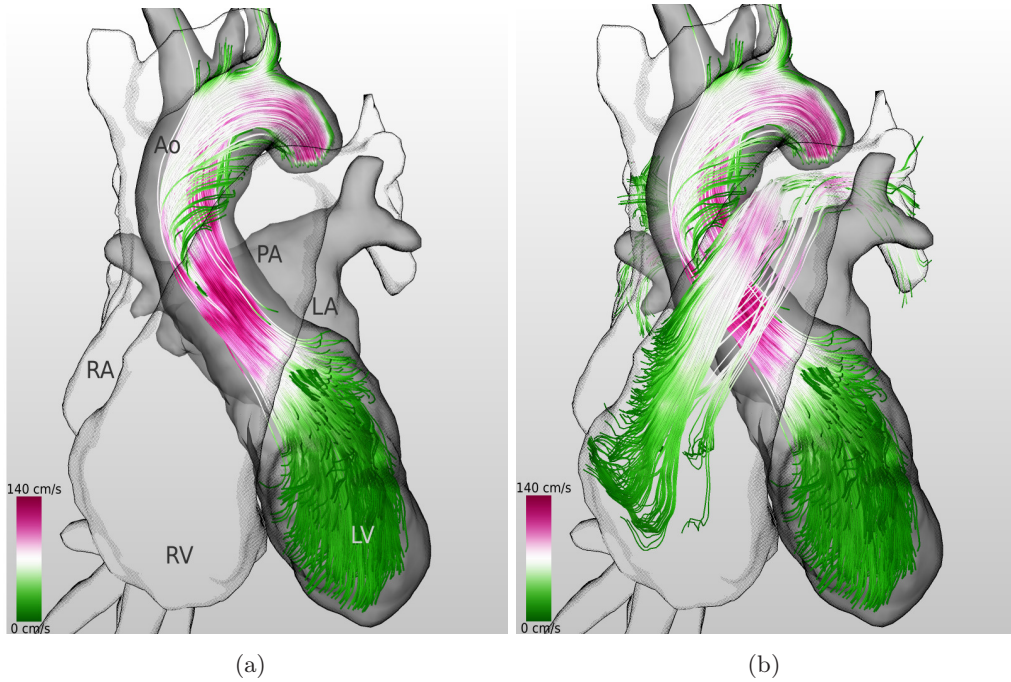


Figure 5.11: Stream lines in the healthy heart at $t = 172ms$. Color-coding represents velocity. (a) Stream lines in the left heart which are faster than $v_{max} > 120, \frac{cm}{s}$. (b) Additionally, stream lines in the right heart are faster than $v_{max} > 90, \frac{cm}{s}$. This confirms that the left heart ejects faster than the right heart (shown for only point in time). Abbreviations: Ao, aorta; PA, pulmonary artery; LV, left ventricle; LA, left atrium; RV, right ventricle; RA, right atrium.

Left Heart: Flow in the Left Atrium Blood flow in the left atrium is - just like in the right atrium - strongly affected by vortices. By automatically detecting these vortices and showing the stream lines running through them for several points in time helps in understanding the atrial flow. For patient datasets this may in the future help to identify abnormal flow structures correlating with certain pathologies (such as mitral valve insufficiencies, atrial fibrillation, or the like).

In Figure 5.13, this blood flow behavior is depicted for four separate timesteps: At the beginning of the cardiac cycle, a vortex can be noticed which remains until the mitral valve opens and blood is transported into the ventricle. Over time, the vortex area gains in size and shifts to the atrium's center (see Fig. 5.13a,b). At $t = 480ms$, the mitral valve has opened and the ventricle is refilled with blood. This involves a significant acceleration of blood. Figure 5.13c depicts the course of the fastest stream lines (with $v_{max} > 100 \frac{cm}{s}$) at that point in time. After the mitral valve is closed again, a vortex develops at about the same position as before. It is accompanied by a second vortex. Its core line runs about orthogonally to the core line of the first vortex (see Fig. 5.13d).

Some of the shown flow structures might also be observable with current analysis techniques, i.e., they can be spotted by an experienced user in a larger number of

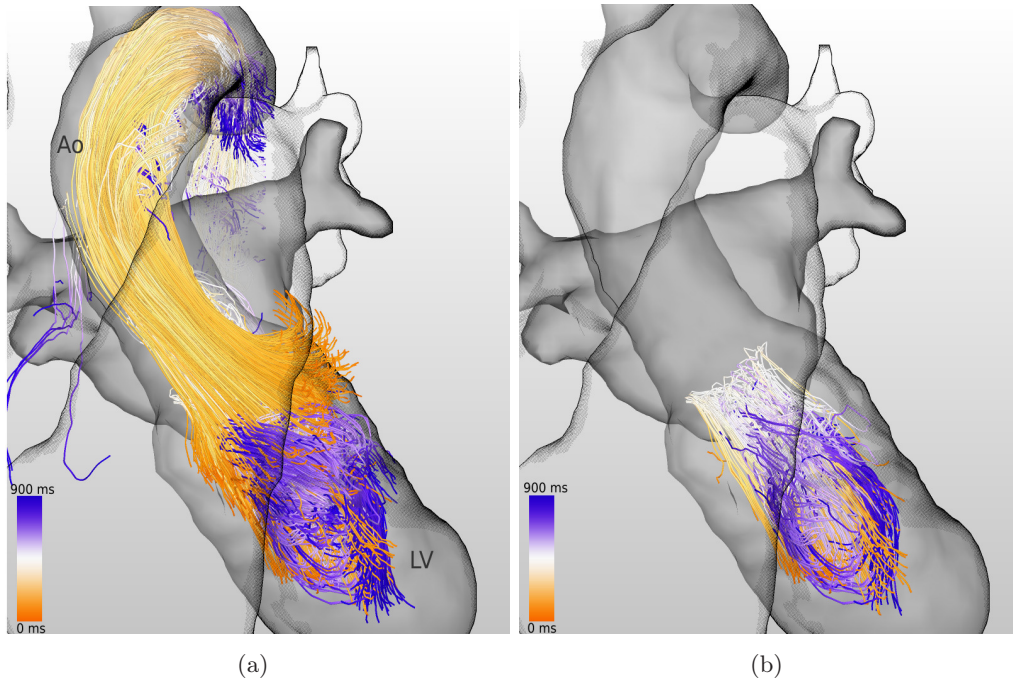


Figure 5.12: Path lines in the left healthy heart. Color-coding represents time. (c) Path lines representing particles that are in the left ventricle at $t = 96ms$ (d) Subset of these path lines which do not leave the left ventricle for the remaining cardiac cycle. The lines are selected with the residence time predicate (with $t_r = 920ms$). Abbreviations: Ao, aorta; LV, left ventricle.

lines, as they are rather large. However, a separate, semi-automatic identification and display of these flow structures as shown here is not possible with other methods.

5.4.2.2 Tetralogy of Fallot

In this section the focus is on the analysis of a pathological case. The examined 4D PC-MRI dataset was acquired from a patient several years after surgical repair of a tetralogy of Fallot. The dataset was acquired with a temporal resolution of $38.4ms$ and velocity encoding of $150 \frac{cm}{s}$. It has a spatial resolution of $2.5 \times 2.5 \times 3.6 [mm^3]$ and covers a heart cycle of $595.2ms$ in 16 timesteps.

Tetralogy of Fallot (TOF) is a congenital heart defect involving a hole in the septum between the ventricles and a narrowing of the right ventricular outflow tract or pulmonary valve or both, as well as right ventricular hypertrophy and an overriding aorta (see Sec. 4.1.1). During surgery very early in life, the septum hole is closed and the narrowing is, in this case, treated by removing muscle tissue around the pulmonary outflow tract and integrating a transannular patch. As a result, this patient has a pulmonary valve incompetence of a certain degree, which usually increases during upgrowth.

Tetralogy of Fallot (TOF) is known to involve a strong retrograde flow from

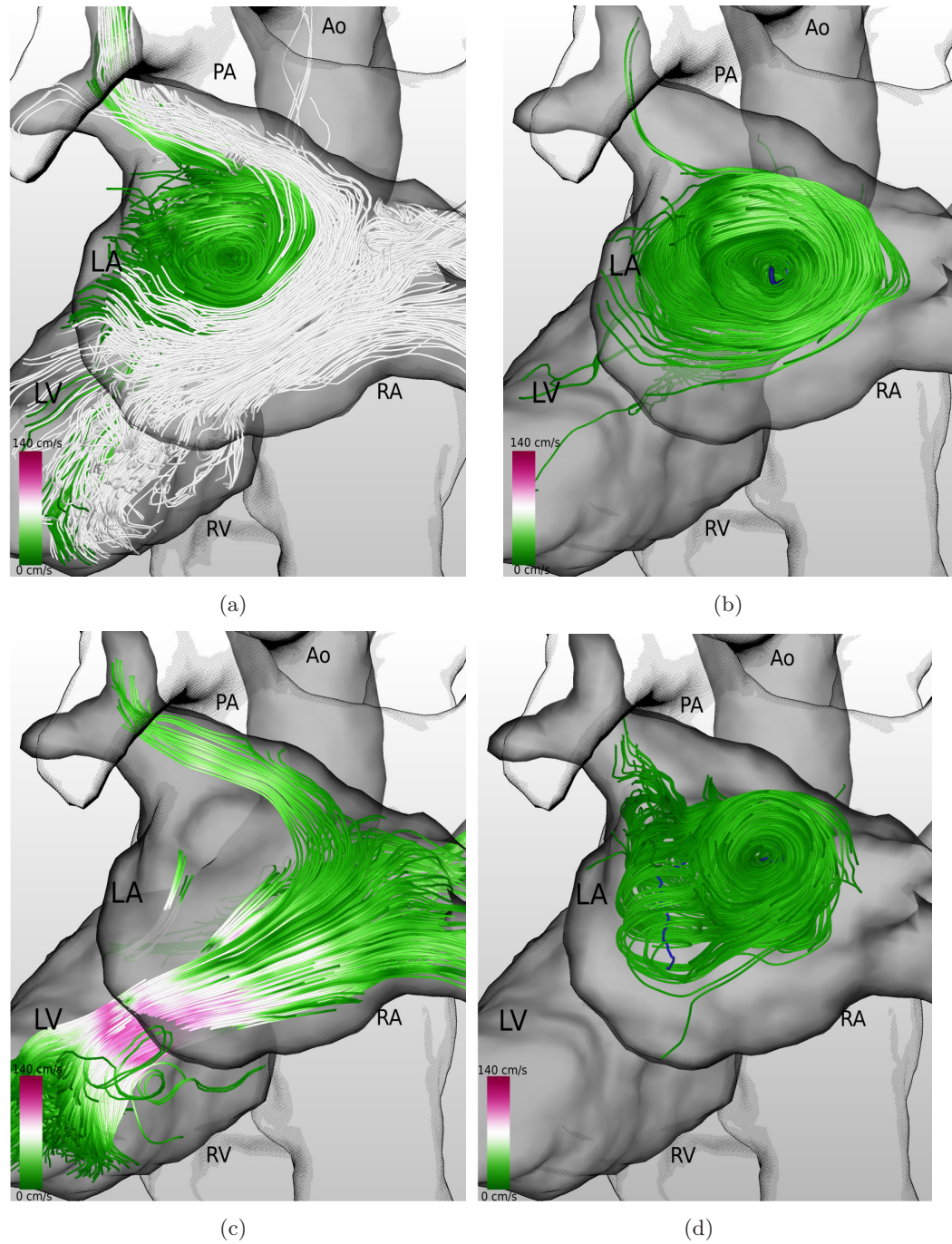


Figure 5.13: Blood flow in the left atrium at different points in time (view from the back). Color-coding represents velocity. (a) At $t = 172ms$, a vortex with a rather small vortex area is detected. Stream lines connect the vortex with the left blood-delivering vessel. Gray lines show stream lines not involved in the vortical flow, b) At $t = 326ms$, the vortex is larger but still rather slow and moves towards the atrium center, © 2013 IEEE, (c) At $t = 480ms$, blood is transported into the left ventricle. Displayed stream lines fulfill the velocity predicate $v_{max} > 100 \frac{cm}{s}$, © 2013 IEEE, (d) At $t = 672ms$, the mitral valve is closed again and two vortices evolve. The shown stream lines run through both vortex areas. Abbreviations see Fig. 5.11.

the pulmonary trunk due to the pulmonary valve incompetence back into the right ventricle during diastole as well as an imbalanced flow into the right and left lung [48, 51]. These two aspects can be very well visualized with our line predicate approach and become especially evident in comparison with the healthy cardiac flow situation.

Flow to Left and Right Lung Figure 5.9d depicts systolic blood flow in a healthy heart from the right ventricle into the pulmonary arteries. The stream lines are selected with the maximal velocity predicate ($v_{max} > 90 \frac{cm}{s}$). The systolic blood flow shows no vortices and is almost evenly distributed to the left and right pulmonary artery. The pulmonary trunk of the TOF patient, however, shows bulges that have developed since surgery. Vortex detection shows large vortices in these bulges over the majority of the cardiac cycle. In Figure 5.14a stream lines running through the vortex, detected in the upper bulge during systole, are depicted. It stands out that this vortex strongly correlates with the vortex below and that all stream lines are directed towards the right pulmonary artery. Further, a cavity in the line bundle can be noticed. Applying a maximal velocity predicate to the same stream lines shows a strong high-velocity jet ($v_{max} > 165 \frac{cm}{s}$) running through the cavity of the previous bundle into the left pulmonary artery (see Fig. 5.14b). Thus, with only these two predicates, one can observe that the right pulmonary artery is affected by the slower vortical flow whereas the left pulmonary artery is directly supplied with fast-flowing blood. With current techniques, the high-velocity jet, flow ratio to the left and right pulmonary artery and, to a certain extent, the large vortices can be determined, but a causal relationship between these flow features cannot be easily observed. Our findings, however, suggest that, in addition to other anatomical changes due to the surgical manipulation, one cause for the imbalanced flow ratio into left and right lung are the large vortices in the bulges. They slow down the flow to the right lung and increase the blood velocity to the left lung. Further, the detected vortices at the beginning of the pulmonary trunk may indicate that the tissue in these areas will further dilate and another surgery will become necessary in the future.

Retrograde Flow Figure 5.10a shows path lines in the healthy heart flowing from the right atrium into the right ventricle at any time during the cardiac cycle, i.e., two ROI line predicates for atrium and ventricle are combined. From this image, one can get a vast impression of the unsteady blood flow. In Figure 5.10b-e, an additional vortex predicate can be used to reduce the number of lines. Here, path lines are selected with a time-dependent vortex predicate that run through a vortex detected in the right atrium at $t = 403ms$. Figure 5.10b,c show that these particles enter the atrium mainly through the inferior vena cava and start rotating in the vortex. As the valves open (see Fig. 5.10d), the vortex transforms into a helical inflow into the right ventricle that starts to form a circular vortex, assumedly as the particles hit the opposite wall of the ventricle (see Fig. 5.10e). Note that in a healthy heart, this is the only way that blood enters the right ventricle. There is no

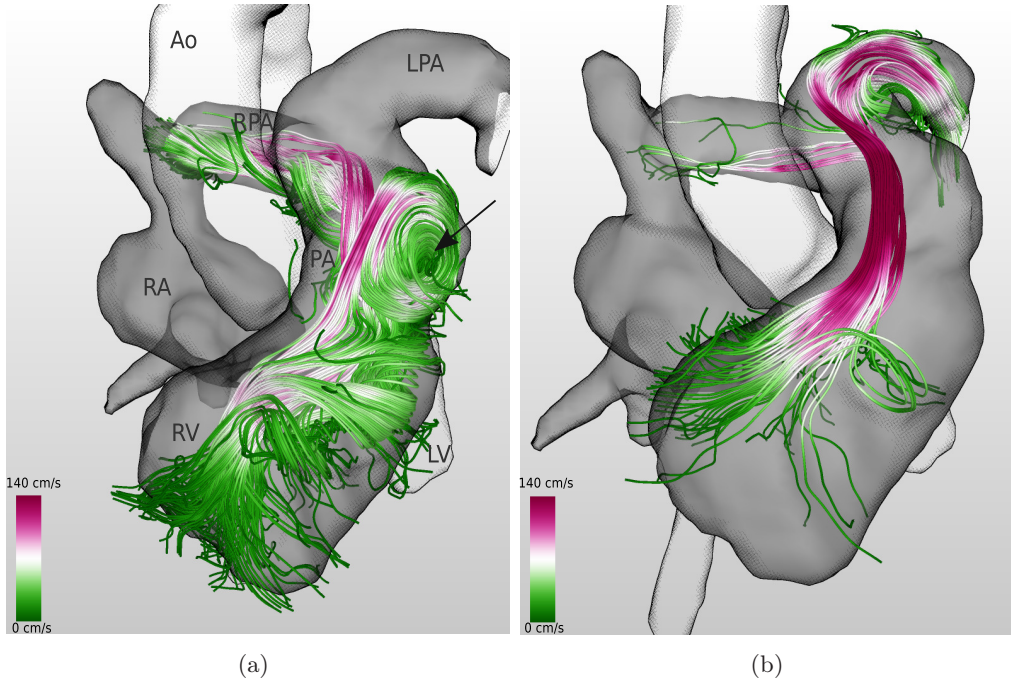


Figure 5.14: Flow ratio into left and right lung during systole. Color-coding represents velocity. (a) Flow through upper detected vortex (arrow) directed at the RPA. (b) High-velocity jet ($v_{max} > 165 \frac{cm}{s}$) directed at the LPA. Both images © 2013 IEEE. Abbreviations: RA, right atrium; RV, right ventricle; LPA, left pulmonary artery; RPA, right pulmonary artery.

retrograde flow from the pulmonary trunk.

Figure 5.15 shows the respective flow situation in the TOF patient. Due to the severely incompetent pulmonary valve, the right ventricle receives blood both from the right atrium and the pulmonary trunk. A reduction to lines running through the previously introduced vortices in the bulges of the pulmonary trunk during systole is shown in Figure 5.15a. Here, a closer look allows qualitative observations like, e.g., that the vortices in the bulges are mainly avoided by the retrograde blood flow.

The retrograde flow hinders inflow from the right atrium and thus influences the flow behavior in the right atrium in terms of higher residence times. In Figure 5.15b, blood particles are shown, which reside in the right atrium for more than 550 ms, i.e., almost the complete cardiac cycle. The same predicate applied to the healthy heart dataset shows no results. This indicates that the blood fraction leaving the right atrium is reduced in the TOF patient when compared to the healthy case. To sum up, the analysis of the healthy dataset showed that it is feasible to visualize the main phases of the cardiac cycle and that interesting aspects can be extracted. In the pathological cardiac flow, irregularities and hypothesis for their cause could be identified that could not be easily detected with current analysis methods.

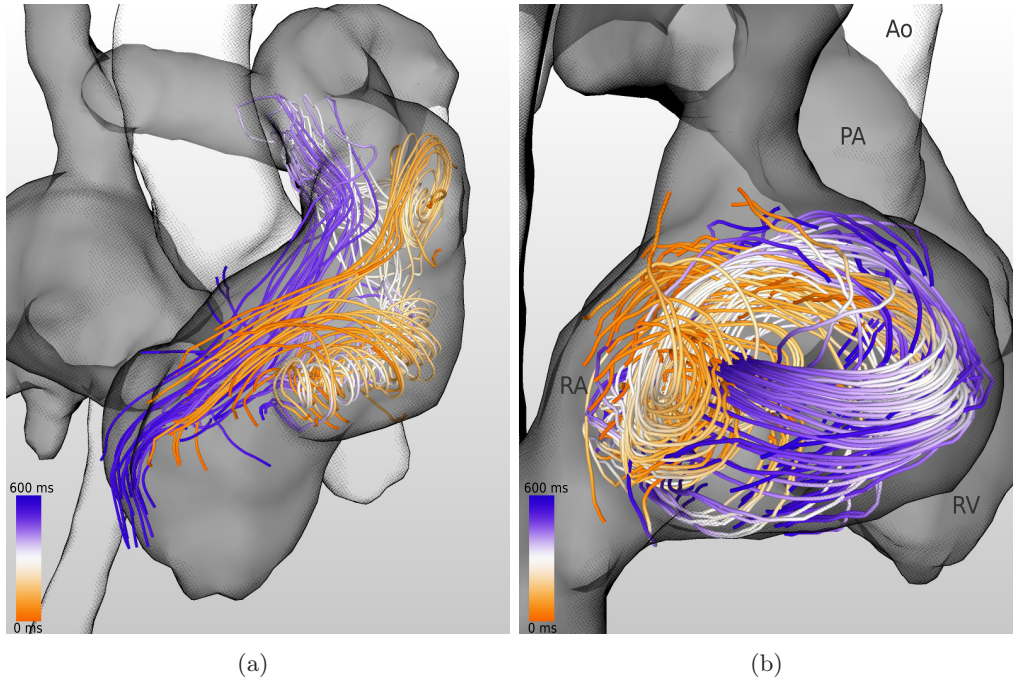


Figure 5.15: Retrograde flow in the TOF dataset. Color-coding represents time. (a) Path lines entering the outflow tract during systole, running through vortices and returning to the right ventricle in diastole. (b) Higher residence times $t_r = 550 \text{ ms}$ can be observed in the right atrium of the TOF dataset. Both images © 2013 IEEE. Abbreviations see Fig. 5.14.

5.4.3 Evaluation

In this section, we discuss the performance concerning preprocessing and run-time and present detailed expert feedback.

5.4.3.1 Performance

Selection and combination of the line predicates as well as setting the parameters is done interactively. The predicate evaluation time depends on the predicate type and the number of lines. For average cases as shown in Figures 5.10, 5.14, 5.15a, the evaluation takes less than 100 ms on a standard PC. A more expensive predicate, such as the residence time predicate, takes about 500 ms , when evaluated on a full set of path lines (as in Fig. 5.8a and 5.15b). The combination of several predicates does not have a significant effect on evaluation time because the number of evaluated lines decreases for each subsequent predicate. The complete update time, i.e., the time needed for predicate evaluation and scene update, is on average below 1 s . This increases of course with the number of rendered lines, but does not exceed 5 s for a full set of path lines. This lag could be altogether reduced with an implementation using the GPU more extensively, but in the current version a comfortable line analysis is well possible.

5.4.3.2 Expert Feedback

The feedback of the three 4D PC-MRI experts (one medical physicist, two radiologists) showed that our new method provides a yet unknown view on the cardiac blood flow data. The experts rated our approach as well-suited to supplement their current analysis techniques in principle, but also commented on drawbacks. In the following, we summarize the experts' feedback in terms of the mentioned advantages, the benefits of the different line predicates, and their suggestions for improvement.

Advantages One of the advantages that were highly valued by the radiologists, is the global view on the data. Instead of examining local flow features, e.g., on 2D cross-sections, they can inspect various large-scale features and by that identify correlations and causal relationships, which is not possible with state-of-the-art methods. They considered the overview as very important since this allows to exploit the advantage of 4D PC-MRI over 2D cine data (where only local data is acquired). Together with the semi-automatic identification of secondary flow patterns, they expect a better understanding of the flow, new insights about correlations with certain pathologies, and by that about the mechanisms of disease development. Also, it might also help to improve surgical procedures because it allows insights about the effect of an intervention on the hemodynamics. [52].

The 4D PC-MRI experts suggested specific research questions where our method could bring a benefit. One such research topic is atrial fibrillation, as it is not yet totally understood why this disease oftentimes correlates with a high risk of blood clot generation, which can lead to stroke or heart attack. Further, they were interested in a flow feature analysis with our technique before and after treatment (ablation) of atrial fibrillation with the goal to improve this intervention and to understand the underlying mechanisms.

While current analysis methods mainly rely on the flow interpretation by the user, a standardized flow analysis is necessary for the clinical routine or the more precise evaluation of research studies. The experts considered our method suitable as basis for such a standardized flow analysis. A more automated approach with pre-defined line predicate combinations could save time and decrease inter-observer variability.

Finally, the experts rated our visualizations as comprehensible and hence saw an advantage in using these images in publications.

Benefit of Predicates The experts' main interests when examining blood flow are the velocity distribution, residence times, and irregularities in general – depending on the dataset at hand. In the following, we describe the usefulness of each of the predicates during flow examination as stated by the experts.

The ROI predicate is useful to allow the examination of the blood flow in or towards a specific area of interest. In most cases, this is applied in combination with other predicates. The length predicate is considered as a supplementary predicate as it is mainly used to reduce clutter.

The velocity predicate is very useful when examining the flow velocity distribution that is one of the medical expert's main interests when it comes to cardiac blood flow. They appreciated the velocity predicate to detect high-velocity flow indicating narrowings (stenosis).

The residence time predicate is better suited to detect blood flowing very slowly or even coming to a halt (see Fig. 5.8a and 5.15b). It is considered highly useful as very slow flow areas may correlate with a lower heart efficiency (e.g., the retrograde flow in the TOF dataset) and a higher risk for thrombogenesis.

The vortex predicate is of special interest for the experts, as irregular vortices are a strong indicator for a pathology or for a higher risk of aneurysm development. Our vortex detection in combination with the vortex predicates allow to analyze vortices more systematically. This was rated as a big advantage by the experts, as they have no possibility for vortex analysis right now – except the visual observation of integral lines.

Disadvantages and Improvements A drawback of our method is the necessary preprocessing time. The experts tolerate about 1h of preprocessing, which we do not reach yet mainly due to the segmentation overhead. Interestingly, the experts did not claim for a fully automated segmentation approach. Instead, they prefer methods allowing them to interact and apply their medical knowledge to adapt the segmentation. The goal should be a semiautomated segmentation approach with advanced interactive tools for well-trained users in cardiac anatomy.

Another limitation concerns the analysis of smaller flow structures, such as around the valves. Here, it is not yet possible to reliably determine whether blood is flowing through opened or leaky valves or whether integral lines are wrongly crossing the closed valves. These inaccuracies are caused by noise and partial volume effects. Our regional velocity predicate deals with this effect in the sense that one can still evaluate lines in a specific area, although it leaks into neighboring chambers. Apart from that, the flow structures detectable with our approach depend mainly on the data quality. Thus, our approach will also benefit from the continuing development of the 4D PC-MRI technique.

In the clinical routine as well as in the medical research, conclusions are preferably drawn based on quantitative parameters like flow rate or blood velocity, instead of qualitative observations. The 4D PC-MRI experts view our method as a valuable supplement to their methods at hand. It would be even more valuable with a quantification of the detected flow features, as this would, e.g., allow the comparison of flow features in healthy and pathological datasets.

Further suggestions are the use of a color-coding convention that is used with, e.g., Doppler-Ultrasound, that depicts both flow direction and velocity at the same time. It is thinkable to depict blood flowing from and towards the heart with red and blue respectively.

Concerning the lack of animations in our method, the experts did not agree with each other. On the one hand, it was noted that the omission of animations is an

advantage for a standardized data evaluation and easier to integrate in the clinical and research routine. On the other hand, an easy, maybe animated, way to step through the streamline depictions of the different timesteps was missed.

Finally, as already mentioned before, the experts encouraged us to work towards standardized predicate combinations, which can be used when analyzing larger amounts of data.

5.5 Discussion and Future Work

4D PC-MRI is a relatively new imaging modality, which allows to gain insight into the 3D blood hemodynamics of the cardiovascular system. With that, it has the potential to significantly enhance diagnostics and therapy of cardiovascular diseases. A challenge however is to overcome problems in the data analysis, which still requires a lot of time and expert knowledge. Medical scientists have asked for more automated and standardized visualization techniques to open the door for the application of 4D PC-MRI in the clinics or for larger patient studies [110]. This request incorporates methods that are capable of giving the expert a quick overview of the main characteristics and any anomalies of the respective dataset. We believe that the development of methods illustrating the overall flow behavior in 3D is an important step towards this goal.

Current analysis techniques usually rely on the seeding of integral lines from 2D planes and therefore have two major problems. First, the seeding pattern on the plane strongly influences the features that can be detected. Second, the tracking of many integral lines leads to visual clutter and makes it difficult to spot the interesting aspects. With our approach, the seeding problem does not evolve and the clutter problem is reduced by focussing on particular flow patterns. Our approach is based on the fact that the main behavior of a flow field is encoded in its full set of integral lines. To make the flow visualization more comprehensible, we segment these lines into groups of similar behavior. For this grouping, we defined line predicates based on the medical experts' needs. These predicates allow to structure the precomputed integral lines according to certain flow characteristics, such as velocity, anatomical passage areas, and vortices. Users can freely combine these predicates and create flow structures answering their questions about the current dataset.

At first, we applied this method to aortic data, which is easier to handle and analyze. Then, we adapted our approach to cardiac 4D PC-MRI blood flow analysis. Here, the challenges are the deformed shape due to the heart beat, the high complexity of the flow, and the lower spatial resolution of the 4D PC-MRI heart data. To cope with them, we adapted the preprocessing, added new features such as the predefinition of meaningful anatomical ROIs, reshaped and added line predicates, and advanced the flow structure visualization. By means of 4D PC-MRI datasets of healthy and pathological cases, we could show that our method allows to easily carve out and display structures of aortic and cardiac flow and to draw conclusions that could not be drawn with other methods presented in literature so far. Three

4D PC-MRI experts gave very positive feedback about our method and stressed the benefit they see for the analysis of 4D PC-MRI data and the medical research on cardiovascular diseases. Still, the preprocessing time is a limitation that will be reduced in the future by introducing an advanced segmentation approach.

In addition to this, there are many other interesting topics for future work that were partly suggested by the medical experts. In order to further improve insight and overview, a reduction of line clutter could be achieved by displaying a line bundle's flow behavior with less lines or more abstract methods (see Chap. 6). As this approach is limited to the detection of features covered by the predicates, more predicates can become necessary when examining other anatomical structures. So, the development of new predicates and the design of the aforementioned standardized flow analysis with predefined predicate combinations appears as an interesting research field. Finally, our visual analysis method helps in gaining overview and understanding the data's overall flow dynamics, but is qualitative so far. Still, for diagnostics and therapy decisions quantitative flow measures are necessary as well. We are confident that our method is suited for an expansion towards a combined qualitative and quantitative analysis of 4D PC-MRI data.

Finally, vortex detection was not the main focus of this work and there are still open questions with regard to the parameterization of the used methods. Also testing other established or new approaches for vortex extraction is an interesting field of research. Since the publication of this work [15, 16], others have picked up this research question as well as the line predicate approach and presented valuable results about the semi-automatic vortex extraction in 4D PC-MRI data. Köhler et al. defined new types of predicates (e.g., smoothing predicates) and formulated several vortex predicates based on established local vortex criteria. They found that λ_2 gave the best results and combined this method with a user-controlled postprocessing step to improve visual appearance [84]. The work of Köhler et al. demonstrates the flexibility of line predicates and with that nicely confirms our approach as well.

Illustrative Visualization of 4D PC-MRI Blood Flow Data

4D phase-contrast MRI (4D PC-MRI) delivers valuable blood flow information, which for the first time provides a three-dimensional insight into the hemodynamics of cardiovascular diseases. Consequently, many advances concerning the diagnostics and therapy of cardiovascular diseases are expected from this imaging modality. However, a prerequisite for successful medical research is the availability of appropriate data analysis tools, which are designed for the particular needs of the medical users. A quick, reproducible, and comprehensible blood flow analysis is needed. Unfortunately, current methods are mainly based on techniques used in the classical fluid dynamics and the resulting images are usually very technical and addressed at an audience familiar with the concepts of flow visualization. Thus, it can be difficult to understand these depictions and to draw reasonable conclusions, especially for users without distinct flow experience. So, we aim for a comprehensible image analysis for users with a substantiated medical and anatomical knowledge, but without an exceptional interest in flow dynamics.

Illustrative methods are the obvious choice for this attempt, as their main purpose is to effectively convey complex information. These methods are inspired by the work of illustrators and try to adapt the techniques that make the artists' work comprehensible. In Figure 6.1 hand-drawn illustrations of the blood flow through heart and aorta are shown. Philip Kilner created them to visualize his findings from 4D PC-MRI studies. The fact that he chose this way of communication, instead of using the images produced by the analysis software, also indicates that current flow visualization methods do not satisfy all expectations concerning comprehensibility and esthetics yet.

Here, we present an approach for the illustrative depiction of 4D PC-MRI blood flow. In contrast to other work aiming at, e.g., a better perception with enhanced rendering techniques, we aim at a higher level of abstraction and information reduction as it was used in the hand-drawn illustrations. We want to depict meaningful flow features relevant for the medical user, in an easy and intuitive way. At the same time, we neglect insignificant and distracting details.

The basis for this illustrative blood flow visualization is formed by the line predicate approach presented in the last chapter. This approach already provides an initial reduction of flow information to the aspects of interest. The result are bundles of integral lines with similar flow characteristics, such as, e.g., passing the same vortex region or a particular velocity. But since not every single line is of interest,

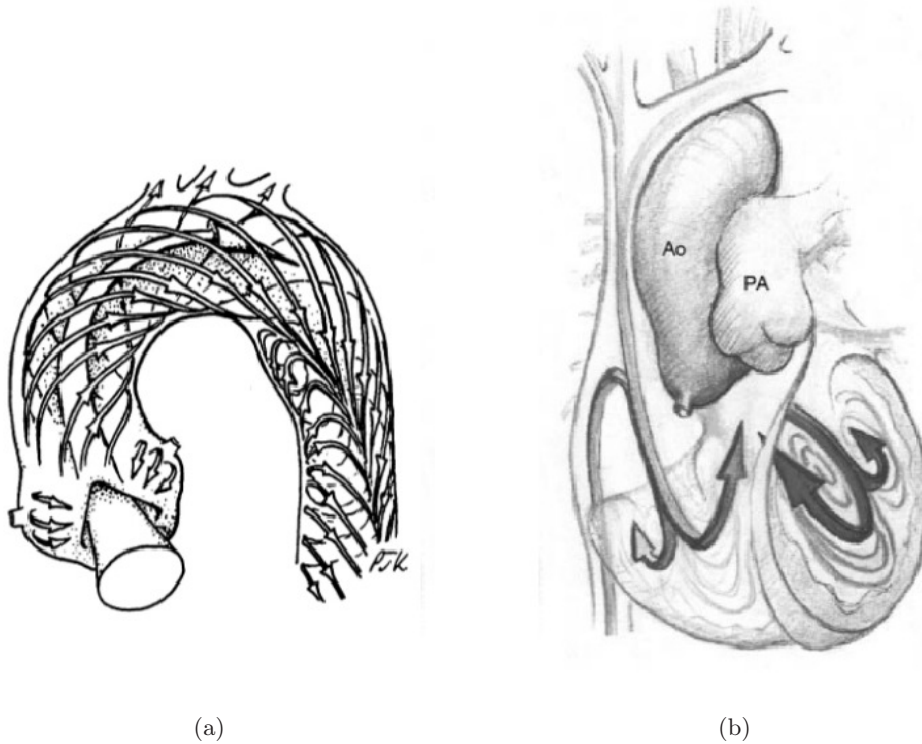


Figure 6.1: Illustrations by Philip Kilner showing blood flow through the aorta and the heart observed in 4D PC-MRI measurements. Figures from (a) Kilner et al. [80] used with the permission by Lippincott Williams and Wilkins/Wolters Kluwer Health, and (b) Kilner et al. [81] reprinted by permission from Macmillan Publishers Ltd: Nature, © 2000)

we suggest a method that reduces a bundle to a minimal set of lines representing the main features of the complete bundle. We control the extraction of the representatives by applying previous knowledge about vortex regions and by that avoid the neglect of important areas. We render these lines as three-dimensional arrows that diverge and converge if necessary. As the presence and location of vortices is another relevant flow aspect, we display vortices in a reduced and cartoon-like fashion. In summary, our main contributions are:

- We depict steady flow features derived from 4D PC-MRI blood flow in an abstract and comprehensible way to support the user with flow interpretation.
- For that, we reduce a flow feature bundle to a minimal set of lines representing its main flow behavior.
- We render the representatives illustratively as three-dimensional arrows. To avoid redundancy, the arrows are fused when they represent the same flow regions.

- We add vortex information to the arrow renderings to provide further context. The vortices are shown as tube-like structures with a hatching texture resembling vortical flow.
- We discuss advantages and limitations of our abstraction method by means of several example flow structures occurring in cardiac and aortic blood flow.

In the next section, we present our abstraction method and explain how we render the lines and vortex regions. In the result section, we show illustratively depicted flow of 4D PC-MRI data of heart and aorta and report about expert feedback. Finally, we discuss our method and present several starting points for future work. This chapter is mainly based on [14].

6.1 Related Work

Due to the fact that flow visualization is concerned with complex data, illustrative methods have found general approval in this research field (see Sec. 2.3). Also for the visualization of 4D PC-MRI blood flow, illustrative techniques have been applied.

As already discussed in Section 5.1, the visualization community introduced several advances to make the blood flow analysis more user-friendly and comprehensible. Oftentimes illustrative methods are successfully applied for this. Van Pelt et al. use illustrative methods like arrow-trails to depict time-dependent blood flow dynamics on 2D vessel cross-sections and exploded planar reformats to connect 3D and 2D views of the flow [181] (see Fig. 6.2a). In another work of van Pelt et al., illustrative methods are used to enhance the probing of a 4D PC-MRI flow field [179]. The perception of interactively injected seeds is enhanced with cartoon-like renderings and animations (see Fig. 6.2b). Gasteiger et al. introduced a ghosted view approach that displays the vessel surface and reveals the blood flow by applying a view-dependent transparency [47].

Several methods addressed the topic of data reduction. For example, van Pelt et al. applied a hierarchical clustering to unsteady 4D blood flow data to achieve a more sparse flow representation [182] (see Fig. 6.2c). A related method was presented by Chen et al. [24] (see Fig. 2.9a). They also used a clustering method to divide the flow (not 4D PC-MRI data) into areas of similar behavior and select lines representing them. Both depict their results as arrows or streamtapes, which is similar to our rendering method. However, the meaning of the resembled flow behavior is still unclear and the interpretation left to the user. Our approach is, therefore, related to these methods because we also aim at an intuitive and simple visualization. But instead of using these methods to enhance visibility or perception, we also want to give support by showing flow features with a clear meaning and, thus, help in interpreting the blood flow data.

Brambilla et al. have recently published a survey about the numerous work in the field of illustrative flow visualization [18] (see Sec. 2.3). They classified the different approaches with respect to the abstraction level of the data and the kind

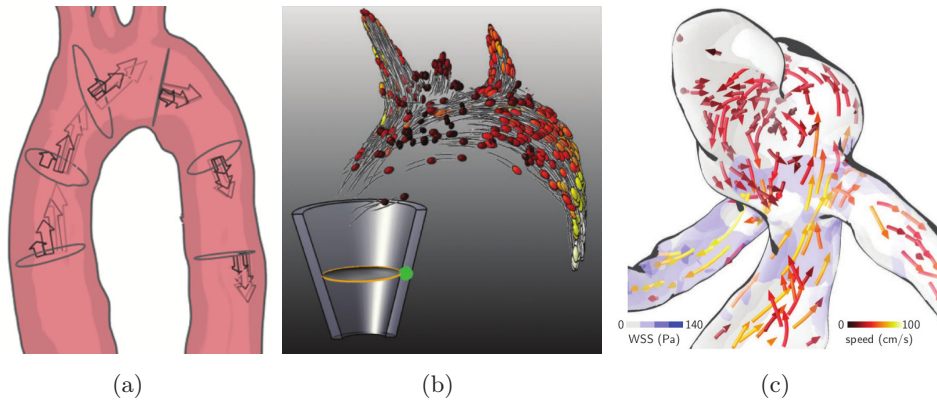


Figure 6.2: Related work of illustrative blood flow visualization. (a) Arrow-trails showing unsteady blood flow at 2D cross-sections [181], © 2010 IEEE, (b) virtual flow probing with cartoon-like seeds [179], © 2011 IEEE, and (c) hierarchical blood flow clustering [182], used with permission by John Wiley & Sons.

of applied visual abstraction (perceptual effectiveness, visibility management, focus emphasis, visual explanation). It became obvious that both, approaches for the illustrative visualization of flow features and approaches aiming at support for flow interpretation (visual explanation), are still seldom. However, this is exactly our goal and we aim for visualizations that ease flow understanding and can also be used for the visual communication of findings between colleagues.

After submission of this work, Gasteiger et al. published work about the detection and visualization of flow patterns in cerebral aneurysms. They picked up the idea to use line predicates for blood flow visualization and used them to detect inflow jets and the impingement zone. They rendered the inflow jets also as abstract arrow glyphs [46]. With that, this work is related to our approach. However, we aim at producing abstract visualizations of arbitrary flow patterns.

6.2 Abstract Visualization of Blood Flow Features

In this section, we describe our methods for an abstract visualization of 4D PC-MRI blood flow. The input for this visualization are flow features extracted with the line predicate method shown in Chapter 5. Since the user is usually not interested in the course of single stream lines but the overall meaning of the flow feature, we process these line bundles to gain an abstract and comprehensible depiction. Here, we first discuss which requirements have to be met for an effective flow feature visualization. Then, we describe the details of our abstraction method, i.e., the determination of line bundle representatives and the rendering of these lines as well as the vortex regions.

6.2.1 Design of an Abstract Flow Depiction

Our goal is an intuitive and abstract visualization of flow features in 4D PC-MRI blood flow data. When designing such a visualization it is helpful to have comprehensible, hand-drawn illustrations in mind – like it is common practice with the illustrative paradigm. Figure 6.1 shows several drawings of blood flow through heart and aorta. The flow is depicted in a reduced way by three-dimensional arrows - either with a rectangular or tube-like cross-section. The converging and diverging of blood flow is represented by the fusion and splitting of these arrows. The aorta image (see Fig. 6.1b) shows that an arrow's width or radius can encode flow parameters such as blood flow volume or velocity. Basically, the arrows follow the main flow paths. Vortical flow that is not covered by these paths is adumbrated with thin hatchings (see Fig. 6.1c). Finally, the anatomical context can be either drawn in fine detail (see Fig. 6.1c) or in a very reduced way. In Fig. 6.1a the anatomy is shown with simple silhouettes and the heart valves are shown as rings for orientation. Thinking a step ahead, one might prefer the simple and reduced version for the depiction of anatomy from 4D PC-MRI data. This is mainly because morphological details about the heart valves or the thickness of the myocard usually do not appear in 4D PC-MRI data.

In conclusion, these images depict the main blood flow behavior for different time steps during the cardiac cycle in the anatomical context. One can easily recognize the main flow paths, the converging and diverging of different paths, their velocities and the occurrence of vortices.

In a similar way, we proceed with the visualization of flow features from 4D PC-MRI data. The main paths of the blood flow to be depicted are extracted and rendered as three-dimensional arrows that diverge and converge like the represented flow. Further, we want to exploit the advantage that we are not restricted to a static two-dimensional drawing and show the vortical regions not as two-dimensional hatchings, but as a three-dimensional, hatched structure. Whereas the illustrations described above mainly give an overview of the complete blood flow in a larger area, we also depict more detailed aspects. With the line predicate approach (see Chap. 5), we have the possibility to extract finer flow structures. Thus, we also want to provide an easy visualization of these subflows and their more detailed features.

6.2.2 Finding Representatives

To depict the main features of a line bundle, we reduce the lines to those representing the bundle best. One could apply clustering methods for this undertaking and use the cluster centroids as representing lines. We decided against this option due to two problems. First, the cluster number is not known in advance as it depends mainly on the line bundle's complexity. Second, it is not trivial to consider prior information with cluster methods, if, e.g., certain flow features should be covered by the representatives.

Another possibility to select representatives is the minimal configuration ap-

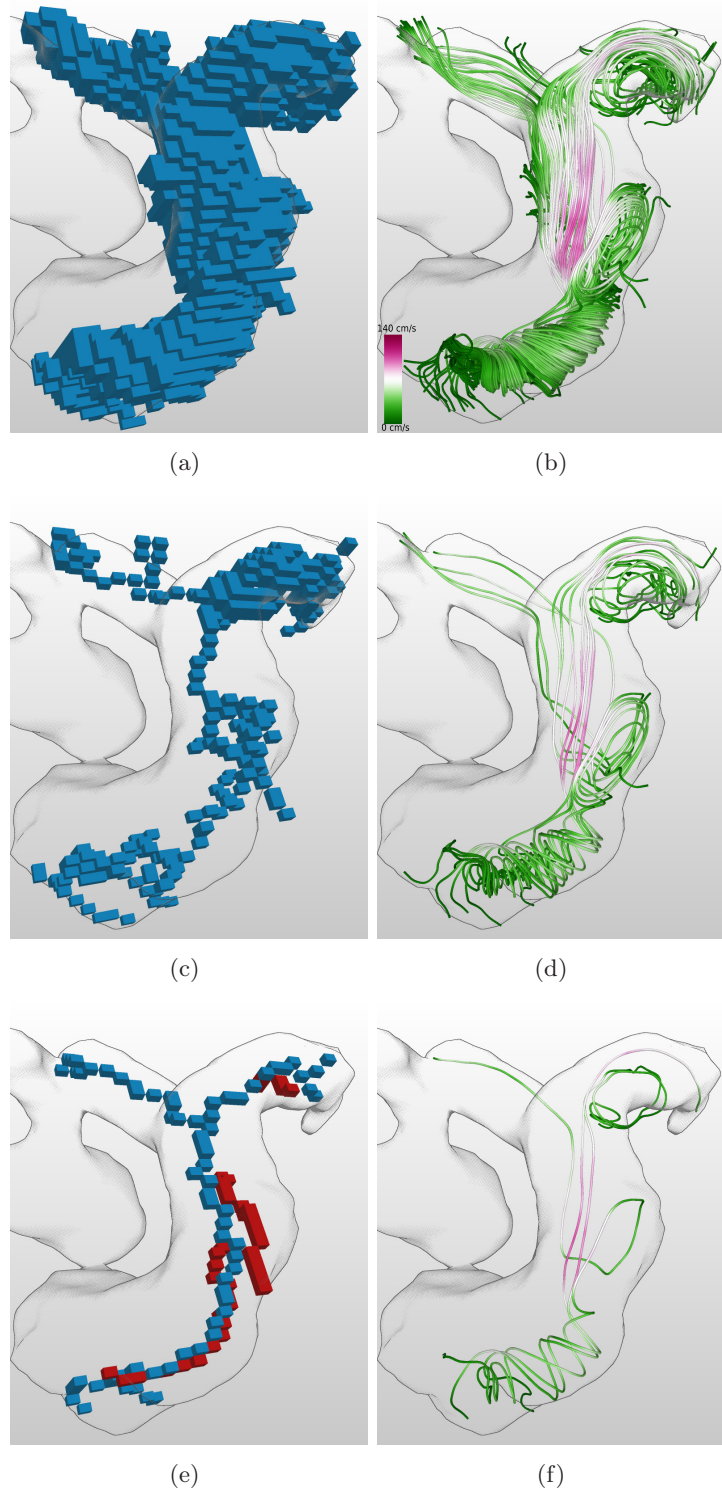


Figure 6.3: Finding representatives of a line bundle. (b) The line bundle to be represented and (a) the voxel representation of its characteristic set. (c) Topologically correct skeleton and (d) the representatives determined based on it (already neglecting lines with a low impact). (e) Skeleton of the smoothed characteristic set and added respective vortex core lines (red). (f) Three representing lines. All images © 2013 IEEE.

proach introduced previously by Salzbrunn [146]. This method has the purpose of finding a minimal number of lines representing all features of a line bundle. For that, a topologically correct skeleton of the characteristic set is generated with the topology-preserving thinning approach by Pal gyi and Kuba [124]. Then, every voxel of the characteristic set is assigned to its closest skeleton voxel. A minimal configuration is the minimal set of lines crossing all skeleton voxels (a skeleton voxel is also considered as crossed when one of its assigned neighboring voxels is visited). This means that multiple crossings of skeleton voxels should be reduced. A deterministic approach to determine this line set is to determine the number of skeleton crossings for the lines. Then, lines with the highest number of skeleton crossings are iteratively selected from the bundle until all skeleton voxels have been visited by at least one representative.

This approach makes sure that all features of the bundle are represented, but for complex bundles the resulting number of lines is still high. Our goal, however, is to find an abstract flow representation and to represent only the main features of a bundle and neglect less significant parts, e.g., behavior shown by only single lines. With that, we accept that not every single aspect is depicted (as this is the point of abstraction). However, due to the big advantage that no user control is needed, we use this minimal configuration method, but adapt it in several ways to further reduce the number of representatives.

The skeleton used for the minimal configuration method is topologically correct. Thus, for complex characteristic sets with, e.g., a bumpy surface due to single lines sticking out, the skeleton can become quite complex. However, for our application we are not interested in the behavior of single lines and do not want thin parts of the bundle (i.e. where only few stream lines run through) or line bundle parts separated by a small gap, to be considered with own skeleton branches. Thus, we simplify the skeleton and accept that it is not topologically correct anymore.

Instead of applying a skeleton postprocessing (e.g., pruning), we try to avoid that the skeleton becomes complex in the first place. To achieve this, we apply a smoothing operation on the characteristic set prior to skeletonization. We use a discrete Gaussian filter (variance = $1.5 \cdot \Delta d$, with voxel size Δd) and apply a thresholding on the result (threshold value = 0.25). This removes single lines and closes small gaps, but preserves the overall shape of the characteristic set. Figure 6.3 shows the difference between an example skeleton generated with and without prior smoothing.

We reduce the number of representatives further by neglecting lines carrying only very few additional information, i.e., when they have a low crossing number with the skeleton voxels. We achieved good results by accepting only lines running through at least 5% of the number of skeleton voxels. The skeleton simplification and the neglect of lines with a low feature information, implies the risk that features are lost and not represented anymore. Although we accept this to a certain extent, we still want to make sure that vortical regions are always represented - as they are of specific importance for understanding the flow. Thus, we add detected vortex core lines (see Sec. 6.2.4) to the skeleton mask again (see Fig. 6.3e,f).

Altogether, these measures result in a reduction of representatives while at the same time, the most important flow features are considered.

6.2.3 Illustrative Rendering of Representatives

To illustrate the flow direction, we depict the selected lines as three-dimensional stream tapes with an arrow head. The stream lines are provided as line strips with velocity and line parameter values for every line point. To create the arrows, we use this information to derive the normal \mathbf{n}_i and binormal \mathbf{b}_i for every line point p_i (Frenet frame). However, in regions with zero line curvature the Frenet frame is undefined, which leads to artifacts in the tapes. We remove these artifacts by averaging the normals and binormals of subsequent line segments. The arrow is then constructed by using the binormal to create a tape-like structure and the normal for the three-dimensionality (arrow height).

With the previously explained minimal configuration method, a line bundle is represented by usually 1-5 lines – depending on the bundle’s complexity. Here, it cannot be avoided that parts of different lines represent the same part of the flow. With respect to cardiac blood flow, this happens, e.g., when two lines emerge from the right ventricle and diverge into the left or right pulmonary artery (see Fig. 6.6b). Thus, the separate display of several lines in areas where they depict the same blood flow, introduces redundancy. Therefore, we fuse stream tapes in these areas, i.e., when they are close to each other and run almost parallel.

To be more precise, we process each pair of representatives and determine for all line points the closest point on the respective other line. Further, we calculate the dot product of the corresponding line segments as a measure of parallelism. Line segment pairs below a certain distance threshold and above a certain parallelism threshold will be fused later on. For the parallelism threshold, we use a value of 0.8. The distance threshold is in the range of 4 – 10mm, but depends on the current representatives.

When fusing stream tape parts, we do not merge them (in the sense that they coincide) but let them run parallel as this is more intuitive (see, e.g., Fig. 6.6b). First, it is easier to observe bifurcations and to distinguish between single stream tapes and fused stream tapes. Second, the representation of fused parts with a larger width complies with what one expects from a merged blood flow, i.e., the adding up of blood volume.

When fusing stream tapes, their course and orientation needs to be adjusted in the fused parts. Here, the main challenge is to create branching points with a smooth transition from the fused to the unfused part of a stream tape (i.e., without wrinkles or turns).

We solve this problem by averaging the course (line strips) and orientation (normals and binormals) of the stream tapes to be fused. When, e.g., merging two stream tapes (see Fig. 6.4a), we process all pairs of points that need to be fused and determine the respective midpoint c and the average normal \mathbf{n} and binormal \mathbf{b} . The new tape segments are then aligned with \mathbf{b} running through c . The new stream line

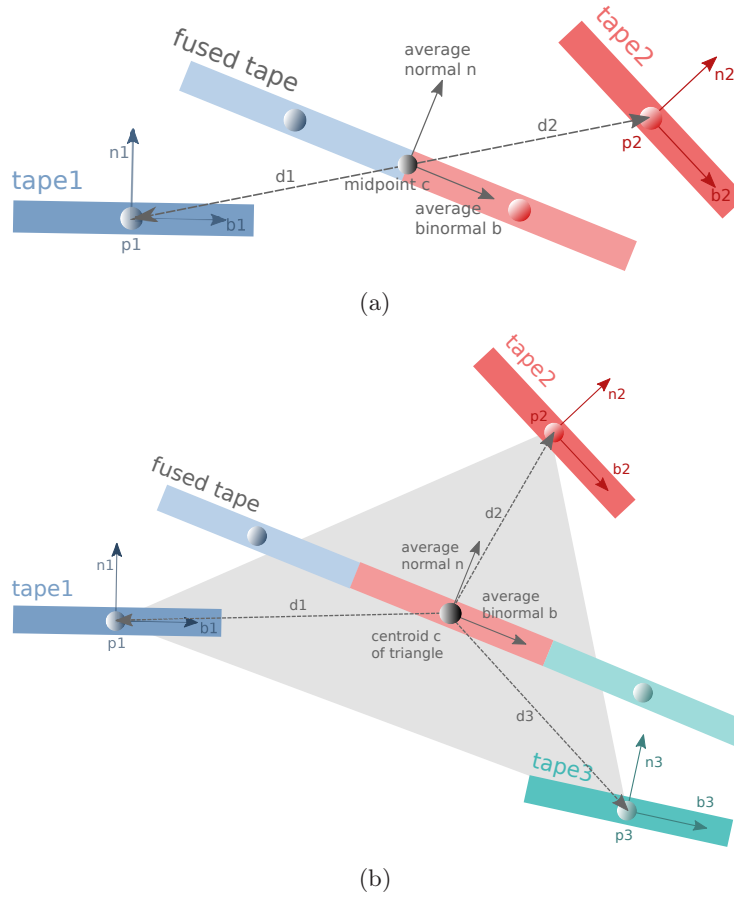


Figure 6.4: Fusion of stream tape segments. (a) Situation for two stream tapes. Tapes are arranged next to each other along the averaged binormal. (b) The ordering of three stream tapes depends on the dot products between \mathbf{d}_i and the average binormal.

points are shifted from the midpoint in or against the direction of \mathbf{b} to achieve a parallel tape arrangement. The shifting direction is determined by calculating the dot product between the vector \mathbf{d}_i (pointing from the midpoint c to the original line point p_i) and the average binormal \mathbf{b} . The dot product specifies the length of the projection of \mathbf{d}_i onto the line along the binormal vector. If this value is positive, the shifting is carried out in positive direction of \mathbf{b} . Otherwise, the negative direction is used.

For more than two tapes, the general procedure is the same (see Fig. 6.4b). The centroid c of the triangle the three points span, the average normal \mathbf{n} and binormal \mathbf{b} are calculated. Again, the fused tape is oriented along \mathbf{b} and runs through c . The ordering is also determined based on the projections of the vectors \mathbf{d}_i onto the binormal \mathbf{b} . Here, the dot products are sorted. The stream tape with the middle value is shifted onto c , the others are shifted with or against the binormal direction respectively.

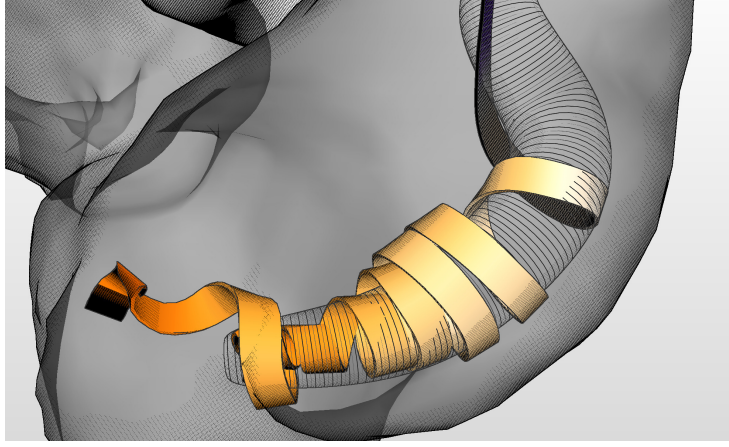


Figure 6.5: Yellow stream tape and a tube-like structure depicting a vortex region. The hatching resembles vortical flow. © 2013 IEEE.

Finally, the line strips are smoothed at the bifurcations with a small kernel size to ensure a nice transition from fused to unfused tapes. The orientation of the arrows does not need to be handled due to the averaging of the binormals. The arrows are color-coded with the respective velocity values. In the fused parts, the average velocity is shown.

6.2.4 Illustrative Rendering of Vortices and Context

Vortices are important aspects for the interpretation of flow behavior and can be a strong indicator for pathological hemodynamics. Therefore, we display these regions in addition to the arrows representing the main flow paths. We identify vortices by applying the λ_2 method as described by Peikert et al. [126] and the vortex core line detection by Sujudi and Haines [167]. For details, we refer to Sections 2.2.4 and 5.3.1. Based on the vortex core line and the known measure of the vortex region, we calculate an ellipsoid-like tube, i.e., a tube with a radius decreasing to its ends. To be more precise, a vortex core line represented as a line strip with n points (with equal distances) has at point i a radius r of

$$r = \sin(\arccos(2 \cdot \frac{i - \frac{n}{2}}{n})) \quad (6.1)$$

Our method is related to the vortex glyphs by Reinders et al. [137]. However, we do not use ellipsoids because they usually simplify the vortex regions too much. Further, we give the vortex structures an appearance resembling rotational flow by applying a hatch texture. For that, we use the approach presented by Hummel et al. for the depiction of flow direction on integral surfaces [63]. By increasing the first texture coordinate along the vortex core line and the second around the tube, we can easily adapt this method to our needs. Figure 6.5 shows an example of a hatched vortex structure.

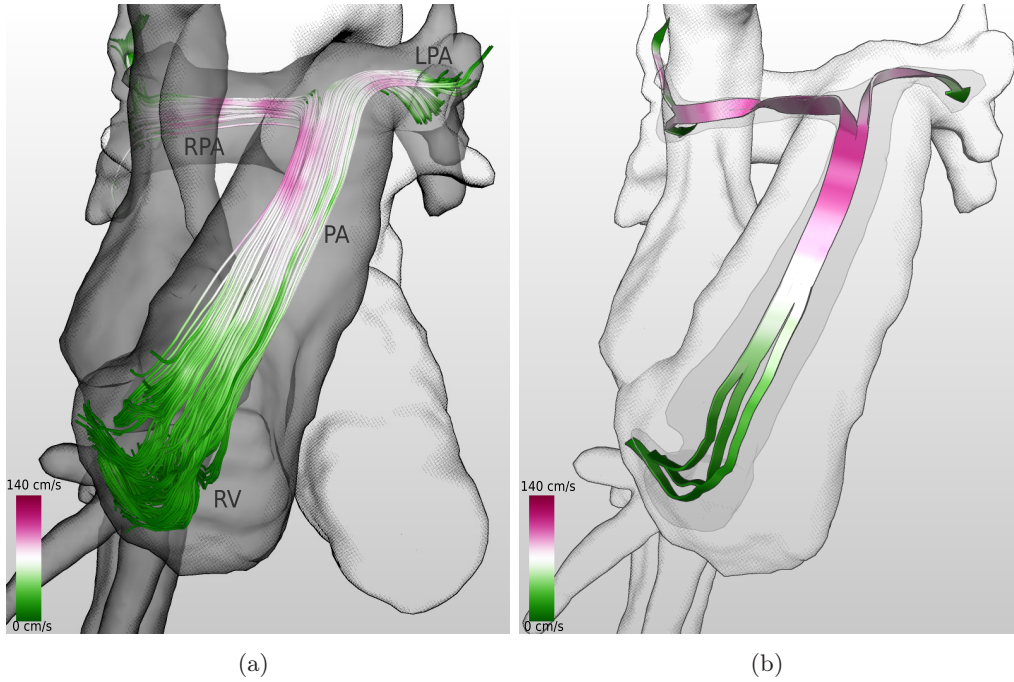


Figure 6.6: (a) Systolic blood flow in the right ventricular outflow tract (i.e., right ventricle (RV), pulmonary trunk (PA), right (RPA) and left pulmonary arteries (LPA)) of a healthy heart. The fastest blood (maximal velocity $\geq 80 \frac{cm}{s}$) flows from the ventricle into the left and right pulmonary artery. (b) Our illustrative visualization depicts this balanced behavior comprehensively with a diverging arrow. Both images © 2013 IEEE.

For the anatomical context, we use outlines and a unobtrusive halftoning technique to communicate shape. In order to depict not only the representatives of the line bundles, but also give a hint on the original bundle’s measures, we render a light transparent model of the characteristic set and augment it with silhouettes (see also Sec. 5.3.4).

6.3 Results

In this section, we present several blood flow features visualized illustratively with our method, discuss advantages and limitations, and compare our abstract visualization with the original line bundle depiction. These examples are derived from three 4D PC-MRI datasets of the heart and aorta that have already been presented in the last chapter. The datasets are the healthy human heart scan from Section 5.4.2.1, the Fallot tetralogy dataset from Section 5.4.2.2, and the scan showing an aorta with an aneurysm in the descending part (see Sec. 5.4.1.3). The first examples show the blood ejection from the right ventricle into the pulmonary arteries (lungs) of a healthy and a pathological heart. Fig. 6.6a shows the flow situation within the

healthy heart. The blood is ejected without vortices and is evenly distributed into the left and right pulmonary arteries. The abstract depiction of this flow comprises three representatives (see Fig. 6.6b). These run parallel at first (and are, therefore, shown as a fused arrow) and diverge into the left and right pulmonary arteries later on. Due to the morphological alterations, this phase shows a different flow behavior in the pathological heart. In Fig. 6.7a, it can be noted that the main path (i.e., the fast flowing blood) is now directed mainly at the left pulmonary artery. The corresponding illustrative visualization depicts this behavior with three lines. One represents the smaller part of the blood flowing into the right pulmonary artery. The two others depict the straight and the vortical flow into the left pulmonary artery respectively. The small superimposed image in Fig. 6.7b shows that in this case, the different streamtapes cannot be consistently arranged such that no overlaps occur at the bifurcations.

One reason for the imbalanced flow in the pathological heart are vortices in the bulges of the pulmonary trunk. Fig. 6.7c depicts the flow into the right pulmonary artery and these vortices. Due to the many lines, details are hard to see in this image. In the reduced version of this line bundle (see Fig. 6.7d) it becomes apparent that the flow structure is divided in blood flowing directly into the right PA and blood taking the detour through the vortices.

The next examples show flow in the left atrium of the healthy heart at two different points during the cardiac cycle. Fig. 6.8a depicts the ejection of blood from the left atrium into the left ventricle. This flow feature is represented in Fig. 6.8b with three representatives, each showing one source of the ejected blood (pulmonary veins). The arrows are fused in the areas of parallel flow. Fig. 6.8c shows the filling of the left atrium when the valve to the left ventricle is closed. Here, a large vortex develops in the center of the atrium that is represented by a single stream line in Fig. 6.8d. In this case, the skeletonization shows unsatisfactory results because of the sphere-like shape of the characteristic set. To solve this problem, we used only the vortex core line as skeleton, which produced the displayed representative.

The final example is the complex flow in the aortic aneurysm dataset. Figure 6.9a shows the stream line influenced by the descending vortex. A high number of lines is depicted and although one can see the overall behavior, the details are mostly occluded. In Figure 6.9b, the abstract version of this flow is displayed. Here, it takes only three arrows and three vortex tubes, to communicate the same information in an easier way.

Concerning the runtime, the rendering of the arrows as well as the fusion runs interactively. The skeletonization is the main bottleneck when determining representatives. Depending on the size of the original line bundle, this takes 1 – 7s.

Expert Feedback We presented the previous results to two radiologists (one with, one without expertise in 4D PC-MRI) and one medical physicist. Here, we summarize their feedback concerning usefulness, comprehensibility, and limitations. After showing the experts the abstract flow depictions without further explanations, they

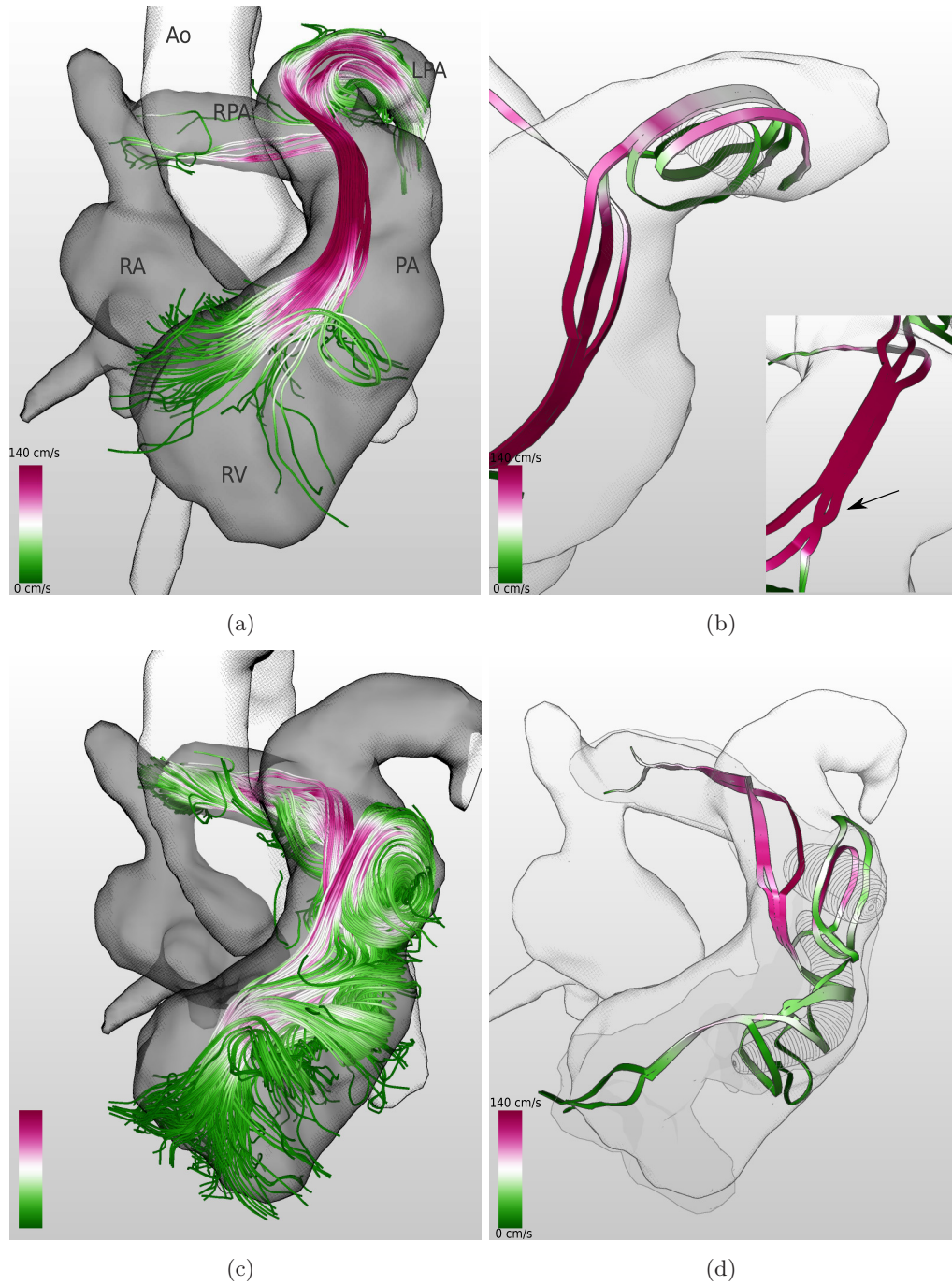


Figure 6.7: Blood flow in the right heart of the pathological TOF dataset. (a) The blood ejected from the right ventricle flows into the left pulmonary artery. (b) The representatives show nicely the flow situation in the left pulmonary artery. A different viewing direction is chosen for a better overview. The subfigure shows that the tape ordering is not consistent in this case (arrow). (c) Blood flow towards the right pulmonary artery shows vortical behavior in the bulges of the pulmonary trunk. (d) The abstract depiction shows the split behavior of this flow feature. All images© 2013 IEEE. Abbreviations: RV, right ventricle; RA, right atrium; PA, pulmonary artery; LPA, left PA; RPA, right PA; Ao, aorta.

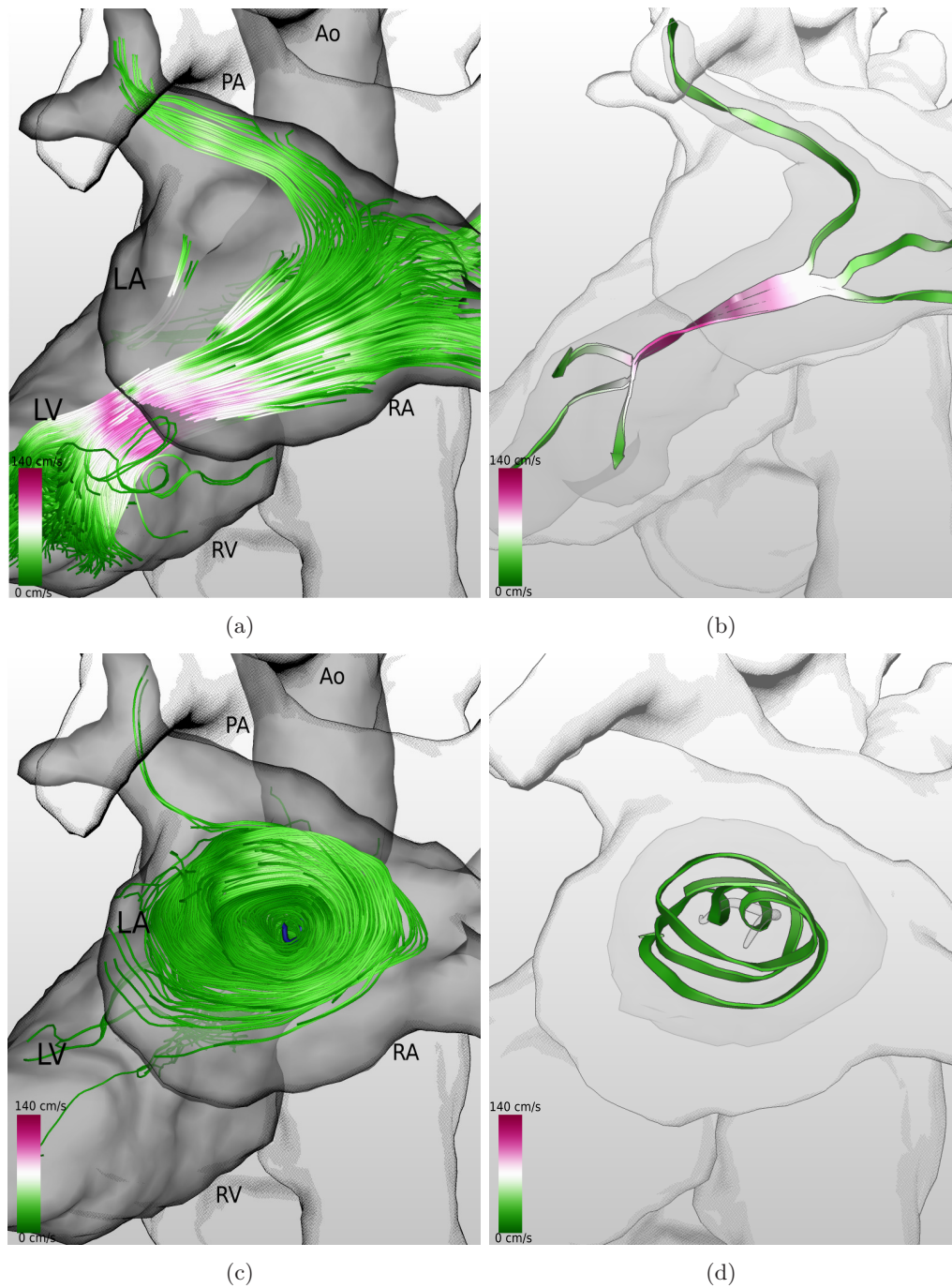


Figure 6.8: Blood flow in the left atrium of a healthy heart during systole and diastole. (a) Original line bundle showing blood flowing from left atrium to left ventricle with a high velocity. (b) This diastolic flow behavior is represented by three stream lines depicting the different sources of the blood. (c) With closed valves, a large centered vortex evolves in the left atrium during systole. (d) The vortical flow is represented by a single stream line. All images © 2013 IEEE. Abbreviations: RV, right ventricle; RA, right atrium; PA, pulmonary artery; LV, left ventricle; LA, left atrium; Ao, aorta.

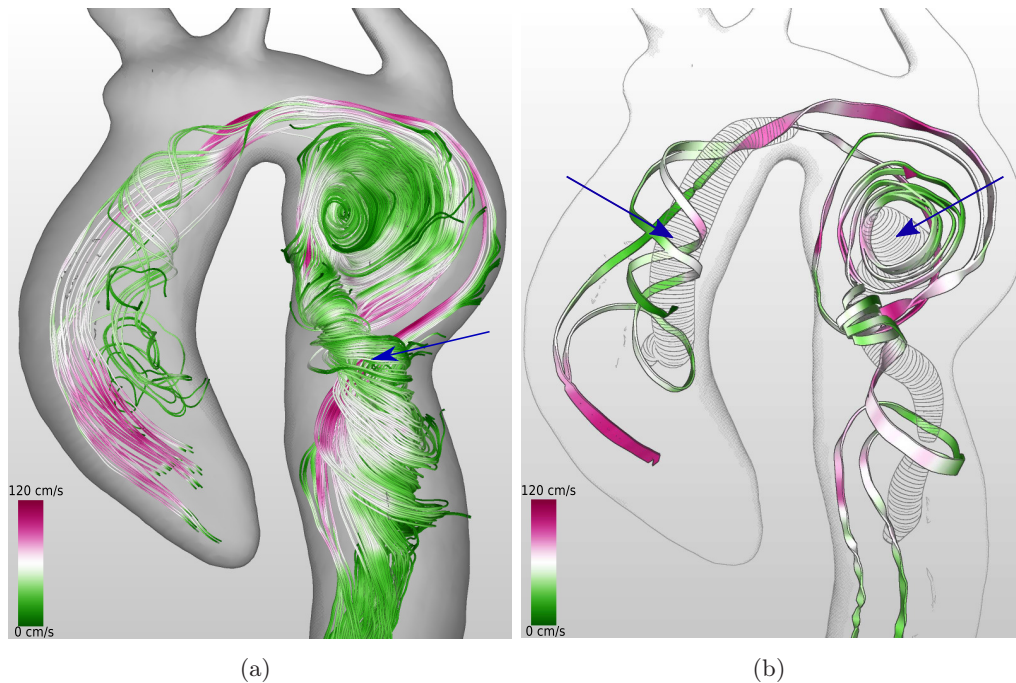


Figure 6.9: Blood flow in an aorta with a large aneurysm in the proximal descending part. (a) The line bundle shows the flow through the vortex in the descending aorta (arrow). (b) The representatives of the previous line bundle show clearly how two other vortices are involved in this flow as well (arrows). Both images © 2013 IEEE.

rated them as easy to understand and intuitive. Especially, the radiologist without experience in flow analysis confirmed the comprehensibility of the arrow images and described them as "clearly arranged". He also noted that the vortex tubes helped him understand the spatial relations of the data. One expert, however, criticized that in some visualizations, the flow direction is sometimes hard to recognize in all cases (e.g., in Fig. 6.7d). More arrow heads or a color-coding depicting the flow direction could solve this problem. Also, an expert suggested to assign further flow information, such as blood volume or velocity, to the width of the arrows. In comparison to the line bundle images, the clear arrangement and the simplification were perceived as main advantage of the abstract visualizations. Due to that, the neglectation of some line bundle details were approved and not considered as disadvantage.

A benefit expected from our method, was the support of a medical expert in comparing different flow measurements of one patient (e.g., pre- and postoperative data) or of a patient with a healthy dataset to detect and observe flow anomalies. Altogether, the experts described the illustrative visualizations as very suitable to give a quick and comprehensible overview over the flow situation in a specific dataset and to complement other analysis techniques focussing on regional flow information or specific flow parameters (e.g., wall shear stress).

6.4 Discussion and Future Work

In this chapter, we addressed the illustrative visualization of flow features in 4D PC-MRI data with the goal to facilitate flow interpretation. In doing so, we aimed primarily at an abstract flow depiction as it is used in hand-drawn illustrations. The input for our method are flow features generated with line predicate analysis. From these line bundles, we identified a minimal set of lines representing the bundle's main flow characteristics. For that, we adapted the minimal configuration approach by Salzbrunn [146] such that insignificant features are ignored, but the important information (e.g., about vortices) is represented. The representatives are rendered as three-dimensional arrows that are easy to understand. To reduce redundant information, these arrows are fused if they represent the same part of the flow.

The morphological structures are denoted by silhouettes and a light halftoning technique. Further context is provided by rendering vortex regions as hatched, tube-like structures and by displaying the outline of the original line bundle's characteristic set.

We presented several flow features of cardiac and aortic blood flow that we visualized with our technique. With these examples we could show that an abstract visualization allows for a clearer and less cluttered display of flow structures. Since the depiction was reduced to the essential features, it also simplifies the comparison of blood flow in different datasets, e.g., a pathological with a healthy dataset. Further, one could take these arrow-like renderings as a tool to communicate the findings and add further information about the data, e.g. by varying the arrow width with the represented blood volume or other derived quantitative parameters. Therefore, we believe that our method offers many possibilities for further applications in comparative visualization and beyond the medical domain. In summary, this abstract flow visualization helps during data analysis, but also supports the visual explanation of the findings to colleagues or in publications.

Despite of its advantages, our technique has some limitations. The skeletonization used for finding representatives of line bundles, shows non-satisfying result when the respective characteristic set has a sphere-like shape (see Fig. 6.8d). Further, when fusing several arrows it is not always possible to arrange them such that no distracting overlaps occur at the bifurcations (see Fig. 6.7b). Also, the fusion depends on a parameter and can therefore not be accomplished automatically yet. Another esthetical drawback are wrinkles and turns in the stream tapes, which make these images appear not as perfect as comparable illustrations. But after all, we are still working with measured and sometimes noisy 4D PC-MRI data. To improve the appearance, one might think of more sophisticated smoothing techniques or even about not using representative streamlines, but creating new representing structures based on the flow features from scratch.

Apart from this, there are many more starting points for future work. As already stated, our method is suitable as basis for comparative visualization techniques and for an augmentation with further derived flow information. One could investigate in these directions and, e.g., create standardized visualizations facilitating the inter-

pretation of blood flow and the comprehension of differences between healthy and pathological data or patient data of different points in time. Further, we address only steady blood flow so far and plan to enhance our method to unsteady flow. Also, the anatomical rendering as shown here, has room for improvement. However, the limited morphological information in the 4D PC-MRI data restricts the depiction of further details. One might investigate whether it is feasible to register morphological MRI or CT data from the same patient with this flow depiction and create fancier anatomical renderings with them.

Conclusion and Outlook

The central theme of this thesis was the use of illustrative techniques to give people more insight into their flow data. More specifically, we have attended to two main problems. The first problem was the reduced visual clarity of complex stream surfaces. Stream surfaces are per se very intuitive visualization tools, but suffer from self-occlusion and clutter when used with complex flow. We found the inspiration for our method from hand-drawn illustrations of flow phenomena. We applied the style of these illustrations to our stream surface meshes in order to achieve the same clarity. But apart from the use of silhouettes, halftoning, and illustrative surface stream lines, we also exploited the advantage of computational flow visualization over static textbook images. We utilized modern computer graphics to provide users with interactive exploration methods. These allow the examination of flow paths on the surface and the view inside a tightly wrapped-up surface part. An expert survey confirmed the value of our method and the application of similar methods to time-dependent integral surfaces appears as an interesting research field.

The second problem we attended to was the analysis of 4D PC-MRI blood flow data. In this field, there is a lack of analysis tools, which allow not only flow experts to understand the hemodynamic situation. We designed a visual analysis method, providing a completely new view on the blood flow in the aorta and the heart. Our suggested method is meant to complement existing methods, which focus on local flow parameters and lack support with flow interpretation. So, we provide users with a toolbox allowing them to query the data according to properties they are interested in. As a result, stream line or path line bundles represent the requested flow features, such as high-velocity jets or vortical flow. In order to further enhance the comprehensibility of blood flow analysis, we depict our results abstractly with easy-to-understand arrows and vortex tubes, which are also used in traditional flow illustrations. This simplified visualization makes the flow information easy to grasp even for an audience without a respective fluid dynamics knowledge.

Altogether, the presented blood flow analysis techniques have the potential to be of great value in the medical research and clinical routine. The line predicate method allows for a simple overview of the hemodynamics situation whereas the abstract depiction takes care of comprehensibility. The advantage of this analysis is the global view on the data, which helps to quickly check a dataset for flow anomalies and to understand the main characteristics of the particular flow. Our method may serve as a complement to the current tools' quantitative measures by helping to interpret these measures. For example, even after surgical repair of a Fallot tetralogy, our medical collaborators previously observed very differing flow velocities in the left

and right pulmonary arteries. However, not until we applied our method to this data, they could see and understand how these velocity differences arose. Also, it may facilitate the side-by-side comparison of several flow datasets. The latter is a frequent task in the clinics when comparing healthy and pathological cases, pre- and postsurgical measurements, or when monitoring disease development over time. In addition to this, it is thinkable to enhance our method towards a standardized analysis tool with predefined queries. This would enhance the reproducibility and decrease the inter-observer variability when analyzing large amounts of data as they occur in the clinical routine or in larger medical studies.

In summary, we presented visualization techniques, which allow a new perspective on the data, were positively evaluated, provide many starting points for future work (see also Sec. 3.5, 5.5, 6.4), and have already triggered other researchers to pick up ideas.

List of Figures

1.1	Examples of non-computational flow visualizations	2
2.1	Different types of feature curves	11
2.2	Different types of periodic orbits	12
2.3	Parameterization of a stream surface	13
2.4	Different types of critical points in 2D	14
2.5	Different types of critical points in 3D	15
2.6	Examples for texture-based and geometric flow visualizations	16
2.7	Examples of visualizations of time-dependent flow features.	22
2.8	Examples of illustrative methods for perceptual enhancement	24
2.9	Examples of illustrative methods for visibility management	25
2.10	Ex. of illustr. methods for focus emphasis and visual explanation	26
3.1	Examples for hand-drawn flow illustrations	32
3.2	Related work for illustrative stream surface rendering	33
3.3	Rendering steps for illustrative stream surfaces	39
3.4	Different silhouette layers	40
3.5	Construction of illustrative surface stream lines	43
3.6	Special cases of illustrative surface stream line construction	44
3.7	Parameter-based and geometry-based slabs	45
3.8	Overview of the delta wing dataset	46
3.9	Illustrative vortex breakdown bubble	47
3.10	Illustrative stream surfaces of the delta wing dataset	48
3.11	Comparison of illustrative stream surface with original illustration.	49
3.12	Illustrative stream surfaces of the periodic orbit dataset	51
3.13	Overview of the combustion chamber dataset	52
3.14	Illustrative stream surfaces of the combustion chamber dataset	52
4.1	Anatomy of heart and aorta	58
4.2	Illustration of the heart cycle	59
4.3	Magnetic resonance by means of a compass needle	63
4.4	Equilibrium of protons in a strong magnetic field	65
4.5	Effect of an RF pulse in a stationary or rotation reference frame	66
4.6	T_1 - and T_2 -weighted MRI scans of a human brain	67
4.7	Spin echo sequence	68
4.8	Spatial encoding within a slice	69
4.9	Bipolar gradients for velocity encoding	72
4.10	v_{enc} and phase wrapping	74
4.11	4D PC-MRI scan of a healthy heart	75
5.1	Example of current 4D PC-MRI analysis methods	78

5.2	Related work of blood flow visualization.	80
5.3	Workflow of line predicate method	85
5.4	Vortex detection (λ_2 , Sujudi-Haimes) in a healthy heart	89
5.5	Analysis results of the healthy aorta dataset	96
5.6	Analysis results of the distorted aorta dataset	97
5.7	Temporal evolution of a vortex in the distorted aorta dataset	98
5.8	Analysis results of the aorta aneurysm dataset	100
5.9	Analysis results showing systolic blood flow in the right heart	103
5.10	Temporal evolution of blood flow in the right heart	104
5.11	Velocity line predicate result of the healthy heart	105
5.12	Residence time predicate in the healthy heart	106
5.13	Blood flow in the left atrium over one heart beat	107
5.14	Systolic blood flow in the TOF dataset	109
5.15	Diastolic blood flow in the TOF dataset	110
6.1	Hand-drawn blood flow illustrations	116
6.2	Related work of illustrative blood flow visualization	118
6.3	Finding representatives of a line bundle	120
6.4	Fusion of stream tape segments	123
6.5	Example of a stream tape and an abstract vortex visualization	124
6.6	Illustrative depiction of systolic flow in the right outflow tract	125
6.7	Abstract visualization of blood flow in the TOF dataset	127
6.8	Abstract visualization of blood flow in a left atrium	128
6.9	Illustrative depiction of vortical flow in aorta aneurysm dataset	129

Bibliography

- [1] ABRAHAM, R., AND SHAW, R. *Dynamics - The Geometry of Behaviour*. Aerial Press, 1984. (Cited on pages 2, 31 and 32.)
- [2] AKIBA, H., WANG, C., AND MA, K.-L. AniViz : A Template-Based Animation Tool for Volume Visualization. *IEEE Comput Graph App* 30, 5 (2010), 61–71. (Cited on page 27.)
- [3] ANGELELLI, P., AND HAUSER, H. Straightening tubular flow for side-by-side visualization. *IEEE TVCG* 17, 12 (2011), 2063–2070. (Cited on pages 80 and 81.)
- [4] APPEL, A., ROHLF, F. J., AND STEIN, A. J. The Haloed Line Effect for Hidden Line Elimination. In *Proc. of ACM SIGGRAPH* (1979), pp. 151–157. (Cited on page 34.)
- [5] ASIMOV, D. Notes on the Topology of Vector Fields and Flows. In *Proc. of IEEE Visualization* (1993), pp. 1–21. (Cited on pages 7, 8, 9, 13 and 14.)
- [6] AUER, C., KASTEN, J., KRATZ, A., ZHANG, E., AND HOTZ, I. Automatic, Tensor-Guided Illustrative Vector Field Visualization. In *Proc. of IEEE Pacific Visualization* (2013), pp. 265–275. (Cited on page 23.)
- [7] BANKS, D. C., AND SINGER, B. A. Vortex Tubes in Turbulent Flows: Identification, Representation, Reconstruction. In *Proc. of IEEE Visualization* (1994), pp. 132–139. (Cited on page 20.)
- [8] BANKS, D. C., AND SINGER, B. A. A Predictor-Corrector Technique for Visualizing Unsteady Flow. *IEEE TVCG* 1, 2 (1995), 151–163. (Cited on pages 20 and 21.)
- [9] BARKER, A., BOCK, J., LORENZ, R., AND MARKL, M. 4D Flow MR Imaging. *Am J Neuroradiol* 18 (2010), 46–52. (Cited on pages 73, 77 and 80.)
- [10] BARKER, A. J., MARKL, M., BÜRK, J., LORENZ, R., BOCK, J., BAUER, S., SCHULZ-MENGER, J., AND VON KNOBELSDORFF-BRENKENHOFF, F. Bicuspid Aortic Valve is Associated with Altered Wall Shear Stress in the Ascending Aorta. *Circulation* 5, 4 (July 2012), 457–66. (Cited on page 61.)
- [11] BAUER, D., AND PEIKERT, R. Vortex Tracking in Scale-Space. *Computing* (2002), 233–240. (Cited on page 21.)
- [12] BOCK, J., FRYDRYCHOWICZ, A., STALDER, A. F., BLEY, T. A., BURKHARDT, H., HENNIG, J., AND MARKL, M. 4D Phase Contrast MRI at 3T: Effect of Standard and Blood-Pool Contrast Agents on SNR, PC-MRA,

- and Blood Flow Visualization. *Magn Reson Med* 63, 2 (2010), 330–338. (Cited on page 95.)
- [13] BOGREN, H. G., AND BUONOCORE, M. H. 4D Magnetic Resonance Velocity Mapping of Blood Flow Patterns in the Aorta in Young vs. Elderly Normal Subjects. *J Magn Reson* 10, 5 (1999), 861–869. (Cited on pages 80 and 96.)
- [14] BORN, S., MARKL, M., GUTBERLET, M., AND SCHEUERMANN, G. Illustrative Visualization of Cardiac and Aortic Blood Flow from 4D MRI Data. In *Proc. of IEEE Pacific Visualization* (2013), pp. 129–136. (Cited on page 117.)
- [15] BORN, S., PFEIFLE, M., MARKL, M., GUTBERLET, M., AND SCHEUERMANN, G. Visual Analysis of Cardiac 4D MRI Blood Flow Using Line Predicates. *IEEE TVCG* 19, 6 (June 2013), 900–912. (Cited on pages 79, 80 and 114.)
- [16] BORN, S., PFEIFLE, M., MARKL, M., AND SCHEUERMANN, G. Visual 4D MRI Blood Flow Analysis with Line Predicates. In *Proc. of IEEE Pacific Visualization* (2012), pp. 105 – 112. (Cited on pages 79, 80 and 114.)
- [17] BORN, S., WIEBEL, A., FRIEDRICH, J., SCHEUERMANN, G., AND BARTZ, D. Illustrative Stream Surfaces. *IEEE TVCG* 16, 6 (2010), 1329–1338. (Cited on page 32.)
- [18] BRAMBILLA, A., CARNECKY, R., PEIKERT, R., VIOLA, I., AND HAUSER, H. Illustrative Flow Visualization : State of the Art, Trends and Challenges. *STAR Proceedings of Eurographics* (2012). (Cited on pages 27 and 117.)
- [19] BRUCKNER, S., GRIMM, S., KANITSAR, A., AND GRÖLLER, M. E. Illustrative Context-Preserving Exploration of Volume Data. *IEEE TVCG* 12, 6 (2006), 1559–1569. (Cited on page 34.)
- [20] BUONOCORE, M. H. Visualizing Blood Flow Patterns using Streamlines, Arrows, and Particle Paths. *Magn Reson Med* 40, 2 (1998), 210–226. (Cited on page 80.)
- [21] CABRAL, B., AND LEEDOM, L. Imaging Vector Fields Using Line Integral Convolution. In *Proc. of ACM SIGGRAPH* (1993), pp. 264–272. (Cited on page 17.)
- [22] CARNECKY, R., FUCHS, R., MEHL, S., JANG, Y., AND PEIKERT, R. Smart Transparency for Illustrative Visualization of Complex Flow Surfaces. *IEEE TVCG* 19, 5 (May 2013), 838–51. (Cited on pages 18, 33 and 34.)
- [23] CHANRAUD, S., ZAHR, N., SULLIVAN, E. V., AND PFEFFERBAUM, A. MR Diffusion Tensor Imaging: A Window into White Matter Integrity of the Working Brain. *Neuropsychol Rev* 20, 2 (2010), 209–225. (Cited on page 62.)

- [24] CHEN, C.-K., YAN, S., YU, H., MAX, N., AND MA, K.-L. An Illustrative Visualization Framework for 3D Vector Fields. *Comput Graph Forum* 30, 7 (2011). (Cited on pages 25 and 117.)
- [25] CHEN, Y., COHEN, J. D., AND KROLIK, J. H. Similarity-Guided Streamline Placement With Error Evaluation. *IEEE TVCG* 13, 6 (2007), 1448–1455. (Cited on page 18.)
- [26] COFFEY, D., MALBRAATEN, N., LE, T. B., BORAZJANI, I., SOTIROPOULOS, F., ERDMAN, A. G., AND KEEFE, D. F. Interactive Slice WIM : Navigating and Interrogating Volume Data Sets Using a Multisurface, Multitouch VR Interface. *IEEE TVCG* 18, 10 (2012), 1614–1626. (Cited on page 18.)
- [27] CORREA, C., SILVER, D., AND CHEN, M. Illustrative Deformation for Data Exploration. *IEEE TVCG* 13, 6 (2007), 1320–1327. (Cited on page 34.)
- [28] DALLMANN, U. *Topological Structures of Threedimensional Flow Separations*. PhD thesis, Deutsches Zentrum für Luft- und Raumfahrt, 1983. (Cited on pages 31, 32 and 49.)
- [29] DECAUDIN, P. Cartoon-Looking Rendering of 3D-Scenes. Tech. Rep. 2919, INRIA Rocquencourt, June 1996. (Cited on page 38.)
- [30] DIEPSTRATEN, J., WEISKOPF, D., AND ERTL, T. Transparency in Interactive Technical Illustrations. In *Proc. of Eurographics* (2002), vol. 21, pp. 317–325. (Cited on page 34.)
- [31] DOOLEY, D., AND COHEN, M. F. Automatic Illustration of 3D Geometric Models : Lines. In *Proc. of ACM Symposium on Interactive 3D Graphics* (1990), pp. 77–83. (Cited on page 34.)
- [32] DOOLEY, D., AND COHEN, M. F. Automatic Illustration of 3D Geometric Models: Surfaces. In *Proc. of IEEE Visualization* (1990), no. 2, pp. 77–82. (Cited on pages 34 and 36.)
- [33] DORMAND, J. R., AND PRINCE, P. J. A Family of Embedded Runge-Kutta Formulae. *J Comput Appl Math* 6, 1 (1980), 19–26. (Cited on page 86.)
- [34] EDMUNDS, M., LARAMEE, R., MALKI, R., MASTERS, I., CROFT, T., CHEN, G., AND ZHANG, E. Automatic Stream Surface Seeding: A Feature Centered Approach. *Comput Graph Forum* 31, 3 (2012), 1095–1104. (Cited on page 18.)
- [35] ENNKER, J., BAUER, S., AND KONERTZ, W. *Herzchirurgie*, checkliste ed. Georg Thieme Verlag, Stuttgart, 2002. (Cited on pages 60, 61 and 77.)
- [36] ERIKSSON, J., CARLHÄLL, C. J., DYVERFELDT, P., ENGVALL, J., BOLGER, A. F., AND EBBERS, T. Semi-Automatic Quantification of 4D Left Ventricular Blood Flow. *J Cardiovasc Magn R* 12, 9 (2010). (Cited on pages 59, 77 and 81.)

- [37] EVERTS, M. H., BEKKER, H., ROERDINK, J., AND ISENBERG, T. Depth-Dependent Halos: Illustrative Rendering of Dense Line Data. *IEEE TVCG* 15, 6 (2009), 1299–306. (Cited on pages 18 and 23.)
- [38] FERSTL, F., BÜRGER, K., THEISEL, H., AND WESTERMANN, R. Interactive Separating Streak Surfaces. *IEEE TVCG* 16, 6 (2010), 1569–77. (Cited on pages 18 and 26.)
- [39] FISCHER, J., BARTZ, D., AND STRASSER, W. Illustrative Display of Hidden Isosurface Structures. In *Proc. of IEEE Visualization* (2005), pp. 663–670. (Cited on pages 34 and 40.)
- [40] FRYDRYCHOWICZ, A., ARNOLD, R., HIRTLE, D., SCHLENSAK, C., STALDER, A. F., HENNIG, J., LANGER, M., AND MARKL, M. Multidirectional Flow Analysis by Cardiovascular Magnetic Resonance in Aneurysm Development Following Repair of Aortic Coarctation. *J Cardiovasc Magn R* 10, 30 (2008). (Cited on page 99.)
- [41] FUHRMANN, A., AND GRÖLLER, E. Real-Time Techniques For 3D Flow Visualization. In *Proc. of IEEE Visualization* (1998), pp. 305–312. (Cited on page 26.)
- [42] FYRENIUS, A., WIGSTROM, L., EBBERS, T., KARLSSON, M., ENGVALL, J., AND BOLGER, A. Three Dimensional Flow in the Human Left Atrium. *Heart* 86, 4 (2001), 448–455. (Cited on page 90.)
- [43] GARTH, C., KRISHNAN, H., TRICOCHÉ, X., BOBACH, T., AND JOY, K. I. Generation of Accurate Integral Surfaces in Time-Dependent Vector Fields. *IEEE TVCG* 14, 6 (2008), 1404–11. (Cited on pages 17, 18, 33 and 54.)
- [44] GARTH, C., TRICOCHÉ, X., SALZBRUNN, T., BOBACH, T., AND SCHEUERMANN, G. Surface Techniques for Vortex Visualization. *Proc. of VisSym* (2004), 155–164. (Cited on page 17.)
- [45] GARTH, C., TRICOCHÉ, X., AND SCHEUERMANN, G. Tracking of Vector Field Singularities in Unstructured 3D Time-dependent Datasets. In *Proc. of IEEE Visualization* (2004), pp. 329–336. (Cited on page 21.)
- [46] GASTEIGER, R., LEHMANN, D. J., VAN PELT, R., JANIGA, G., BEUING, O., VILANOVA, A., THEISEL, H., AND PREIM, B. Automatic Detection and Visualization of Qualitative Hemodynamic Characteristics in Cerebral Aneurysms. *IEEE TVCG* 18, 12 (Dec. 2012), 2178–2187. (Cited on page 118.)
- [47] GASTEIGER, R., NEUGEBAUER, M., KUBISCH, C., AND PREIM, B. Adapted Surface Visualization Of Cerebral Aneurysms With Embedded Blood Flow Information. *Proc. of Eurographics Workshop on Visual Computing for Biology and Medicine (VCBM)* (2010), 25–32. (Cited on pages 81 and 117.)

- [48] GEIGER, J., MARKL, M., JUNG, B., GROHMANN, J., STILLER, B., LANGER, M., AND ARNOLD, R. 4D-MR Flow Analysis in Patients after Repair for Tetralogy of Fallot. *Eur Radiol* 21, 8 (Aug. 2011), 1651–1657. (Cited on page 108.)
- [49] GLASER, K. J., MANDUCA, A., AND RICHARD L EHMAN. Review of MR Elastography Applications and Recent Developments. *J Magn Reson Imaging* 36, 4 (2012), 757–774. (Cited on page 62.)
- [50] GOOCH, B., AND GOOCH, A. *Non-Photorealistic Rendering*. A K Peters, Natick, MA, 2001. (Cited on pages 23 and 38.)
- [51] GROTHOFF, M., SPORS, B., ABDUL-KHALIQ, H., AND GUTBERLET, M. Evaluation of Postoperative Pulmonary Regurgitation after Surgical Repair of Tetralogy of Fallot: Comparison between Doppler Echocardiography and MR Velocity Mapping. *Pediatr Radiol* 38, 2 (2008), 186–191. (Cited on page 108.)
- [52] GUTBERLET, M., BOECKEL, T., HOSTEN, N., VOGEL, M., KÜHNE, T., OELLINGER, H., EHRENSTEIN, T., VENZ, S., HETZER, R., BEIN, G., AND FELIX, R. Arterial Switch Procedure for D-transposition of the Great Arteries: Quantitative Midterm Evaluation of Hemodynamic Changes with Cine MR Imaging and Phase-Shift Velocity Mapping-Initial Experience. *Radiology* 214, 2 (2000), 467–475. (Cited on pages 59 and 111.)
- [53] HANSON, L. G. Is Quantum Mechanics Necessary for Understanding Magnetic Resonance? *Concepts in Magnetic Resonance Part A* 32A, 5 (Sept. 2008), 329–340. (Cited on page 62.)
- [54] HANSON, L. G. Introduction to Magnetic Resonance Imaging Techniques. Tech. rep., University Hospital Hvidovre, DRCMR, Copenhagen, Denmark, 2009. (Cited on pages 62 and 69.)
- [55] HASHEMI, R. H., BRADLEY JR., W. G., AND LISANTI, C. J. *MRI: The Basics*. Lippincott Williams & Wilkins, Philadelphia, 2004. (Cited on page 62.)
- [56] HEIBERG, E., EBBERS, T., WIGSTROM, L., AND KARLSSON, M. Three-Dimensional Flow Characterization using Vector Pattern Matching. *IEEE TVCG* 9, 3 (2003), 313–319. (Cited on pages 21 and 81.)
- [57] HEIBERG, E., SJÖGREN, J., UGANDER, M., CARLSSON, M., ENGBLOM, H., AND ARHEDEN, H. Design and Validation of Segment - Freely Available Software for Cardiovascular Image Analysis. *BMC Med Imaging* 10 (2010), 1. (Cited on page 80.)
- [58] HELMAN, J. L., AND HESSELINK, L. Visualizing Vector Field Topology in Fluid Flows. *IEEE Comput Graph App* 11, 3 (1991), 36–46. (Cited on pages 13 and 14.)

- [59] HOPE, T. A., MARKL, M., WIGSTRÖM, L., ALLEY, M. T., MILLER, D. C., AND HERFKENS, R. J. Comparison of Flow Patterns in Ascending Aortic Aneurysms and Volunteers using Four-Dimensional Magnetic Resonance Velocity Mapping. *J Magn Reson Imaging* 26, 6 (2007), 1471–1479. (Cited on page 96.)
- [60] HORNAK, J. P. The Basics of MRI. <http://www.cis.rit.edu/htbooks/mri/> (1996), Last accessed Nov. 21, 2013. (Cited on page 62.)
- [61] HSU, W., MEI, J., CORREA, C., AND MA, K. Depicting Time Evolving Flow with Illustrative Visualization Techniques. In *Arts and Technology*. 2010, pp. 1–12. (Cited on page 25.)
- [62] HULTQUIST, J. Interactive Numerical Flow Visualization Using Stream Surfaces. *Computing Systems in Engineering* 1, 2-4 (1990), 349–353. (Cited on pages 11 and 17.)
- [63] HUMMEL, M., GARTH, C., HAMANN, B., HAGEN, H., AND JOY, K. I. IRIS: Illustrative Rendering for Integral Surfaces. *IEEE TVCG* 16, 6 (2010), 1319–1328. (Cited on pages 18, 24, 33, 54 and 124.)
- [64] INTERRANTE, V., AND GROSCH, C. Strategies for Effectively Visualizing a 3D Flow Using Volume Line Integral Convolution. *Strategies* (1997). (Cited on page 17.)
- [65] ISENBERG, T., FREUDENBERG, B., HALPER, N., SCHLECHTWEG, S., AND STROTHOTTE, T. A Developer’s Guide to Silhouette Algorithms for Polygonal Models. *IEEE Comput Graph App* 23, 4 (2003), 28–37. (Cited on page 38.)
- [66] JACKSON, B., COFFEY, D., AND KEEFE, D. F. Force Brushes: Progressive Data-Driven Haptic Selection and Filtering for Multi-Variate Flow Visualizations. In *Proc. of EuroVis (Shortpapers)* (2012), pp. 7–11. (Cited on page 18.)
- [67] JELLISON, B. J., FIELD, A. S., MEDOW, J., LAZAR, M., SALAMAT, M. S., AND ALEXANDER, A. L. Diffusion Tensor Imaging of Cerebral White Matter: A Pictorial Review of Physics, Fiber Tract Anatomy, and Tumor Imaging Patterns. *Am J Neuroradiol* 25, 3 (Mar. 2004), 356–69. (Cited on page 62.)
- [68] JEONG, J., AND HUSSAIN, F. On the Identification of a Vortex. *J Fluid Mech* 285 (1995), 69–94. (Cited on pages 19, 87 and 88.)
- [69] JEZZARD, P., MATTHEWS, P. M., AND SMITH, S. M. *Functional MRI: An Introduction to Methods*. Oxford University Press, New York, USA, 2001. (Cited on page 62.)
- [70] JIANG, M., MACHIRAJU, R., AND THOMPSON, D. A Novel Approach To Vortex Core Region Detection. In *Proc. of VisSym* (2002), pp. 217–225. (Cited on page 20.)

- [71] JIANG, M., MACHIRAJU, R., AND THOMPSON, D. Detection and Visualization of Vortices. In *The Visualization Handbook*, C. D. Hansen and C. R. Johnson, Eds. Academic Press, 2005, ch. 14, pp. 295–309. (Cited on page 19.)
- [72] JIANU, D. Visualizing Spatial Relations Between 3D-DTI Integral Curves Using Texture Patterns. In *Proc. of IEEE Visualization (Posters)* (2007). (Cited on page 18.)
- [73] JOBARD, B., AND LEFER, W. Creating Evenly-Spaced Streamlines of Arbitrary Density. *Visualization in Scientific Computing* (1997), 43–56. (Cited on pages 18 and 24.)
- [74] JONES, C., AND MA, K.-L. Visualizing Flow Trajectories using Locality-Based Rendering and Warped Curve Plots. *IEEE TVCG* 16, 6 (2010), 1587–1594. (Cited on page 26.)
- [75] JOSHI, A., CABAN, J., RHEINGANS, P., AND SPARLING, L. Case Study on Visualizing Hurricanes Using Illustration-Inspired Techniques. *IEEE TVCG* 15, 5 (2009), 709–718. (Cited on page 23.)
- [76] JOSHI, A., AND RHEINGANS, P. Illustration-Inspired Techniques for Visualizing Time-Varying Data. In *Proc. of IEEE Visualization* (2005), pp. 679–686. (Cited on page 25.)
- [77] KAINMÜLLER, D., UNTERHINNINGHOFEN, R., LEY, S., AND DILLMANN, R. Level Set Segmentation of the Heart from 4D Phase Contrast MRI. *Proc. of SPIE* 6914, 1 (2008). (Cited on page 86.)
- [78] KALIVAS, D. S., AND SAWCHUK, A. A. A Region Matching Motion Estimation Algorithm. *Computer Vision, Graphics, and Image Processing: Image Understanding* 54, 2 (1991), 275–288. (Cited on page 21.)
- [79] KAPLAN, M. Hybrid Quantitative Invisibility. In *Proc. of Symposium on Non-Photorealistic Animation and Rendering* (2007), pp. 51–52. (Cited on page 34.)
- [80] KILNER, P. J., YANG, G. Z., MOHIADDIN, R. H., FIRMIN, D. N., AND LONGMORE, D. B. Helical and Retrograde Secondary Flow Patterns in the Aortic Arch Studied by Three-Directional Magnetic Resonance Velocity Mapping. *Circulation* 88 (1993), 2235–2247. (Cited on pages 90, 96 and 116.)
- [81] KILNER, P. J., YANG, G. Z., WILKES, A. J., MOHIADDIN, R. H., FIRMIN, D. N., AND YACOB, M. H. Asymmetric Redirection of Flow Through the Heart. *Nature* 404, 6779 (2000), 759–761. (Cited on pages 59, 77 and 116.)
- [82] KIPFER, P., SEGAL, M., AND WESTERMANN, R. UberFlow : A GPU-Based Particle Engine. *Proc. of ACM SIGGRAPH/Eurographics Conference on Graphics Hardware* (2004), 115–122. (Cited on page 17.)

- [83] KIRBY, R., MARMANIS, H., AND LAIDLAW, D. Visualizing Multivalued Data from 2D Incompressible Flows Using Concepts from Painting. In *Proc. of IEEE Visualization* (1999), pp. 333–340. (Cited on pages 23 and 24.)
- [84] KÖHLER, B., GASTEIGER, R., PREIM, U., THEISEL, H., GUTBERLET, M., AND PREIM, B. Semi-Automatic Vortex Extraction in 4D PC-MRI Cardiac Blood Flow Data using Line Predicates. *IEEE TVCG* 19, 12 (Dec. 2013), 2773–82. (Cited on page 114.)
- [85] KRISHNAN, H., GARTH, C., GUHRING, J., GULSUN, M. A., GREISER, A., AND JOY, K. I. Analysis of Time-Dependent Flow-Sensitive PC-MRI Data. *IEEE TVCG* 17, 11 (Apr. 2011). (Cited on pages 80 and 81.)
- [86] KRISHNAN, H., GARTH, C., AND JOY, K. Time and Streak Surfaces for Flow Visualization in Large Time-Varying Data Sets. In *Proc. of IEEE Visualization* (2009), pp. 1267–1274. (Cited on pages 33 and 54.)
- [87] KRISHNAN, H., GARTH, C., AND JOY, K. I. Time and Streak Surfaces for Flow Visualization in Large Time-Varying Data Sets. *IEEE TVCG* 15, 6 (2009), 1267–74. (Cited on pages 11, 12, 16 and 18.)
- [88] KRÜGER, J., KIPFER, P., KONCLRATIEVA, P., AND WESTERMANN, R. A Particle System for Interactive Visualization of 3D Flows. *IEEE TVCG* 11, 6 (2005), 744–756. (Cited on page 17.)
- [89] KRÜGER, J., AND WESTERMANN, R. Efficient Stipple Rendering. In *Proc. of IADIS Computer Graphics and Visualization* (2007). (Cited on page 41.)
- [90] LAIDLAW, D. H., KIRBY, R. M., DAVIDSON, J. S., MILLER, T. S., SILVA, M. D., WARREN, W. H., AND TARR, M. Quantitative Comparative Evaluation of 2D Vector Field Visualization Methods. In *Proc. of IEEE Visualization* (2001), pp. 143–150. (Cited on page 17.)
- [91] LARAMEE, R., GARTH, C., SCHNEIDER, J., AND HAUSER, H. Texture Advection on Stream Surfaces: A Novel Hybrid Visualization Applied to CFD Simulation Results. In *Proc. of EuroVis* (2006), pp. 155–162. (Cited on pages 33 and 37.)
- [92] LARAMEE, R. S., HAUSER, H., DOLEISCH, H., VROLIJK, B., POST, F. H., AND WEISKOPF, D. The State of the Art in Flow Visualization: Dense and Texture-Based Techniques. *Comput Graph Forum* 23, 2 (2004), 203–221. (Cited on pages 8, 16 and 17.)
- [93] LARAMEE, R. S., HAUSER, H., ZHAO, L., AND POST, F. H. Topology-Based Flow Visualization, the State of the Art. In *Topology-Based Methods in Visualization (Proc. of TopoInVis 2005)*. Springer Berlin Heidelberg, 2007, pp. 1–19. (Cited on page 19.)

- [94] LARAMEE, R. S., JOBARD, B., AND HAUSER, H. Image Space Based Visualization of Unsteady Flow on Surfaces. In *Proc. of IEEE Visualization* (2003), pp. 131–138. (Cited on page 17.)
- [95] LEVY, Y., DEGANI, D., AND SEGNER, A. Graphical Visualization of Vortical Flow by Means of Helicity. *AIAA J* 28, 8 (1990), 1347–1352. (Cited on page 20.)
- [96] LI, G.-S., BORDOLOI, U. D., AND SHEN, H.-W. Chameleon: An Interactive Texture-Based Rendering Framework for Visualizing Three-Dimensional Vector Fields. In *Proc. of IEEE Visualization* (2003), pp. 241–248. (Cited on page 18.)
- [97] LI, G.-S., TRICOCHÉ, X., WEISKOPF, D., AND HANSEN, C. Flow Charts: Visualization of Vector Fields on Arbitrary Surfaces. *IEEE TVCG* 14, 5 (2008), 1067–1080. (Cited on pages 16 and 17.)
- [98] LI, W., RITTER, L., AGRAWALA, M., CURLESS, B., AND SALESIN, D. Interactive Cutaway Illustrations of Complex 3D Models. In *Proc. of ACM SIGGRAPH* (2007), no. 31. (Cited on pages 18 and 34.)
- [99] LÖFFELMANN, H. *Visualizing Local Properties and Characteristic Structures of Dynamical Systems*. PhD thesis, Technische Universität Wien, 1998. (Cited on pages 18 and 33.)
- [100] LÖFFELMANN, H., KUCERA, T., AND GRÖLLER, E. Visualizing Poincaré Maps Together with the Underlying Flow. In *Mathematical Visualization, Algorithms, Applications, and Numerics*, H.-C. Hege and K. Polthier, Eds. Springer Berlin Heidelberg, 1998, pp. 315–328. (Cited on page 34.)
- [101] LÖFFELMANN, H., MROZ, L., GRÖLLER, E., AND PURGATHOFER, W. Stream Arrows: Enhancing the Use of Stream Surfaces for the Visualization of Dynamical Systems. *Visual Comput* 13, 8 (1997), 359–369. (Cited on pages 18, 24 and 33.)
- [102] LOTZ, J., MEIER, C., LEPPERT, A., AND GALANSKI, M. Cardiovascular Flow Measurement with Phase-Contrast MR Imaging : Basic Facts and Implementation. *RadioGraphics* 22 (2002), 651–671. (Cited on pages 70 and 72.)
- [103] LU, A., AND SHEN, H.-W. Interactive Storyboard for Overall Time-Varying Data Visualization. In *Proc. of IEEE Pacific Visualization* (2008), pp. 143–150. (Cited on page 26.)
- [104] MALLO, O., PEIKERT, R., SIGG, C., AND SADLO, F. Illuminated Lines Revisited. In *Proc. of IEEE Visualization* (2005), pp. 19–26. (Cited on page 18.)
- [105] MARCHESIN, S., CHEN, C.-K. C. C.-K., HO, C., AND MA, K.-L. M. K.-L. View-Dependent Streamlines for 3D Vector Fields. *IEEE TVCG* 16, 6 (2010), 1578–1586. (Cited on page 18.)

- [106] MARKL, M. Velocity Encoding and Flow Imaging. <http://ee-classes.usc.edu/ee591/library/Markl-FlowImaging.pdf> (2006), Last access Nov. 21, 2013. (Cited on pages 70 and 74.)
- [107] MARKL, M. Marked Three-Dimensional Flow Pattern Changes in Distorted Aortic Geometry. *Eur Heart J* 32, 6 (2011), 679. (Cited on pages 97 and 98.)
- [108] MARKL, M., CHAN, F. P., ALLEY, M. T., WEDDING, K. L., DRANEY, M. T., ELKINS, C. J., PARKER, D. W., WICKER, R., TAYLOR, C. A., HERFKENS, R. J., AND PELC, N. J. Time-Resolved Three-Dimensional Phase-Contrast MRI. *J Magn Reson Imaging* 17, 4 (2003), 499–506. (Cited on page 80.)
- [109] MARKL, M., HARLOFF, A., BLEY, T. A., ZAITSEV, M., JUNG, B., WEIGANG, E., LANGER, M., HENNIG, J., AND FRYDRYCHOWICZ, A. Time-Resolved 3D MR Velocity Mapping at 3T: Improved Navigator-Gated Assessment of Vascular Anatomy and Blood Flow. *J Magn Reson Imaging* 25, 4 (2007), 824–831. (Cited on page 95.)
- [110] MARKL, M., KILNER, P. J., AND EBBERS, T. Comprehensive 4D Velocity Mapping of the Heart and Great Vessels by Cardiovascular Magnetic Resonance. *J Cardiovasc Magn R* 13, 7 (2011). (Cited on pages 3, 62, 75, 78, 80 and 113.)
- [111] MARSDEN, J. E., AND TROMBA, A. J. *Vector Calculus*, 5th ed. W.H. Freeman and Company, New York, USA, 2003. (Cited on page 7.)
- [112] MATTAUSCH, O., THEUSS L, T., HAUSER, H., AND GRÖLLER, E. Strategies for Interactive Exploration of 3D Flow Using Evenly-Spaced Illuminated Streamlines. In *Proc. of Spring Conference on Computer Graphics* (2001), pp. 213–222. (Cited on pages 18 and 26.)
- [113] MCLOUGHLIN, T., LARAMEE, R., PEIKERT, R., POST, F., AND CHEN, M. Over Two Decades of Integration-Based, Geometric Flow Visualization. In *Eurographics State of the Art Reports* (2009), pp. 73–92. (Cited on page 10.)
- [114] MCLOUGHLIN, T., LARAMEE, R. S., PEIKERT, R., POST, F. H., AND CHEN, M. Over Two Decades of Integration-Based, Geometric Flow Visualization. *Comput Graph Forum* 29, 6 (2010), 1807–1829. (Cited on pages 11, 17 and 18.)
- [115] MCROBBIE, D. W., MOORE, E. A., GRAVES, M. J., AND PRINCE, M. R. *MRI From Picture to Proton*. Cambridge University Press, New York, USA, 2006. (Cited on pages 62 and 75.)
- [116] MOIN, P., AND KIM, J. The Structure of the Vorticity Field in Turbulent Channel Flow: Analysis of Instantaneous Fields and Statistical Correlations. *J Fluid Mech* 155 (1985), 441–464. (Cited on page 19.)

- [117] MUELDER, C., AND MA, K.-L. Interactive Feature Extraction and Tracking by Utilizing Region Coherency. *Proc. of IEEE Pacific Visualization* (Apr. 2009), 17–24. (Cited on pages 21 and 22.)
- [118] MUNZNER, T., JOHNSON, C., MOORHEAD, R., PFISTER, H., RHEINGANS, P., AND YOO, T. S. NIH-NSF Visualization Research Challenges Report Summary. *IEEE Comput Graph* 26, 2 (2006), 20–24. (Cited on page 1.)
- [119] NEUGEBAUER, M., JANIGA, G., BEUING, O., SKALEJ, M., AND PREIM, B. Anatomy-guided Multi-Level Exploration of Blood Flow in Cerebral Aneurysms. *Comput Graph Forum* 30, 3 (2011), 1041–1050. (Cited on page 81.)
- [120] NIENHAUS, M., AND DÖLLNER, J. Blueprints - Illustrating Architecture and Technical Parts Using Hardware-Accelerated Non-Photorealistic Rendering. In *Proc. of Graphics Interface* (2004), pp. 49–56. (Cited on pages 34 and 40.)
- [121] NISHIMURA, R. Aortic Valve Disease. *Circulation* 106, 7 (Aug. 2002), 770–772. (Cited on page 61.)
- [122] OBERMAIER, H., KUHNERT, J., HERING-BERTRAM, M., AND HAGEN, H. Stream Volume Segmentation of Grid-less Flow Simulation. In *Topological Methods in Data Analysis and Visualization: Theory, Algorithms and Applications (Proc. of TopoInVis 2009)*. Springer Berlin Heidelberg, 2011, pp. 127–138. (Cited on page 18.)
- [123] PAGENDARM, H., HENNE, B., AND RÜTTEN, M. Detecting Vortical Phenomena in Vector Data by Medium-Scale Correlation. In *Proc. of IEEE Visualization* (1999), pp. 409–412. (Cited on page 20.)
- [124] PALÁGYI, K., AND KUBA, A. A 3D 6-Subiteration Thinning Algorithm for Extracting Medial Lines. *Pattern Recogn Lett* 19, 7 (1998), 613–627. (Cited on page 121.)
- [125] PATEL, D., GIERTSEN, C., THURMOND, J., GJELBERG, J., AND GRÖLLER, E. The Seismic Analyzer - Interpreting and Illustrating 2D Seismic Data. In *Proc. of IEEE Visualization* (2008), pp. 1571–1578. (Cited on page 23.)
- [126] PEIKERT, R., AND ROTH, M. The Parallel Vectors Operator: A Vector Field Visualization Primitive. *Proc. of IEEE Visualization* (1999), 263–270. (Cited on pages 88 and 124.)
- [127] PEIKERT, R., AND SADLO, F. Topology-guided Visualization of Constrained Vector Fields. In *Topology-Based Methods in Visualization* (2007), H. Hauser, H. Hagen, and H. Theisel, Eds., Springer-Verlag, pp. 21–34. (Cited on page 50.)

- [128] PEIKERT, R., AND SADLO, F. Topologically Relevant Stream Surfaces for Flow Visualization. In *Proc. of Spring Conference on Computer Graphics* (2009), pp. 43–50. (Cited on pages 12, 15 and 18.)
- [129] PELC, N. J., SOMMER, F. G., LI, K. C., BROSNAN, T. J., HERFKENS, R. J., ENZMANN, D. R., AND CALIFORNIA, S. Quantitative Magnetic Resonance Flow Imaging. *Magn Reson Quart* 10, 3 (1994), 125–147. (Cited on pages 74 and 75.)
- [130] POBITZER, A., LEZ, A., MATKOVIC, K., AND HAUSER, H. A Statistics-Based Dimension Reduction of the Space of Path Line Attributes for Interactive Visual Flow Analysis. In *Proc. of IEEE Pacific Visualization* (2012), pp. 113–120. (Cited on page 81.)
- [131] POINCARÉ, H. *Les Méthodes nouvelles de la Mécanique Céleste*, vol. I. Gauthier-Villar et fils, Paris, 1899. (Cited on page 10.)
- [132] POST, F. H., VROLIJK, B., HAUSER, H., LARAMEE, R. S., AND DOLEISCH, H. The State of the Art in Flow Visualisation: Feature Extraction and Tracking. *Comput Graph Forum* 22, 4 (Dec. 2003), 775–792. (Cited on pages 15, 19 and 21.)
- [133] POST, F. J., WALSUM, T. V., POST, F. H., AND SILVER, D. Iconic Techniques for Feature Visualization. In *Proc. of IEEE Visualization* (1995), no. Section 3, pp. 288–295. (Cited on page 21.)
- [134] RAUTEK, P., BRUCKNER, S., AND GRÖLLER, E. Semantic Layers for Illustrative Volume Rendering. *IEEE TVCG* 13, 6 (2007), 1336–1343. (Cited on page 82.)
- [135] RAUTEK, P., BRUCKNER, S., GRÖLLER, E., AND VIOLA, I. Illustrative Visualization – New Technology or Useless Tautology? *Comput Graph* 42, 3 (2008), 4:1–4:8. (Cited on page 23.)
- [136] REINDERS, F., POST, F. H., AND SPOELDER, H. J. Attribute-Based Feature Tracking. In *Data Visualization*. 1999, pp. 63–72. (Cited on page 21.)
- [137] REINDERS, F., POST, F. H., AND SPOELDER, H. J. Visualization of Time-Dependent Data with Feature Tracking and Event Detection. *Visual Comput* 17, 1 (Feb. 2001), 55–71. (Cited on pages 21, 22 and 124.)
- [138] REZK-SALAMA, C., HASTREITER, P., TEITZEL, C., AND ERTL, T. Interactive Exploration of Volume Line Integral Convolution Based on 3D-Texture Mapping. In *Proc. of IEEE Visualization* (1999), pp. 233–240. (Cited on page 17.)
- [139] ROBINSON, S. A Review of Vortex Structures and Associated Coherent Motions in Turbulent Boundary Layers. In *Proc. of IUTAM Symposium on Structure of Turbulence and Drag Reduction* (1989), pp. 23–50. (Cited on page 19.)

- [140] ROBINSON, S. K. Coherent Motions In The Turbulent Boundary Layer. *Ann Rev of Fluid Mech* 23 (Jan. 1991), 601–639. (Cited on page 19.)
- [141] RODÉS-CABAU, J. Transcatheter Aortic Valve Implantation: Current and Future Approaches. *Nat Rev Cardiol* 9, 1 (Jan. 2012), 15–29. (Cited on page 61.)
- [142] ROTH, M., AND PEIKERT, R. A Higher-Order Method for Finding Vortex Core Lines. In *Proc. of IEEE Visualization* (1998), pp. 143–150. (Cited on pages 21 and 87.)
- [143] SADARJOEN, I., POST, F., BANKS, D., AND PAGENDARM, H.-G. Selective Visualization of Vortices in Hydrodynamic Flows. In *Proc. of IEEE Visualization* (1998), pp. 419–422,. (Cited on page 20.)
- [144] SADARJOEN, I. A., AND POST, F. H. Geometric Methods for Vortex Extraction. In *Proc. of VisSym* (1999), pp. 53–62. (Cited on pages 20 and 21.)
- [145] SALAH, Z., BARTZ, D., STRASSER, W., AND TATAGIBA, M. Expressive Anatomical Illustration Based on Scanned Patient Data. *GMS Current Topics in Computer and Robot Assisted Surgery Journal* 1 (2006). (Cited on page 23.)
- [146] SALZBRUNN, T. *Flow Visualization and Analysis Based on Integral Line Predicates*. PhD thesis, University of Leipzig, 2008. (Cited on pages 83, 121 and 130.)
- [147] SALZBRUNN, T., GARTH, C., SCHEUERMANN, G., AND MEYER, J. Pathline Predicates and Unsteady Flow Structures. *Visual Comput* 24, 12 (2008), 1039–1051. (Cited on pages 78, 81 and 82.)
- [148] SALZBRUNN, T., AND SCHEUERMANN, G. Streamline Predicates. *IEEE TVCG* 12, 6 (2006), 1601–1612. (Cited on pages 78, 81, 82, 86 and 87.)
- [149] SAYEED, R., AND T.L.J.HOWARD. State-of-the-art of Non-Photorealistic Rendering (NPR) for Visualisation. In *Theory and Practice of Computer Graphics 2006* (2006). (Cited on page 23.)
- [150] SCHAFHITZEL, T., TEJADA, E., WEISKOPF, D., AND ERTL, T. Point-Based Stream Surfaces and Path Surfaces. In *Proc. of Graphics Interface* (2007), pp. 289–296. (Cited on page 18.)
- [151] SCHAFHITZEL, T., WEISKOPF, D., AND ERTL, T. Interactive Investigation and Visualization of 3D Vortex Structures. In *Proc. of International Symposium on Flow Visualization* (2006). (Cited on page 21.)
- [152] SCHEUERMANN, G., BOBACH, T., HAGEN, H., MAHROUS, K., HAMANN, B., JOY, K., AND KOLLMANN, W. A Tetrahedra-Based Stream Surface Algorithm. In *Proc. of IEEE Visualization* (2001), pp. 151–158. (Cited on page 17.)

-
- [153] SCHMIDT, R. F., AND THEWS, G. *Physiologie des Menschen*. Springer Verlag, Berlin Heidelberg, 2000. (Cited on pages 57 and 58.)
- [154] SCHNEIDER, D., REICH, W., WIEBEL, A., AND SCHEUERMANN, G. Topology Aware Stream Surfaces. *Comput Graph Forum* 29, 3 (2010), 1153–1161. (Cited on page 18.)
- [155] SCHNEIDER, D., WIEBEL, A., AND SCHEUERMANN, G. Smooth Stream Surfaces of 4th Order Precision. *Comput Graph Forum* 28, 3 (2009), 871–878. (Cited on page 17.)
- [156] SENGUPTA, P. P., PEDRIZZETTI, G., KILNER, P. J., KHERADVAR, A., EBBERS, T., TONTI, G., FRASER, A. G., AND NARULA, J. Emerging Trends in CV Flow Visualization. *JACC. Cardiovascular Imaging* 5, 3 (Mar. 2012), 305–16. (Cited on pages 59 and 77.)
- [157] SETTLES, G. The unseen, shimmering world around a candle burning in a breeze (photograph). http://www.nytimes.com/slideshow/2008/10/27/science/102808-Cough_7.html (2008), Last access Nov. 21, 2013. (Cited on page 2.)
- [158] SHI, K., THEISEL, H., HAUSER, H., WEINKAUF, T., MATKOVIC, K., HEGE, H.-C., AND SEIDEL, H. P. Path Line Attributes - An Information Visualization Approach to Analyzing the Dynamic Behavior of 3D Time-Dependent Flow Fields. In *Topology-Based Methods in Visualization (Proc. of TopoInVis 2007)*, Mathematics and Visualization. Springer Berlin Heidelberg, Berlin Heidelberg, 2009, pp. 75–88. (Cited on page 81.)
- [159] SILVA, S., SOUSA SANTOS, B., AND MADEIRA, J. Using Color in Visualization: A Survey. *Comput Graph* 35, 2 (2011), 320–333. (Cited on page 94.)
- [160] SILVER, D., AND WANG, X. Tracking Scalar Features in Unstructured Data Sets. In *Proc. of IEEE Visualization* (1998), pp. 79–86,. (Cited on page 21.)
- [161] SOBEL, J. S., FORSBERG, A. S., LAIDLAW, D. H., ZELEZNIK, R. C., KEEFE, D. F., PIVKIN, I., AND KARNIADAKIS, G. E. Particle Flurries - Synoptic 3D Pulsatile Flow Visualization. *IEEE Comput Graph* 24, 2 (2004), 76–85. (Cited on page 18.)
- [162] SOUSA, M., EBERT, D., STREDNEY, D., AND SVAKHINE, N. Illustrative Visualization for Medical Training. In *Proc. of EG Workshop on Computational Aesthetics in Graphics, Visualization and Imaging* (2005), pp. 201–208. (Cited on page 23.)
- [163] STALLING, D. *Fast Texture-Based Algorithms for Vector Field Visualization*. PhD thesis, Freie Universität Berlin, 1998. (Cited on page 18.)

- [164] STEGMAIER, S., RIST, U., AND ERTL, T. Opening the Can of Worms: An Exploration Tool for Vortical Flows. In *Proc. of IEEE Visualization* (2005), pp. 59–59. (Cited on pages 20 and 21.)
- [165] STOLL, C., GUMHOLD, S., AND SEIDEL, H.-P. Visualization with Stylized Line Primitives. In *Proc. of IEEE Visualization* (2005), pp. 695–702. (Cited on page 24.)
- [166] STROTHOTTE, T., AND SCHLECHTWEG, S. *Non-Photorealistic Computer Graphics: Modeling, Rendering, and Animation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, 2002. (Cited on pages 23 and 41.)
- [167] SUJUDI, D., AND HAIMES, R. Identification of Swirling Flow in 3D Vector Fields. Tech. Rep. AIAA Paper 95–1715, American Institute of Aeronautics and Astronautics, 1995. (Cited on pages 21, 87, 88 and 124.)
- [168] SVAKHINE, N., JANG, Y., EBERT, D., AND GAITHER, K. Illustration and Photography Inspired Visualization of Flows and Volumes. In *Proc. of IEEE Visualization* (2005), pp. 687–694. (Cited on page 23.)
- [169] SVAKHINE, N. A., EBERT, D. S., AND ANDREWS, W. M. Illustration-Inspired Depth Enhanced Volumetric Medical Visualization. *IEEE TVCG* 15, 1 (2009), 77–86. (Cited on page 23.)
- [170] TAO, J., MA, J., WANG, C., AND SHENE, C.-K. A Unified Approach to Streamline Selection and Viewpoint Selection for 3D Flow Visualization. *IEEE TVCG* 19, 3 (2013), 393–406. (Cited on page 18.)
- [171] TELEA, A., AND WIJK, J. J. V. 3D IBFV: Hardware-Accelerated 3D Flow Visualization. In *Proc. of IEEE Visualization* (2003), pp. 233–240. (Cited on page 17.)
- [172] THEISEL, H., AND SEIDEL, H. Feature Flow Fields. In *Proc. of VisSym* (2003), pp. 141–148. (Cited on page 21.)
- [173] THEISEL, H., WEINKAUF, T., HEGE, H., AND SEIDEL, H. Saddle Connectors - An Approach to Visualizing the Topological Skeleton of Complex 3D Vector Fields. In *Proc. of IEEE Visualization* (2003), pp. 225–232. (Cited on page 33.)
- [174] TIETJEN, C., ISENBERG, T., AND PREIM, B. Combining Silhouettes, Surface, and Volume Rendering for Surgery Education and Planning. In *Proc. of EuroVis* (2005), pp. 303–310. (Cited on page 23.)
- [175] TRICOCHÉ, X., GARTH, C., KINDLMANN, G., DEINES, E., SCHEUERMANN, G., RUETTEN, M., AND HANSEN, C. Visualization of Intricate Flow Structures for Vortex Breakdown Analysis. In *Proc. of IEEE Visualization* (2004), pp. 187–194. (Cited on page 54.)

- [176] TRICOCHÉ, X., WISCHGOLL, T., SCHEUERMANN, G., AND HAGEN, H. Topology Tracking for the Visualization of Time-Dependent Two-Dimensional Flows. *Comput Graph* 26, 2 (2002), 249–257. (Cited on page 21.)
- [177] TURK, G., AND BANKS, D. Image-Guided Streamline Placement. In *Proc. of ACM SIGGRAPH* (1996), pp. 453–460. (Cited on pages 18 and 24.)
- [178] UENG, S.-K. U. S.-K., SIKORSKI, C., AND MA, K.-L. M. K.-L. Efficient Streamline, Streamribbon, and Streamtube Constructions on Unstructured Grids. *IEEE TVCG* 2, 2 (1996), 100–110. (Cited on pages 17 and 18.)
- [179] VAN PELT, R., BESCÓS, J. O., BREEUWER, M., CLOUGH, R. E., GRÖLLER, M. E., ROMENIJ, B. T. H., AND VILANOVA, A. Interactive Virtual Probing of 4D MRI Blood-Flow. *IEEE TVCG* 17, 12 (Dec. 2011), 2153–62. (Cited on pages 81, 117 and 118.)
- [180] VAN PELT, R., NGUYEN, H., TER HAAR ROMENY, B., AND VILANOVA, A. Automated Segmentation of Blood-Flow Regions in Large Thoracic Arteries using 3D-Cine PC-MRI Measurements. *Int J Comput Assist Radiol Surg* 7, 2 (2011), 217–224. (Cited on page 86.)
- [181] VAN PELT, R., OLIVÁN BESCÓS, J., BREEUWER, M., CLOUGH, R. E., GRÖLLER, M. E., TER HAAR ROMENY, B., AND VILANOVA, A. Exploration of 4D MRI Blood-Flow using Stylistic Visualization. *IEEE TVCG* 16 (2010), 1339–1347. (Cited on pages 81, 117 and 118.)
- [182] VAN PELT, R. F. P., JACOBS, S. S. A. M., TER HAAR ROMENY, B. M., AND VILANOVA, A. Visualization of 4D Blood-Flow Fields by Spatiotemporal Hierarchical Clustering. *Comput Graph Forum* 31 (June 2012), 1065–1074. (Cited on pages 81, 117 and 118.)
- [183] VAN WIJK, J. J. Spot Noise Texture Synthesis for Data Visualization. In *Proc. of ACM SIGGRAPH* (1991), pp. 309–318. (Cited on page 16.)
- [184] VAN WIJK, J. J. Image Based Flow Visualization. *ACM Trans. on Graphics* 21, 3 (2002), 745–754. (Cited on page 17.)
- [185] VAN WIJK, J. J. Image Based Flow Visualization for Curved Surfaces. In *Proc. of IEEE Visualization* (2003), pp. 123–130. (Cited on page 17.)
- [186] VERMA, V., KAO, D., AND PANG, A. A Flow-Guided Streamline Seeding Strategy. In *Proc. of IEEE Visualization* (2000), pp. 163–170. (Cited on page 18.)
- [187] VILLASENOR, J., AND VINCENT, A. An Algorithm for Space Recognition and Time Tracking of Vorticity Tubes in Turbulence. *Computer Vision, Graphics, and Image Processing: Image Understanding* 55, 1 (1992), 27–35. (Cited on page 19.)

- [188] VIOLA, I., AND GRÖLLER, E. Smart Visibility in Visualization. In *Proc. of International Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging* (2005), pp. 209–216. (Cited on page 25.)
- [189] WAKTARE, J. E. Atrial Fibrillation. *Circulation* 106, 1 (2002), 14–16. (Cited on page 60.)
- [190] WALKER, P. G., CRANNEY, G. B., SCHEIDEGGER, M. B., WASELESKI, G., POHOST, G. M., AND YOGANATHAN, A. P. Semiautomated Method for Noise Reduction and Background Phase Error Correction in MR Phase Velocity Data. *J Magn Reson Imaging* 3, 3 (1993), 521–530. (Cited on page 101.)
- [191] WALSUM, T. V., POST, F. H., SOCIETY, I. C., SILVER, D., AND POST, F. J. Feature Extraction and Iconic Visualization. *IEEE TVCG* 2, 2 (1996), 111–119. (Cited on page 21.)
- [192] WANDELL, B., DUMOULIN, S., AND BREWER, A. Visual Cortex in Humans. *Encyclopedia of Neuroscience* 10 (2009), 251–257. (Cited on page 1.)
- [193] WARE, C. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, 2004. (Cited on page 36.)
- [194] WEBER, J. ProteinShader: Illustrative Rendering of Macromolecules. *BMC Struct Biol* 9, 19 (2009). (Cited on page 23.)
- [195] WEIGLE, C., AND BANKS, D. C. Extracting Iso-Valued Features in 4-Dimensional Scalar Fields. In *Proc. of IEEE Symposium on Volume Visualization* (1998), pp. 103–110. (Cited on page 21.)
- [196] WEINKAUF, T., HEGE, H.-C., AND THEISEL, H. Advected Tangent Curves: A General Scheme for Characteristic Curves of Flow Fields. *Comput Graph Forum* 31, 2 (2012), 825–834. (Cited on page 10.)
- [197] WEINKAUF, T., SAHNER, J., THEISEL, H., AND HEGE, H.-C. Cores of Swirling Particle Motion in Unsteady Flows. *IEEE TVCG* 13, 6 (2007), 1759–1766. (Cited on pages 21 and 22.)
- [198] WEINKAUF, T., AND THEISEL, H. Streak Lines as Tangent Curves of a Derived Vector Field. *IEEE TVCG* 16, 6 (2010), 1225–34. (Cited on page 10.)
- [199] WIKIMEDIA COMMONS (RHCASTILHOS). Gray506.png vector version. <http://en.wikipedia.org/wiki/File:Gray506.svg> (2007), License: Public Domain. Last access Nov. 21, 2013. (Cited on page 58.)
- [200] WIKIMEDIA COMMONS (WAPCAPLET,YADDAH). Diagram of the Human Heart. [http://commons.wikimedia.org/wiki/File:Diagram_of_the_human_heart_\(cropped\).svg](http://commons.wikimedia.org/wiki/File:Diagram_of_the_human_heart_(cropped).svg), License: GFDL. Last access Nov. 21, 2013. (Cited on page 58.)

-
- [201] WU, K., LIU, Z., ZHANG, S., AND MOORHEAD II, R. J. Topology-Aware Evenly Spaced Streamline Placement. *IEEE TVCG* 16, 5 (2010), 791–801. (Cited on page 18.)
- [202] YE, X. Y. X., KAO, D., AND PANG, A. Strategy for Seeding 3D Streamlines. In *Proc. of IEEE Visualization* (2005), pp. 471–478. (Cited on page 18.)
- [203] ZÖCKLER, M., STALLING, D., AND HEGE, H. C. Interactive Visualization of 3D-Vector Fields using Illuminated Stream Lines. In *Proc. of IEEE Visualization* (1996), pp. 107–113. (Cited on page 18.)

Selbständigkeitserklärung

Hiermit erkläre ich, die vorliegende Dissertation selbständig und ohne unzulässige fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angeführten Quellen und Hilfsmittel benutzt und sämtliche Textstellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen wurden, und alle Angaben, die auf mündlichen Auskünften beruhen, als solche kenntlich gemacht. Ebenfalls sind alle von anderen Personen bereitgestellten Materialien oder erbrachten Dienstleistungen als solche gekennzeichnet.

Zürich, 16. Juli 2014

(Ort, Datum)

S. Born

(Silvia Born)