Visual Exploration and Comparison of Cardiac and Cerebral Blood Flow Data

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke Universität Magdeburg

von M.Sc. Benjamin Behrendt
geb. 14.10.1989 in Magdeburg

OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

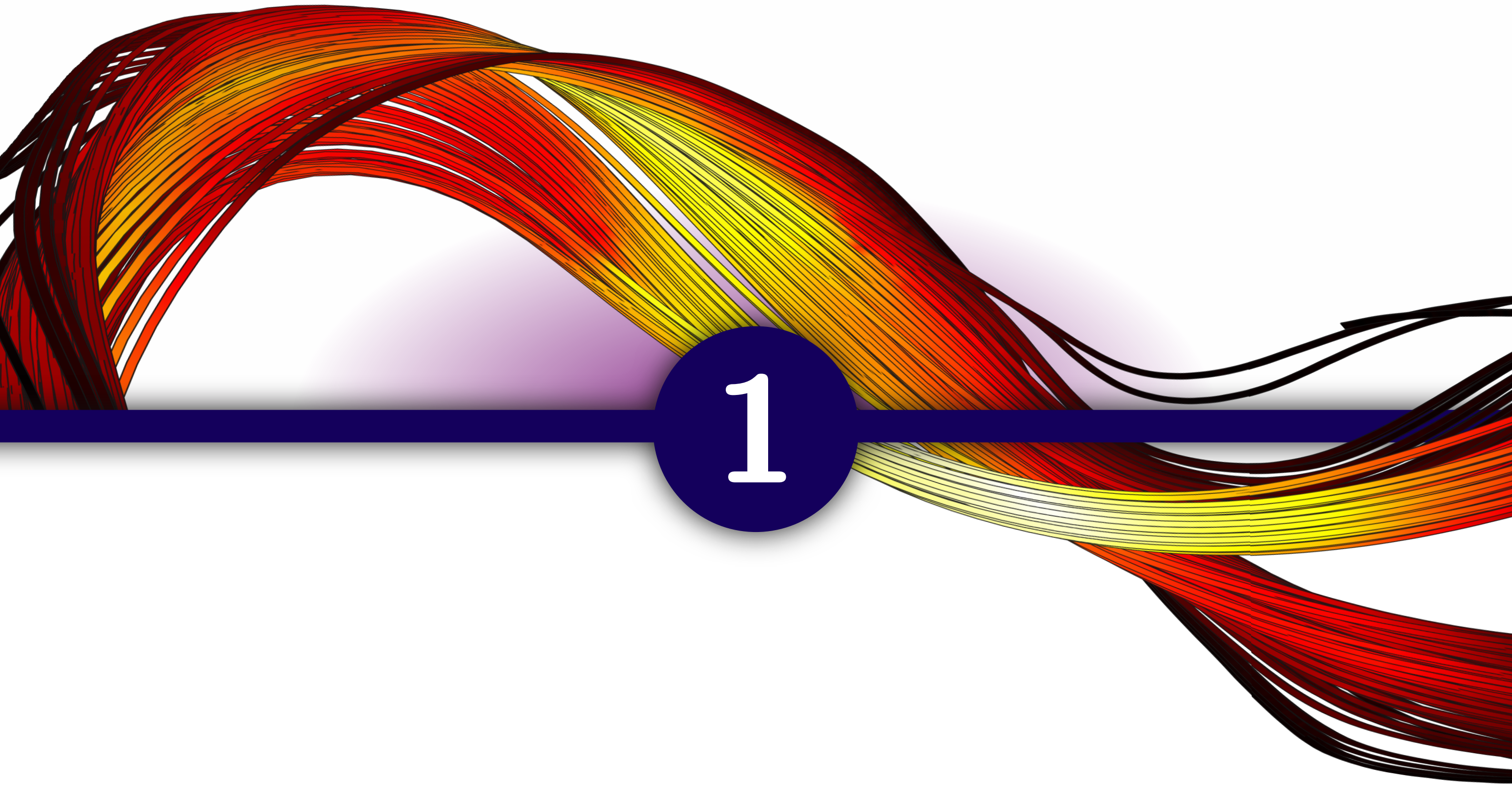
Gutachter

Prof. Dr.-Ing. Bernhard Preim
Prof. Dr. rer. nat. Christoph Garth
Prof. Dr.techn. Eduard Gröller

Magdeburg, den 03.12.2020

Benjamin Behrendt: *Visual Exploration and Comparison of Cardiac and Cerebral Blood Flow Data*; Dissertation, Otto-von-Guericke Universität Magdeburg. © 03.12.2020

Contents



1

Preliminaries

Abstract 1.1

The ability to visually explore and compare blood flow data can be a valuable aid for medical research regarding pathologies of the human vascular system. The use of explorative visualizations for blood flow can lead to a deeper understanding of the interrelation of vascular morphology and hemodynamics in general and the effects of specific pathologies on the vessels in particular. For example, the rupture risk assessment for cerebral aneurysms requires a detailed understanding of the complex interactions between vessel wall and underlying blood flow behavior. By comparing flow patterns, systematic differences and similarities within and between cohorts can be detected to formulate new medical hypotheses. As an example, such an analysis could support identifying the difference between changes in vascular hemodynamics as a result of "healthy" aging and the genesis of a cardiovascular disease.

Blood flow data can be acquired by various techniques. 4D PC-MRI is a non-invasive imaging modality that allows for the acquisition of in-vivo flow information. Vessels that are not large enough to be captured adequately by 4D PC-MRI can still be acquired using angiography techniques, such as CT Angiography, 3D Rotational Subtraction Angiography or MR Angiography. As these modalities do not record blood flow, computational fluid dynamics simulations are employed to obtain flow information based vessel geometries in these cases.

The visualization of the resulting three-dimensional, time-resolved blood flow data is challenging due to the high amount of occlusion in the spatio-temporal domain. Thus, the visualization has to be carefully crafted to reduce the occlusion based on the researchers' specific interests without compromising their orientation and spatial awareness. An additional problem, especially for simulated data, is the coverage of all existing features in the datasets domain using integral objects, such as pathlines, as many established seeding strategies do not guarantee full coverage without investing an unfeasible amount of computing power. Lastly, new insights into the vascular system cannot only be gained by exploring single datasets, but also by performing comparisons of multiple datasets between both patients and healthy volunteers. While a simple juxtaposition may be feasible for a lower amount of datasets, comparisons within a database of hundreds or more datasets require specialized metrics as well as interaction and visualization techniques to effectively support the user in gaining knowledge.

This thesis presents and evaluates techniques to support the effective visual exploration and comparison of blood flow data with the goal of gaining new insights into the human vascular system. This encompasses both general visualization and preprocessing methods (e.g. to improve depth perception), as well as tailored solutions for specific medical research questions. Our focus is on both blood flow data acquired in-vivo by means of 4D PC-MRI and using computational fluid dynamics simulations. Although there are significant differences between measured and simulated blood flow, all techniques implemented as part of this PhD thesis are integrated into a single framework.

Zusammenfassung 1.2

Die visuelle Exploration und der Vergleich von Blutflussdaten können ein wertvolles Werkzeug für die Erforschung des menschlichen kardiovaskulären Systems darstellen. Der Einsatz von explorativen Visualisierungen für die Darstellung von Blutfluss unterstützt das Verständnis der Zusammenhänge zwischen Gefäßmorphologie und Hämodynamik im Allgemeinen sowie dem Einfluss verschiedener Pathologien im Speziellen. Beispielsweise ist für die Vorhersage von Ruptur-Risiken zerebraler Aneurysmen ein detailliertes Verständnis der komplexen Beziehung von Gefäßwand und dem Verhalten des darunterliegenden Blutflusses erforderlich. Durch den Vergleich von Flussmustern können systematische Unterschiede und Gemeinsamkeiten sowohl *in* als auch *zwischen* Kohorten ermittelt und daraus neue medizinische Hypothesen abgeleitet werden. Beispielsweise könnten Unterschiede zwischen Veränderungen der Gefäßhämodynamik durch „gesundes" Altern und der Herausbildung einer kardiovaskulären Pathologie identifiziert werden.

Blutflussdaten können durch verschiedene Techniken akquiriert werden. 4D PC-MRI ist eine nichtinvasive Bildgebungsmodalität mit der sich in-vivo Flussinformationen aufnehmen lassen. Gefäße, die für eine Erfassung in 4D PC-MRI Scannern nicht groß genug sind, können trotzdem durch andere angiographische Techniken wie CT-Angiographie, 3D Rotationssubtraktionsangiographie sowie MR-Angiographie aufgenommen werden. Da diese Bildgebungsmodalitäten den Blutfluss selbst nicht abbilden können, werden in diesen Fällen die Blutflussinformationen durch Methoden der numerischen Strömungsmechanik basierend auf der Gefäßgeometrie berechnet.

Die Visualisierung der entstehenden dreidimensionalen und zeitaufgelösten Blutflussinformationen ist aufgrund der zahlreichen zeitlichen und räumlichen Überlappungen von Strukturen herausfordernd. Daher müssen die verwendeten Visualisierungstechniken basierend auf den Forschungsinteressen des Anwenders so konzipiert werden, dass diese Überlappungen aufgelöst oder reduziert werden, ohne die Orientierungsfähigkeit oder das Raumverständnis des Betrachters zu beeinträchtigen. Ein zusätzliches Problem, welches insbesondere bei simulierten Daten auftritt, ist es eine ausreichende Repräsentation der Flusseigenschaften durch integrale Objekte, wie Pfadlinien, zu erreichen. Viele Saat-Strategien können ohne den Einsatz unverhältnismäßig großer Rechenleistung keine vollständige Abdeckung des Flussfeldes garantieren. Natürlich können neue Erkenntnisse nicht nur durch die Betrachtung einzelner Datensätze, sondern insbesondere durch den Vergleich von Patienten und gesunden Probanden erreicht werden. Während eine einfache Gegenüberstellung für eine kleinere Menge an Datensätzen möglich ist, erfordern große Datenbanken von hunderten Flussdatensätzen spezifische Metriken sowie Visualisierungs- und Interaktionstechniken, um den Anwender effizient bei der Hypothesengenerierung zu unterstützen.

Diese Doktorarbeit präsentiert und evaluiert Techniken, um effektive visuelle Exploration und Vergleiche von Blutflussdaten mit dem Ziel, neue Erkenntnisse über das menschliche kardiovaskuläre System zu gewinnen, zu ermöglichen. Dies umfasst sowohl allgemeine Visualisierungs- und Verarbeitungstechniken (beispielsweise um die Tiefenwahrnehmung zu verbessern), als auch auf die Beantwortung spezifischer medizinischer Forschungsfragen zugeschnittene Lösungen. Unser Fokus liegt hierbei auf Blutflussdaten, die mittels 4D PC-MRI oder hämodynamischen Simulationen generiert wurden. Obwohl signifikante Unterschiede zwischen gemessenen und simulierten Blutflussdaten existieren, wurden die im Rahmen dieser Dissertation implementierten Techniken in einem gemeinsamen Framework integriert.

Introduction 1.3

According to the World Health Organization, cardiovascular diseases are the leading cause of death worldwide [1]. Especially arteriosclerosis and its accompanying diseases, such as myocardial infarction or thrombosis, may lead to severe complications [2]. Various cardiovascular pathologies have been related to changes in blood flow patterns [3, 4]. For example, defects such as a bicuspid aortic valve often cause vortices [5, 6] and increased wall shear stress [7] within the thoracic aorta. In turn, changes in blood flow patterns may cause the formation of aneurysms, which are pathologic dilations of the arterial wall.

Aortic aneurysms can ultimately lead to aortic dissection or rupture, with a reported risk of rupture of up to 74 % for the thoracic aorta [8, 9], depending on the size of the aneurysm [10]. The rupture of a thoracic aortic aneurysm is almost always fatal [11]. Without treatment, aneurysms of the thoracic aorta with a diameter greater than *6 cm* exhibit a yearly death rate of almost 12 % and a five-year survival rate around 55 % [10, 12]. Although the surgical repair of aneurysms on the thoracic aorta itself has a mortality between 3 % and 11 %, it can significantly improve the long-term survival rate of a patient [12].

In case of cerebral aneurysms, a rupture may cause an ischemic stroke with often fatal consequences for the patient. Especially with respect to increased imaging in the clinical routine, more and more asymptomatical cerebral aneurysms are detected [13]. Since aneurysm treatment, e.g. endovascular stenting and coiling or neurosurgical clipping, is accompanied with risks for the patient, clinical research aims to avoid unnecessary treatment of incidentally found aneurysms and to identify the best possible treatment with respect to patient-specific anatomy and blood flow behavior. For rupture risk assessment, several clinical risk factors have been identified, e.g. the patient's age, hypertension, aneurysm size and location [14]. Cerebral aneurysms often comprise a daughter aneurysm, also called a *bleb*, which is a prominent bulging on the aneurysm. These blebs are of particular interest, since they have been identified as an additional factor for increased rupture risk [15, 16]. State-of-the-art rupture risk models combine morphological parameters characterizing the aneurysm's shape as well as hemodynamical parameters extracted from simulated blood flow to provide further information about the patient-specific risk [17]. The simulated blood flow not only allows for the extraction of flow patterns, but for the extraction of parameters related to pressure and stress as well, e.g. the wall shear stress (WSS), describing the friction of the blood flow on the aneurysm surface, or the oscillatory shear index (OSI), characterizing the variation of the WSS vector over the course of the heart cycle.

For endovascular treatment support, clinical researchers are strongly interested in predicting flow characteristics after treatment. Hence, the flow patterns and parameters provide valuable information about the intra-aneurysmal flow [18]. Implant placement typically aims at aneurysm occlusion, a redirection of the blood flow causing the aneurysm as well as a stabilization of the blood supply of the vessel harboring the aneurysm and its outlets. Virtual implant placement can support the clinician by predicting the resulting blood flow [19, 20].

Motivation

The evaluation of blood flow data is increasingly used in medical research to improve the understanding of vascular diseases as well as for treatment planning. An example is the placement of vascular implants to treat aneurysms, which strongly depends on the patient-specific anatomy and hemodynamics. Thus, the visualization of hemodynamic information can support medical research with respect to the understanding of vascular diseases and improvements in patient-specific treatment planning. Questions relate to the interactions between hemodynamic flow patterns, morphological changes and vascular biology [21].

Commonly, 3D blood flow data visualizations are comprised of either path- or streamlines embedded in a surface visualization of the vessel. Unlike other integral lines, such as streak- or timelines, stream- and pathlines directly represent the movement of blood through the vessels. Attributes, such as pressure, wall shear stress and flow velocity magnitude, can be mapped to the surface or pathlines using color scales. Often, interesting flow structures such as vortices are nested within laminar flow, creating visual obstructions. Having too many pathlines visible at once will increase this effect, while having too little pathline coverage may cause specific flow characteristics to be underrepresented in the resulting visualization. Hence, specialized techniques are required to focus on specific hemodynamic phenomena, such as vortices, in order to alleviate occlusion while maintaining dense line coverage in areas of interest. These techniques can range from *filtering* occluding pathlines to dynamically *adding* additional pathlines to increase the pathline coverage for interesting features. In addition to pathlines, the vessel surface itself provides valuable context information and therefore should not be culled from the visualization completely.

As a result, the scene contains many intertwined layers of pathlines and surfaces. One of the more general challenges in visualizing blood flow data is mapping this complex and highly patient-specific three-dimensional vascular structure onto a two-dimensional screen while maintaining spatial awareness of the user. Similarly, establishing comparability between multiple patient-specific variations of the same anatomical region poses a challenge.

Thesis Structure and Contribution

This thesis addresses the aforementioned challenges and proposes solutions to support medical researchers. All the techniques developed as part of this PhD thesis are integrated into a single interactive framework supporting the visual analysis of both measured and simulated blood flow data. Due to the stark differences between simulated and measured blood flow regarding acquisition modalities, data format and general data quality, some of these techniques proposed in this thesis are tailored specifically towards 4D PC-MRI. However, many general properties are shared between these types of flow data, thus allowing many of the presented techniques to be applied to both measured and simulated blood flow data with only minimal adjustments. This thesis is structured as follows:

- Chapter 2 gives a brief overview of the human cardiovascular system and selected related pathologies. Furthermore, core principles of flow visualization and the imaging modalities used to acquire the medical images we base our techniques on are explained.

- Chapter 3 summarizes existing techniques and approaches to visualize, explore and analyze blood flow data. It also gives a brief overview of previous work within the Visualization group on the topic of visualizing hemodynamic data.

- Chapter 4 describes the main contributions of this thesis. At the beginning of each section, the publications on which the respective section is based on are listed. Changes that were made to the presented techniques after the publication are also mentioned there.

  Section 4.1 proposes an image processing technique to support the *segmentation of low-contrast 4D PC-MRI datasets*. Strongly varying flow velocities in the aorta, which can appear as a result of stenoses or defects of the aortic valve, negatively affect the contrast of 4D PC-MRI data. These low-contrast images are often hard to segment, especially when directional information is discarded for the segmentation by performing it entirely on magnitude images. Thus, we propose to use a combination of magnitude and FTLE images, which incorporate directional information. We were able to show that, in comparison to a solely magnitude-based segmentation approach, our method requires less manual input and produces more satisfying results for low-contrast datasets.

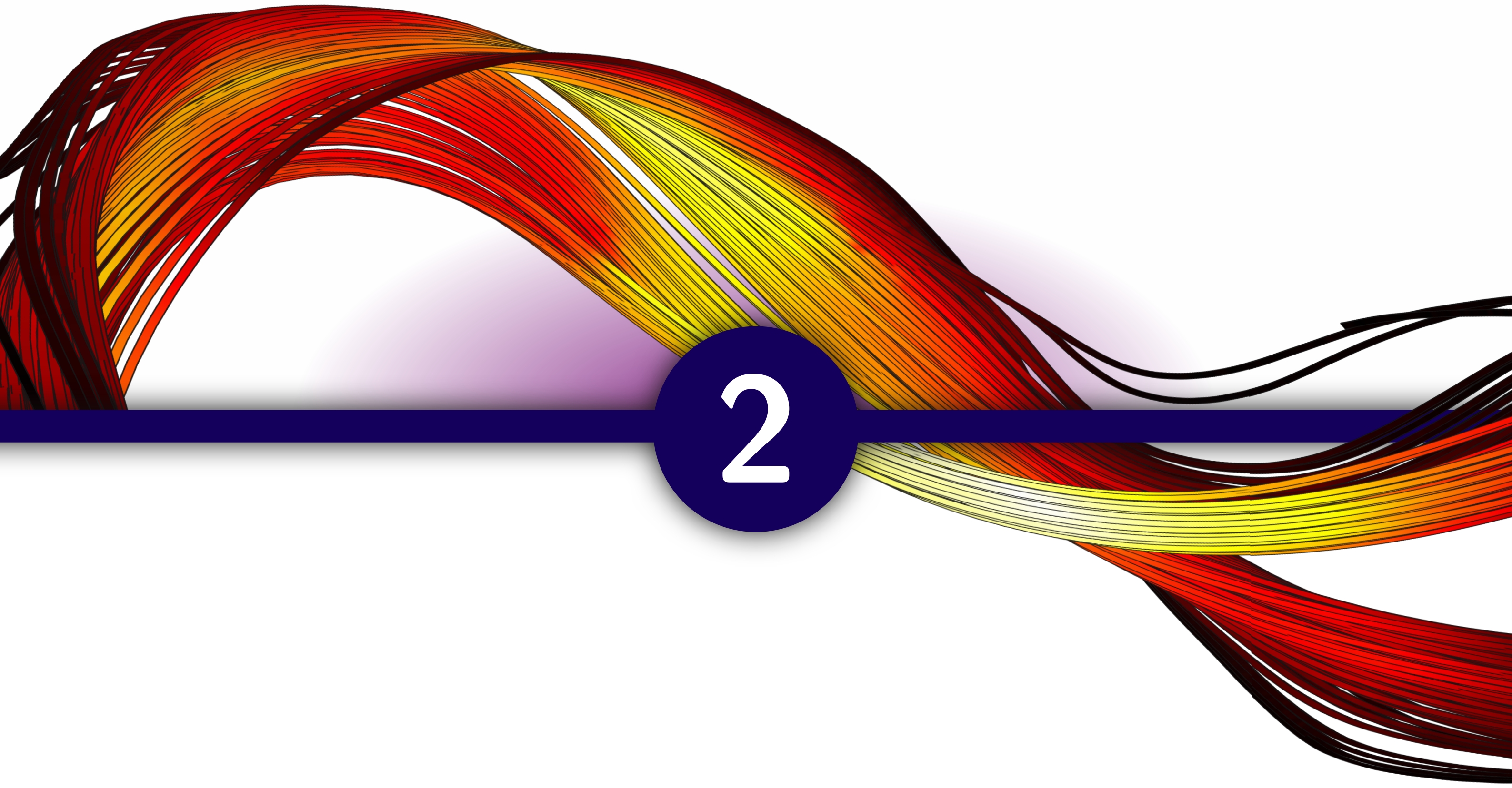
  Section 4.2 focuses on *optimizing the perceptibility of vessels and the inlaying blood flow*. As mentioned previously, blood flow visualizations on a traditional 2D screen suffer from occlusion and a lack of depth perception. We present a technique to solve overlaps between vessel wall and pathlines for direct volume rendering, similar to front-face culling. Based on the current viewing direction, the back side of vessels is identified. The front side and lumen are culled from the visualization, exposing pathlines drawn within the vessel.
  Additionally, we present a shading technique that increases depth perception for vascular models without impairing the ability to map hemodynamic parameters onto the vessel surface. Our approach overlays a pseudo-chromadepth color scale to encode depth at the edges of the vascular surface and an additional color scale representing hemodynamic parameters at the center. In a study with 105 participants, we were able to show that our technique increases depth perception while only slightly impairing the ability to perceive the hemodynamic color scale.

  Section 4.3 presents several approaches to support *exploration and comparison of blood flow datasets*. Hemodynamic parameters on the vessel surface are often directly connected to underlying near-wall flow. As our clinical collaborators are interested in this connection, we developed a technique to interactively select hemodynamic features on the vessel surface and extract related flow structures with minimal interaction. In addition to the qualitative assessment of the resulting flow structures, our prototype allows to quantify them and export the findings within structured reports.
  A common problem in several of our datasets was low pathline density in regions with low blood flow activity, e.g. blebs, as the pathline integrator had trouble “reaching” these regions. This situation could partially be alleviated by increasing the amount of seed points, which would significantly increase computation time and also produce many redundant pathlines in other regions. Thus, we developed a seeding strategy based on evolutionary algorithms, which would directly target specific regions of interest and ensure that flow in these regions was sufficiently represented by pathlines. We were able to show that our seeding approach was able to reach a better line coverage than the commonly used uniform seeding with a lower number of total line integrations.
  Commonly, the quantification of hemodynamic parameters for an aneurysm is tied

to the location of its neck region. However, no general consensus exists between practitioners on how exactly the location of the neck curve is defined. To support making steps towards such a consensus, we implemented a web-based application for medical experts to easily draw neck curves based on their own definitions. Additionally, they are able to deform the resulting ostium to model the original vessel surface without the aneurysm. The resulting neck curve segmentations are then stored in a database to be used as ground truth data for ongoing research.

As previously mentioned, finding systematic differences between pathological and healthy flow is an important part of gaining a better understanding of vascular pathologies. A simple juxtaposition of datasets is unsuitable due to the complex vascular structures and patient-specific vessel morphology. Thus, we present an approach to normalize the spatio-temporal domain of cardiac 4D PC-MRI datasets and facilitate a comparison using 2D bull's eye plots. For a more in-depth analysis of flow differences, a visual comparison of slices at normalized positions is also possible.

- Chapter 5 concludes this thesis and discusses both possible short- and long-term research projects in relation to this work, and blood flow visualization in general.



2 Background

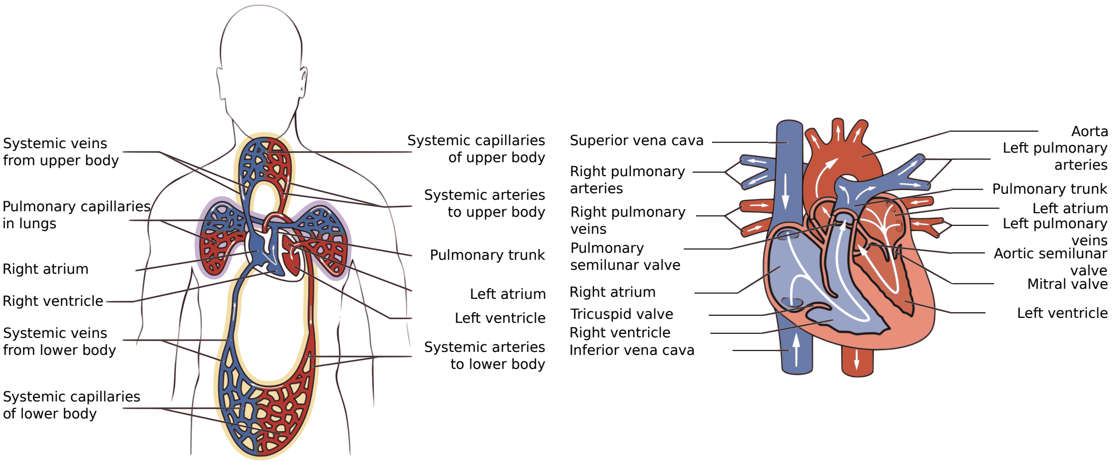
Medical Background 2.1

This section presents anatomical background information about the human cardiovascular system in general and selected cardiac and cerebral blood vessels in particular. Relevant cardiovascular diseases affecting these systems are also introduced.

The Human Cardiovascular System 2.1.1

The cardiovascular system is comprised of blood vessels spanning the entire body, providing all cells with nutrients and oxygen via the blood stream. There are two distinct blood circuits that make up the cardiovascular system [23]. The *systemic circuit* provides oxygen to the tissue, transferring oxygenated blood from the heart via systemic arteries through the body. After the oxygen has been passed to the cells within systemic capillaries, the now de-oxygenated blood travels back to the heart through systemic veins. From there it enters the *pulmonary circuit*, where pulmonary arteries transport the blood to the lung to gather new oxygen within the pulmonary capillaries. The newly oxygenated blood moves back to the heart via pulmonary veins and once again enters the systemic circuit [22]. A schematic overview of both circuits can be found in Figure 2.1a.

The central organ of both circuits is the heart (Figure 2.1b), a pump that facilitates the blood movement. It consists of two sides (*left* and *right*), each with two chambers (*atrium* and *ventricle*). The atria are connected to their respective ventricle through a valve (the *tricuspid valve* on the right side of the heart, the *bicuspid* or *mitral valve* on the left), preventing blood to flow from the ventricle back into the atrium [23]. The left ventricle is connected to the *aorta*, the right ventricle to the *pulmonary artery*. Once again, the direction of blood flow is controlled by another set of valves (*aortic semilunar valve* and *pulmonary semilunar valve*). All valves in the heart consist of tree cusps, with the exception of the mitral valve, which only consists of two. They are not



(a) (b)

Figure 2.1: *Schematic illustration of the human cardiovascular system (a) and heart (b), showing oxygenated blood in red and de-oxygenated blood in blue [22].*

Both images are taken from the OpenStax publication "Anatomy and Physiology", released under the Creative Commons Attribution 4.0 license. They were slightly modified to better fit the layout of this thesis. ©1999–2021, Rice University. URL: https://openstax.org/books/anatomy-and-physiology/pages/19-1-heart-anatomy#fig-ch20_01_03 License: https://creativecommons.org/licenses/by/4.0/deed actively controlled by muscles, instead they open and close entirely based on pressure differences [23].

Each heart cycle consists of two phases, the *systolic phase* and *diastolic phase*. During the systole, both ventricles contract, thus pumping blood out of the heart and into the aorta (from the left ventricle) and pulmonary artery (from the right ventricle). During the diastole, both the ventricles and atria relax, allowing new blood to flow into the atria from the *pulmonary veins* (into the left atrium) and *inferior* and *superior vena cava* (into the right atrium). Afterwards, the atria contract to pump the blood into their respective ventricles. From there, the cycle begins anew [22].

As the starting point of the systemic cycle, the aorta is the largest artery in the human body [24]. Its overall shape somewhat resembles that of a walking cane and it is commonly subdivided into the *aortic trunk* (where it is connected to the left ventricle), followed by *ascending aorta*, *aortic arch* and *descending aorta*. Three major branches spring from the aortic arc: the *brachiocephalic artery*, *left common carotid artery* and *left subclavian artery* [22]. The brachiocephalic artery itself branches into the *right common carotid artery* and *right subclavian artery*. Each of the subclavian arteries spawns a *vertebral artery*. Both the carotid and vertebral arteries provide the blood supply to the neck and head, whereas the subclavian arteries supply the chest and arms. The carotid arteries eventually split into an *external carotid artery* and *internal carotid artery*. The vertebral arteries, on the other hand, merge together and form the *basilar artery* [22]. An overview of the branching of the aorta can be found in Figure 2.2. Studies have shown that this branching structure of the aortic arch is present in around 74% of the western population. A common variant with a prevalence of 20%, often referred to as *bovine arch*, has the brachiocephalic artery and left common carotid artery originating from the same point on the aortic arch [25].

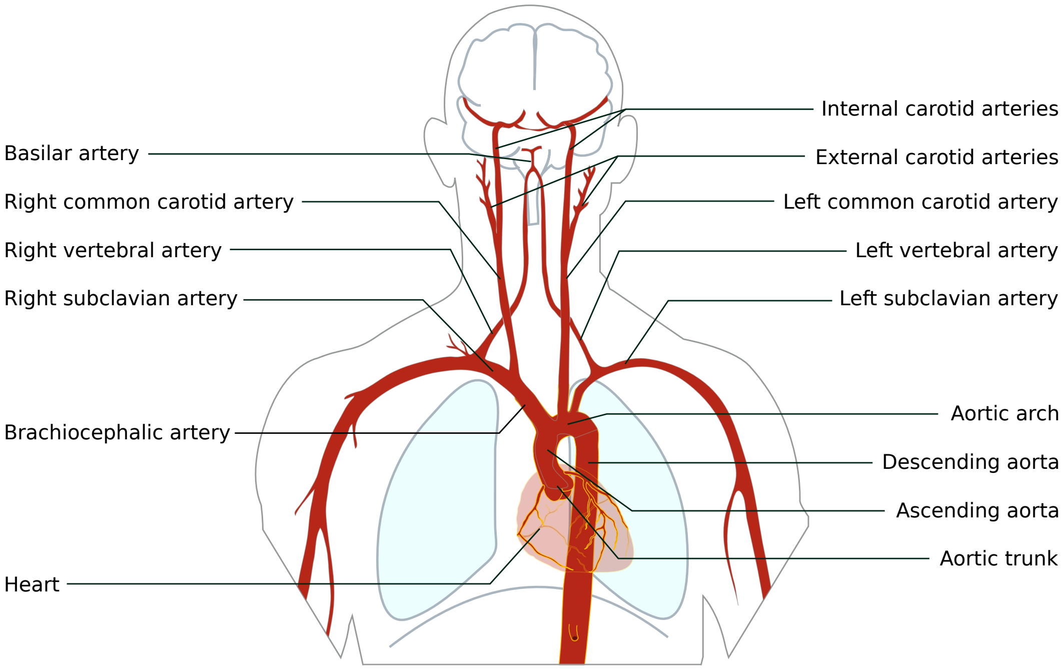
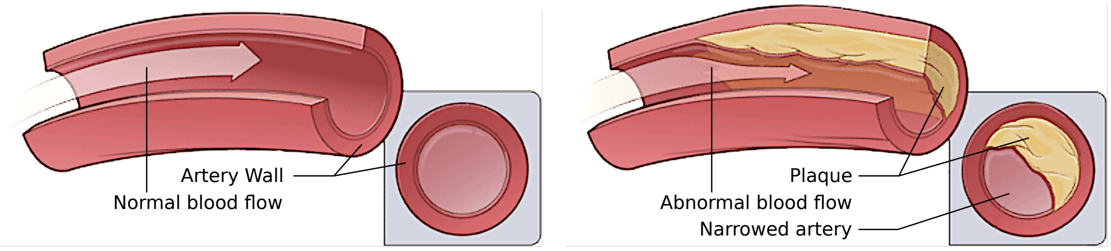


Figure 2.2: Schematic illustration of the aortic branches.

This figure is based on an image by Edoarado, made available via Wikimedia Commons under the Creative Commons Attribution-Share Alike 3.0 Unported license. The labels and some of the coloring were modified to better fit the wording and layout of this thesis. URL: https://commons.wikimedia.org/wiki/File:Aorta_scheme_en.svg License: https://creativecommons.org/licenses/by-sa/3.0/deed



(a) (b)

Figure 2.3: *Illustration of a normal artery (a) and an artery affected by plaque buildup, causing the artery to narrow (b) [29].*

The images are taken from the NHLBI article on atherosclerosis and are released in the Public Domain. Slight modifications have been made to the image layout and descriptions.

The internal carotid arteries and the basilar artery supply the brain and eventually lead into the *circulus arteriosus cerebri*, often referred to as the “Circle of Willis”. The Circle of Willis is a vaguely circular vascular structure connecting the internal carotids and basilar artery and thus creating redundancy for the brain’s blood supply. If the blood flow from one of the supplying vessels is progressively blocked, e.g. by a stenosis, the other vessels can compensate this blockage. A sudden and complete occlusion of one of the vessels, however, can usually not be fully compensated for. The Circle of Willis is a highly variable structure, as depending on the person, various connecting vessels within the circle may be reduced in size or missing completely [26].

Cardiovascular Diseases 2.1.2

With social, medical and technological advancements, the typical disease-related causes of death shift from infectious diseases to degenerative diseases [27]. Today, according to the World Health Organisation, cardiovascular diseases (*CVDs*) are the most common cause of death in most regions, accounting for around 25% of all deaths worldwide [28]. This section describes a selection of common cardiovascular diseases as well as their effect on hemodynamics.

Pathologic changes in the vessel wall

Changes in the vessel wall may cause the vessel diameter to increase or decrease. Generally, a narrowing of a blood vessel is called *stenosis*. A common cause for stenosis is *atherosclerosis*, whereas so-called *plaques* consisting of “fatty deposits, inflammation, cells, and scar tissue” [30], are deposited within the vessel wall (Figure 2.3). The narrowing of the vessel may cause blood pressure and flow speed in the affected area to increase. Furthermore, a more severe narrowing may decrease the amount of blood supplied to the surrounding tissue or completely stop the blood flow altogether. This can lead to potentially fatal consequences, such as *stroke* or *myocardial infarction*.

An increase in vessel diameter is referred to as an *aneurysm*. Aneurysms bear a risk of rupturing, causing potentially fatal internal bleeding. Generally, aneurysms can be classified as either saccular or fusiform (Figure 2.4). Saccular aneurysms appear as *sacs* or *pouches* on the parent vessel and result from the dilation of only part of the vessel wall (Figure 2.4a). Fusiform aneurysms, on the other hand, are formed when an entire section of the vessel wall dilates (Figure 2.4b).

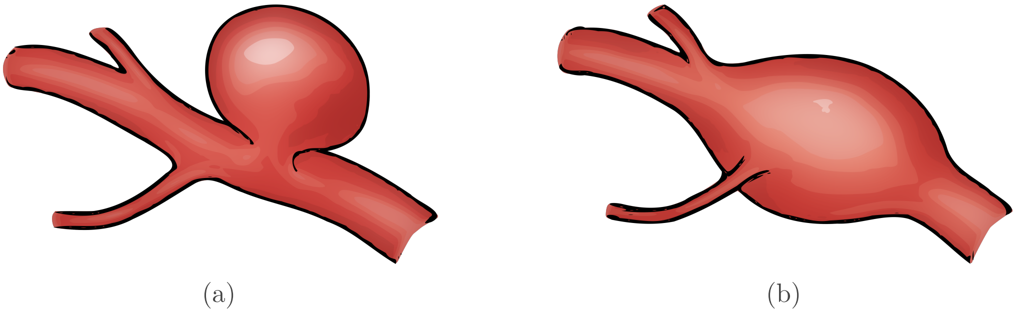


Figure 2.4: Sketches of the two main aneurysm types: Saccular aneurysms (a) and fusiform aneurysms (b) [31].

The images are taken from an open-access paper by Withers et al., which is published under the Creative Commons Attribution-NonCommercial 1.0 Generic license. Modifications have been made to the image layout and coloring. ©The Author(s) 2013. License: https://creativecommons.org/licenses/by-nc/1.0

The prevalence and rupture fatality depends strongly on the location of the aneurysm. Coronary artery aneurysms are most commonly caused by atherosclerosis and bear a prevalence between 1.5-5 % [32]. Thoracic aortic aneurysms, e.g. aneurysms of the aorta within the thoracic region, are mostly caused by medial degeneration, which weakens the aortic wall and thus leads to dilation [33]. After reaching a diameter of *6 cm* or greater, the yearly mortality for thoracic aneurysms reaches around 11% [12, 34]. While some amount of medial degeneration results simply from aging, the process can be accelerated by hypertension or defects of the aortic valve. Another cause for medial degeneration is the *marfan syndrome*, a genetic disorder that affects connective tissue all over the body [33].

Aneurysms on cerebral vessels show a high prevalence in the western population (3-5 %) [35], while their annual risk of rupture is below 1 % [36]. The bleeding caused by their rupture can have fatal consequences, with 40 - 60 % of the patients dying within the first 30 days. However, especially in the case of small, asymptomatic cerebral aneurysms, the mortality rate of the treatment may actually exceed the risk of rupture [37]. In clinical practice, the most important rupture risk factors are the type of aneurysm (i.e., asymptomatic or symptomatic), age, sex, and aneurysm size and position [38]. Furthermore, morphological parameters such as irregular shape, orientation and diameter [39, 40] were correlated with rupture risk. Studies have shown that hemodynamic parameters, such as changes in pressure or wall shear stress, on the vessel wall, correlate with the rupture of aneurysms and are therefore vital for risk assessment [41–43]. Thus, the study of these hemodynamics plays an increasing role in current research [44].

Valve Pathologies

Not only the blood vessels can be affected by diseases, but so can the valves. Their purpose is to prevent blood from flowing in the wrong direction, for example from the aorta back into the ventricle. The failure of a valve to properly close (called *insufficiency*) causes reverse flow (*regurgitation*) and thus adversely affects the cardiac function by reducing the pumping efficiency of the heart. The most common causes for aortic valve insufficiency are inflammations and dilatation of the aortic root [46].

Similar to vessels, it is also possible for a valve itself to be *stenotic*, i.e. narrowed, making it harder for the heart to pump blood through the valve. Such a narrowing is regarded as a severe stenosis of the aortic valve (also commonly referenced to as *aortic*

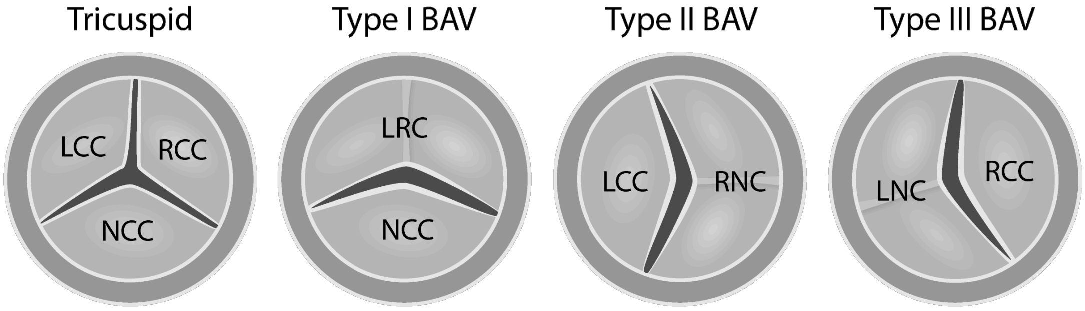


Figure 2.5: *Different types of BAVs in comparison to a tricuspid aortic valve [45].*

The image is taken from an open access article by Martin et al., published under the Creative Commons Attribution 4.0 International license. ©2015 by the authors. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5438177/ License: https://creativecommons.org/licenses/by/4.0

stenosis or *AS*) when the opening area of the valve is reduced to less than one fourth of its original size [47]. At this point, significant hemodynamic changes such as increased pressure gradients start to appear. In industrialized countries, aortic stenosis mostly appears as a degenerative disease [48] and thus typically affects older patients (with a prevalence of 1-2% between the ages of 64 to 74, and 4-5% above 74) [49]. In developing countries, it mostly results from rheumatic infections [48].

Aortic stenosis can also be congenital [50], with the most common form being a *bicuspid aortic valve*, or *BAV*. With a prevalence of around 1-2%, it is the most common congenital cardiac defect [51]. In a BAV, two of the three cusps are fused together, usually resulting in a valve with two differently sized cusps [52]. BAVs are classified into three different types depending on which of the cusps are fused together (Figure 2.5). Type 1 is the most common form, accounting for more than 70 % of BAV patients, followed by Type 2 (10-20 %) and Type 3 (5-10 %) [53]. BAVs have been connected to various complications, such as dilatations of the aortic root [54], stenois in the aortic arch and regurgitation [46, 51]. Even in the case of an initially asymptomatic BAV, the valve is more prone to degenerative aortic valve disease due to the higher amount of mechanical stress [55].

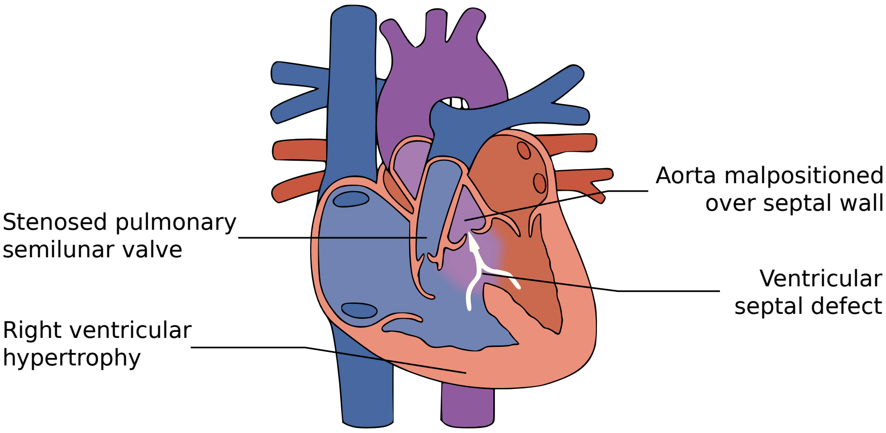

Stenosed pulmonary semilunar valve
Right ventricular hypertrophy
Aorta malpositioned over septal wall
Ventricular septal defect

Figure 2.6: *Illustration of the anatomical changes within the heart caused by a Tetralogy of Fallot [22].*

The image was taken from the OpenStax publication "Anatomy and Physiology", released under the Creative Commons Attribution 4.0 license. It was cropped and the labels were slightly altered to be in line with the terminology used in this chapter. ©1999–2021, Rice University. URL: https://openstax.org/books/anatomy-and-physiology/pages/19-1-heart-anatomy#fig-ch20_01_09 License: https://creativecommons.org/licenses/by/4.0/deed

Pathologies of the Heart

Congestive heart failure (CHF) describes the general inability of the heart to provide adequate blood supply to the body and can be a result of various cardiovascular diseases. In many cases, a precursor to CHF is hypertension or coronary heart disease [56]. The general prevalence of CHF is around 2 %, although the prevalence increases with age [57].

The *tetralogy of fallot* (TOF) is the most common congenital heart disease and consists of multiple pathologic alterations of the cardiac anatomy. A *ventricular septal defect* allows blood to flow directly from the right into the left ventricle (*right-to-left shunt*). As a result, some amount of de-oxigenated blood is pumped into the systemic circuit via the aorta instead of the pulmonary circuit, reducing the overall oxygen level in the systemic circuit. This effect is increased by a pulmonary stenosis and a malpositioning of the aorta over the ventricular septal wall instead of the left ventricle. The increased effort to pump blood through the pulmonary artery leads to an enlargement of the muscles around the right ventricle (*right ventricular hypertrophy*). Usually, a TOF is diagnosed and surgically corrected a few months after birth [58].

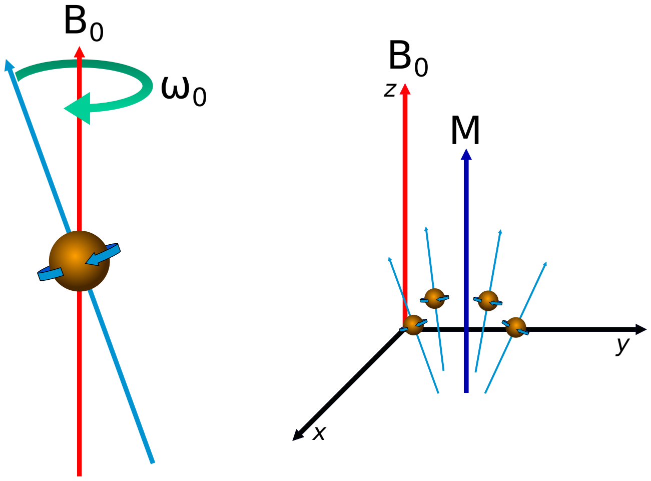
Technical Background 2.2

In current research, several cardiovascular pathologies have been related to blood flow pattern changes [3, 59–61]. This section describes the technical aspects of blood flow data acquisition, as well as the general concepts and challenges for visualizing the resulting data.

Cardiac Blood Flow Acquisition using 4D PC-MRI 2.2.1

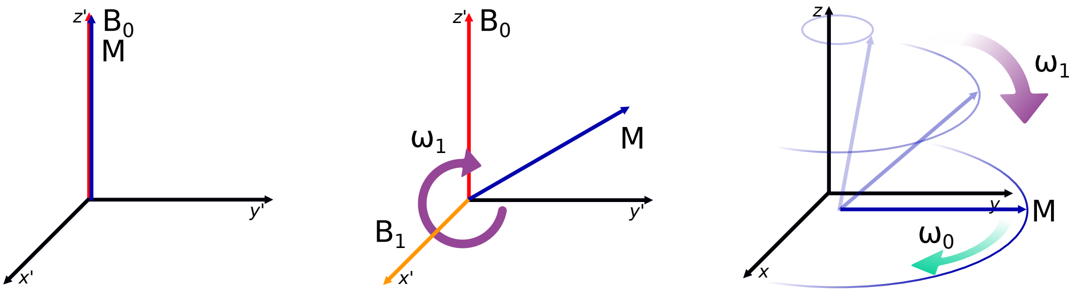
Magnetic resonance imaging (*MRI*) is a widely available, non-invasive imaging modality used in a variety of different medical applications. Magnetic fields are used for the image acquisition which, unlike the ionizing radiation used for computer tomography (*CT*) scans, have no known adverse effects on living tissue. The following section aims at providing a general overview over the physical properties and processes exploited for the image acquisition. MRI is based on the *spin*, a rotational movement of atomic nuclei. Under normal circumstances, the axis of this rotation is arbitrary. When a strong magnetic field B_0 is introduced, the spin axes of the protons within the field align with the field direction ($\vec{B_0}$), causing a *magnetization* proportional to the density of protons within the tissue and aligned to $\vec{B_0}$ (Figure 2.7) [62]. However, the individual rotational axes are not perfectly aligned to $\vec{B_0}$ and instead rotate (*precess*) around it (Figure 2.7a). The rotation frequency ω_0, called *Larmor frequency*, depends on the type of nuclei and field strength of B_0. In case of a single proton in a magnetic field B_0 with a strength of *1 T*, it is *42.6 MHz*.

By introducing a second magnetic field B_1 in the form of a radio frequency (*RF*) signal, the magnetization direction ($\vec{M}$) can be thrown out of alignment with $\vec{B_0}$ (Figure 2.8). Specifically, if the field direction $\vec{B_1}$ is perpendicular to $\vec{B_0}$ and rotates around it with a frequency of ω_0, $\vec{M}$ precesses with a Larmor frequency of ω_1, based on the field strength of B_1. To illustrate this process, we construct a rotating coordinate system, whereas the z' axis is aligned with $\vec{B_0}$ and the $x'y'$ plane rotates with a frequency of ω_0. Before the introduction of B_1, $\vec{M}$ aligns with $\vec{B_0}$ and the z' axis (Figure 2.8a). As $\vec{B_1}$ is



(a) (b)

Figure 2.7: *Alignment of a proton's rotation ω_0 to the magnetic field B_0 (a); overall magnetization direction M for multiple protons within B_0 (b).*



(a) (b) (c)

Figure 2.8: *Magnetization direction M in the rotation coordinate system $x'y'z'$ before (a) and after (b) the introduction of the second magnetic field B_1. This results in a composite rotation of M around z with a frequency of ω_0 and around x with a frequency of ω_1 in the static coordinate system xyz.*

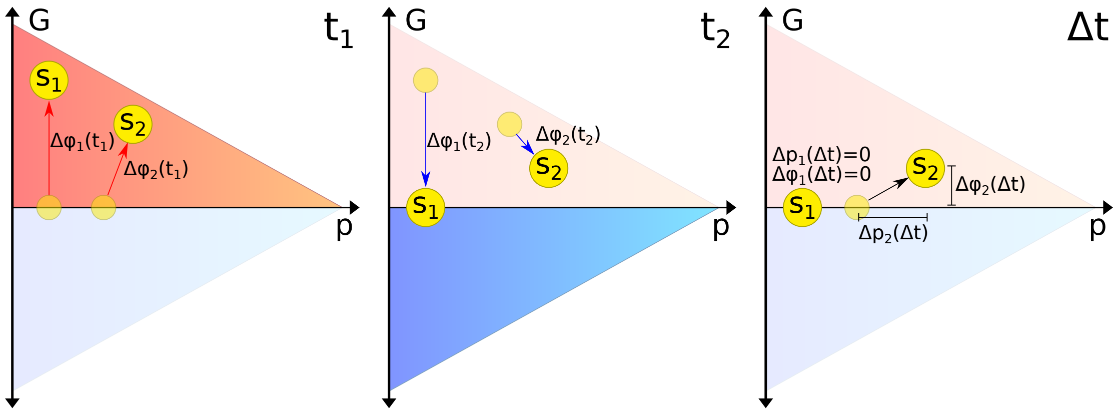
perpendicular to $\vec{B}_0$ and rotating with the same frequency as our coordinate system, it appears as a static vector in it. For the sake of demonstration, we will assume that it aligns with the x' axis (Figure 2.8b). Due to the influence of B_1, $\vec{M}$ now precesses around the x' axis of our rotating coordinate system with a frequency of ω_1. After a certain amount of time, depending on ω_1, $\vec{M}$ will have rotated by 90 ° and reaches the y' axis. At this point, $\vec{M}$ rotates perpendicular around $\vec{B}_0$ with a frequency of ω_1 in the static coordinate system (Figure 2.8c). Using a coil that *resonates* with magnetization perpendicular to $\vec{B}_0$, a current can be induced that is the signal used to construct the final image. This signal decays over time, as the individual proton spins de-synchronize or *de-phase* due to the influence of magnetic fields of other molecules (T_2 decay). The time T_2 for this to happen depends on the type of tissue. When B_1 is turned off, $\vec{M}$ realigns with $\vec{B}_0$ within a time T_1, which is also dependent on the type of tissue, but is not directly related to T_2. Depending on the tissue the physician wants to examine, the image acquisition parameters of the MR scanner can be adjusted to produce T_1- or T_2-weighted images.

With the previously described methods, we can now detect the presence of different tissue types based on their respective T_1 or T_2 time, but for the construction of an image, additional spatial information is required [62]. To only measure tissue in a specific slice, a magnetic field with a linear gradient in the same direction as B_0 and increasing in z-direction can be overlayed. This produces a Lamor frequency ω_0' specific to each slice position, as the frequency is directly dependent on the field strength of B_0. By modulating the rotation speed of B_1 to fit a specific ω_0', only protons within this *selected slice* resonate with the coil. Multiple slices can be acquired by repeated measurements for different ω_0' frequencies. A similar process allows selecting specific positions in x-direction, using an additional gradient field. For selecting a position in y-direction, a different technique is employed. A third gradient field is introduced, where the strength increases in y-direction. However, unlike the x and z gradient fields, its field direction is not aligned to B_0 and the z axis, but instead to the y axis. Additionally, this field is only active for a short time after B_1 is turned off. This causes a phase shift within the precession of protons depending on their y-position. By performing multiple measurements, the individual measurements for each y-position can be calculated from the sums of all y-positions. With this procedure, a three-dimensional data volume can be constructed.

Phase Contrast (PC)-MRI is a specific MRI scan sequence first applied to humans in the 1980s [63], with the principle being established in 1954 [64]. It allows for the in-vivo acquisition of blood flow information in the vascular system [65] and can be applied to a variety of different vascular structures, including the left and right ventricle and even various neurovascular regions [66]. In clinical practice, only 2D PC-MRI (which describes the acquisition of a time-resolved 2D slice) is commonly used [67]. In current clinical research, however, 4D PC-MRI (time-resolved 3D volumes) plays an increasingly important role [68]. The image acquisition process is similar to the one previously described, with the addition of two bipolar gradient fields. These fields are identical in strength, but have opposite directions (Figure 2.9). Their purpose is to alter the phase of spinning protons (Figure 2.9a). For static tissue, the phase shift of the first field is undone by the second field (Figure 2.9a). Moving tissue, however, will retain a certain amount of phase shift due to changing position between the activation of the first and second fields (Figure 2.9b). As this shift is proportional to the movement speed, the speed can be reconstructed from it (Figure 2.9c). This process is referred to as velocity encoding (V_{enc}). However, if the final phase shift after the application of both gradient fields exceeds the range $[-\pi, \pi]$, the actual speed can no longer be determined reliably [62]. For example, as phase shift of 200 ° is indistinguishable from a phase shift of -120 °.

Data acquired using 4D PC-MRI consists of three flow images representing direction and three magnitude images representing speed for each slice and time step (Figure 2.10). From the flow images, a four-dimensional velocity vector field can be constructed. The pathlines can be integrated from this field, i.e. using fourth-order Runge-Kutta ($RK4$) integration [69]. The magnitude images can be used for segmentation purposes, as they exhibit higher contrast and are less susceptible to noise. The datasets also contain the patient coordinate matrix, which represents a transformation matrix to translate voxel coordinates from the dataset into world coordinates, including both position and orientation of the patient.

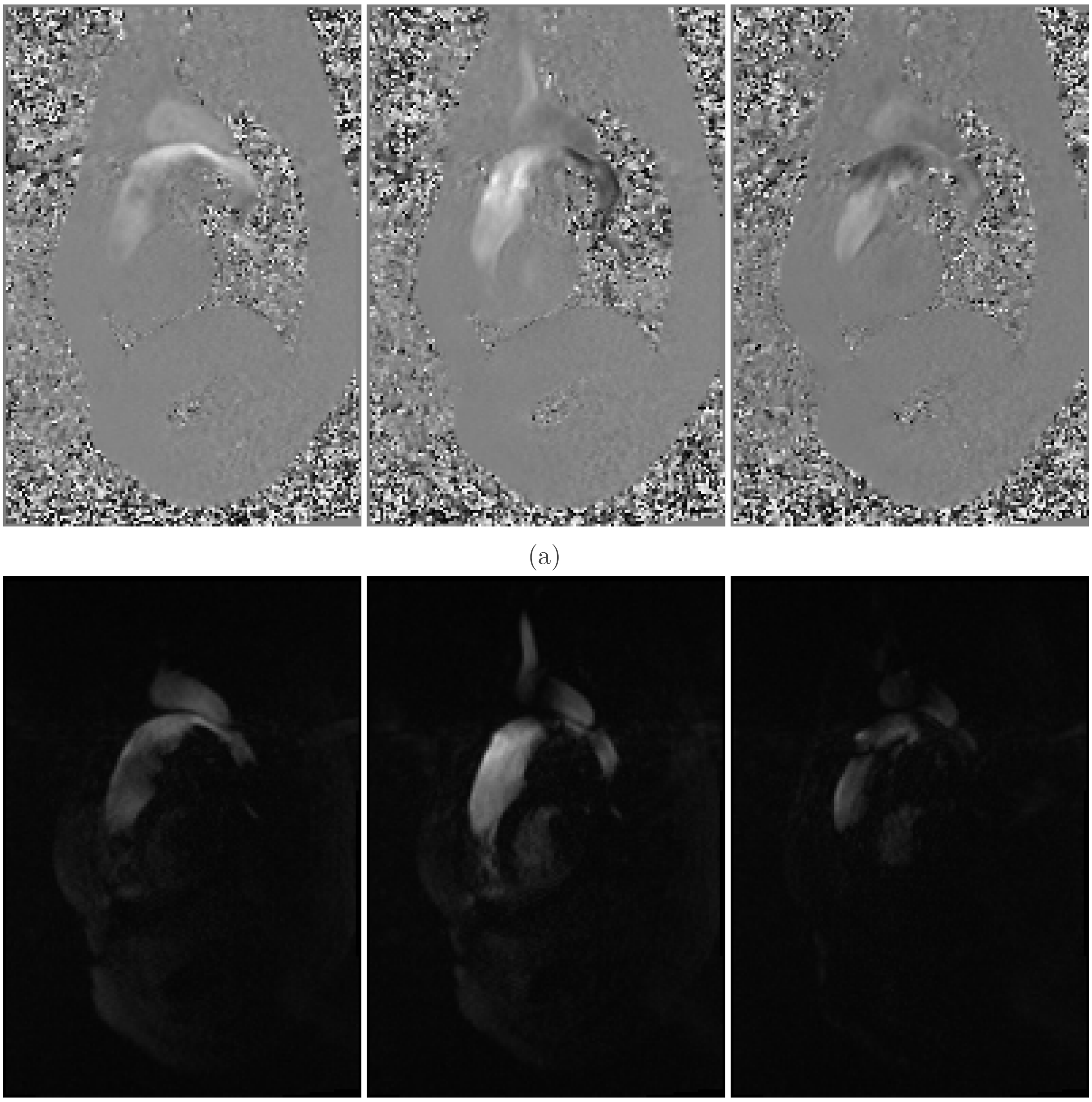
The acquisition requires configuring the V_{enc}, denoting the highest expected velocity in x, y and z-direction. Values between 150 and 200cm/s are commonly used for acquisitions of the thoracic aorta [66], although the flow velocity in pathological cases may exceed this range. As mentioned before, if the actual flow velocity exceeds the configured V_{enc}, the



(a) (b) (c)

Figure 2.9: Effect of two bipolar gradient fields on static (s_1) and moving (s_2) tissue particles; the first gradient field (red) shifts the phase of both particles according to their position p at time t_1 (a). When the first field is turned off and the second field (blue) turned on, the phase shift is fully reversed for s_1, but not for s_2, as it is subjected to a different field strength at the new position (b). The movement Δp of s_2 and can be reconstructed from the time difference Δt and phase shift $\Delta\varphi$.

acquired value for the respective direction flips its algebraic sign and the flow therefore appears as if it were moving in the opposite direction. These artifacts are called *phase wraps* and manifest either as white areas surrounded by black or black areas surrounded by white in individual slice images (Figure 2.11). Increasing the V_{enc}will prevent these artifacts, but also lower both the overall image contrast and signal-to-noise ration, thus increasing the difficulty of segmenting the anatomy and decreasing the accuracy of the flow data [70]. Cases with areas of both very high (e.g. inflow jets) and low (e.g. parts of the aortic arch) flow speed, which are often clinically interesting, can therefore pose problems [71]. Here, the V_{enc}has to be increased in order to prevent phase wrapping artifacts in certain regions, therefore lowering contrast in others. When not using direct volume rendering to visualize the data, a surface mesh needs to be constructed from the image segmentation. As 4D PC-MRI images often suffer from poor contrast due to the V_{enc}, the segmentation is usually performed on a temporal maximum intensity projection (*TMIP*) of the magnitude images [72]. In this TMIP, voxels containing high flow velocity in any time step (i.e. voxels within the vessel) are highlighted.


(a)

(b)

Figure 2.10: *Single slices from the flow (a) and magnitude (b) time-resolved volume images, representing flow in x, y and z-direction.*

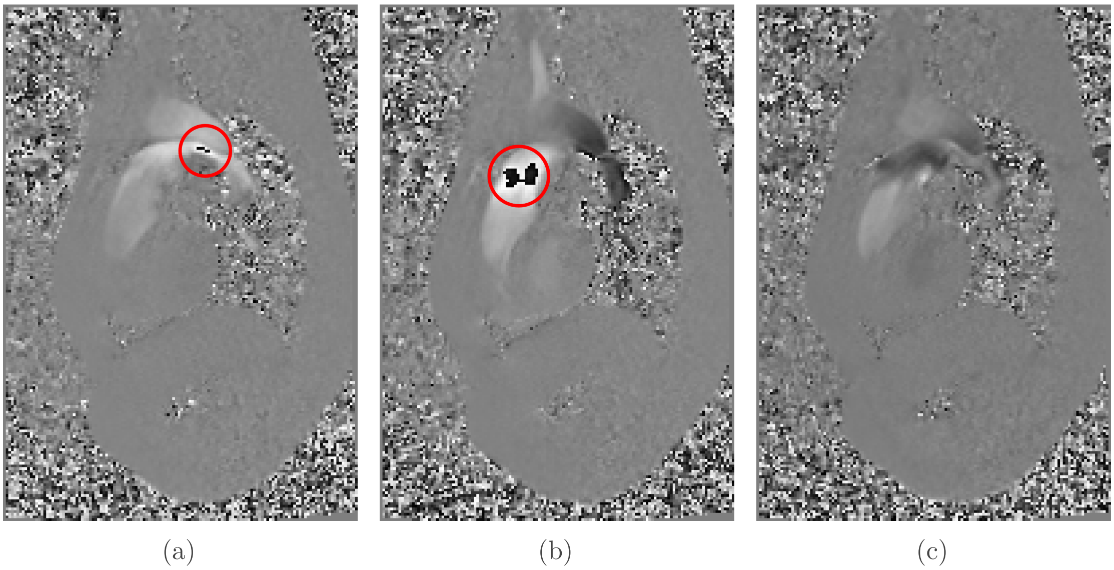


Figure 2.11: Visible phase wraps in directional flow image slices; both the x and y slices (a and b) contain phase wraps (red circle), whereas the z slice (c) does not.

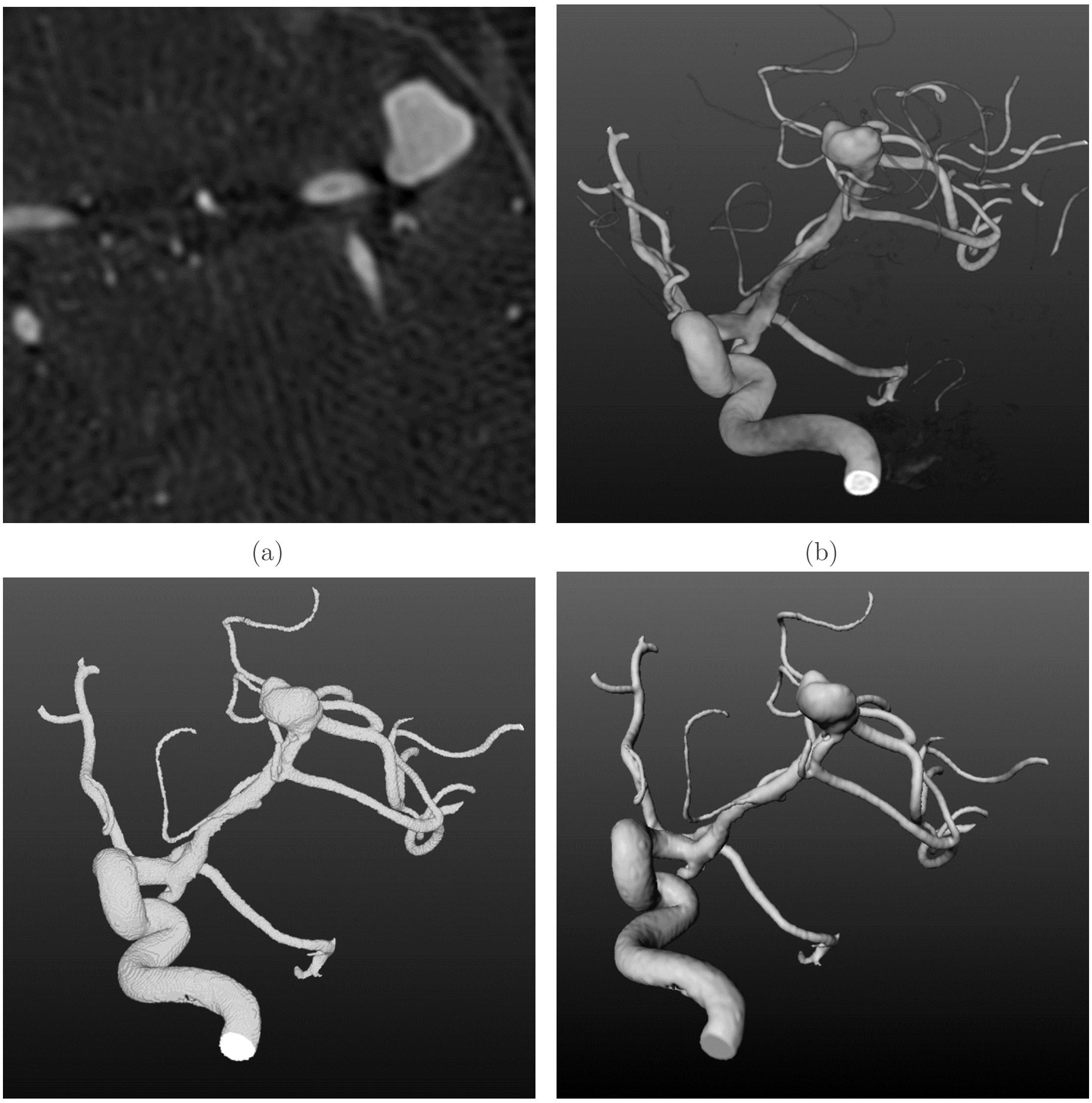
Cerebral Blood Flow Acquisition using Hemodynamic Simulations 2.2.2

Many cerebral vessels are too small to be adequately captured using 4D PC-MRI. Thus, instead of direct measurement, computational fluid dynamics (*CFD*) simulations based on patient-specific vessel geometry are commonly used to acquire cerebral blood flow data. The base for such a simulation is a patient-specific model of the vessel anatomy reconstructed from tomographical images [73]. Commonly, CT Angiography (*CTA*), 3D Rotational Subtraction Angiography (*3D RSA*) or MR Angiography (*MRA*) images are used for this purpose [74–76].

CTA is an invasive imaging technique, as it subjects the patient to both X-ray radiation and the administration of a contrast agent. The resulting images reliably indicate the presence of subarachnoid hemorrhage, which makes them a valuable diagnostic tool. However, CTA lacks sensitivity to specific regions and vessels, such as the carotid artery at the skull base [75]. Additionally, the use of contrast agent can be problematic for patients suffering from impaired renal function [77]. 3D RSA also uses X-rays and a contrast agent to highlight vessels, which is applied via a catheter. Similar to traditional Digital Subtraction Angiography (*DSA*), an image made before the introduction of the contrast agent is subtracted from an image after the introduction of the contrast agent. This effectively eliminates surrounding tissue from the image and only leaves the vessels visible. However, DSA is limited to a single viewing angle that has to be pre-configured and possibly re-adjusted after the image acquisition. 3D RSA instead uses a rotating C-arm to acquire multiple images from different angles, which allows for the reconstruction of a 3D volume [78].

MRA image acquisitions take significantly more time and thus can be problematic for patients in critical conditions [77]. In non-acute cases, Time-of-Flight MR (*ToF MRA*) sequences are often used. A ToF acquisition is based on “fresh” blood flowing into a region of interest that has already been magnetically “saturated” by a previous RF impulse. This blood exhibits a stronger response signal than the surrounding tissue, thus creating a contrast difference [62]. The downside of using ToF MRI is its sensitivity to patient movement, which is problematic in case of acute subarachnoid hemorrhage which may prevent the patient from remaining still [75]. An alternative for ToF MRI is a contrast-enhanced acquisition (*CE MRA*), where a contrast agent is admitted to the blood stream to shorten T_1 relaxation times, increasing image contrast and decreasing acquisition time [62].

From the tomographic image, a 3D surface mesh needs to be extracted. Generally, the images provide enough contrast for an initial segmentation to be created from thresholding or region growing [76] with the subsequent application of the *Marching Cubes* [79] algorithm (see Figure 2.12). However, this approach requires manual post-processing to correct artifacts such as holes or melted vessels. Instead, more sophisticated methods to extract a vessel mesh directly from the image data can be employed, such as implicit models [80].


(a)

(b)

(c)

(d)

Figure 2.12: *Slice image (a) and volume rendering (b) of a 3D rotational subtraction angiography (*RSA*); binary segmentation of the vessel lumen (c) and final surface mesh (d).*

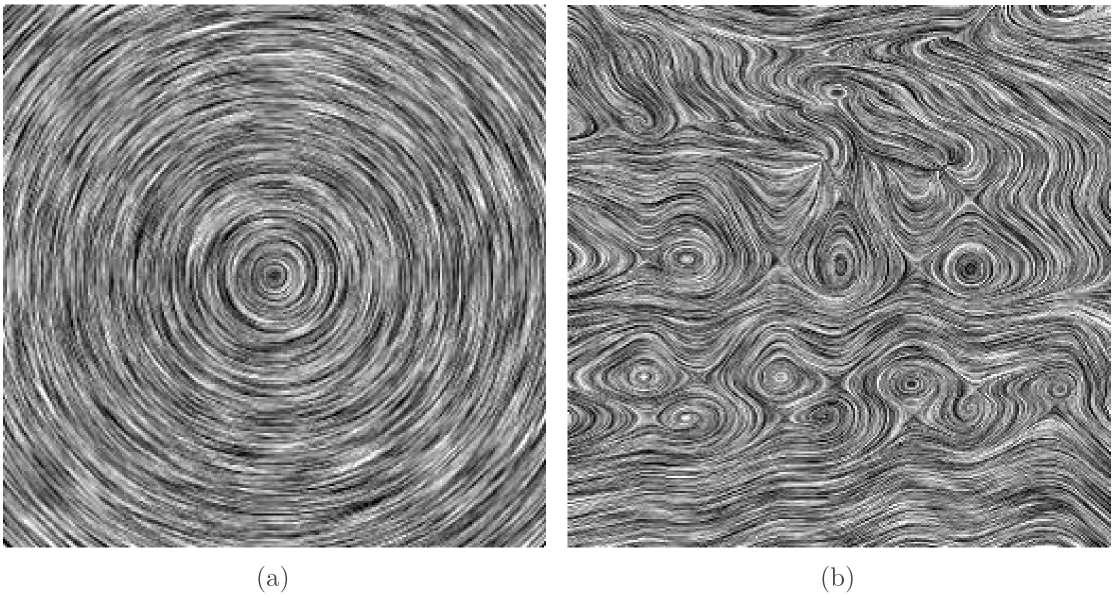
Images courtesy of Sylvia Saalfeld.

Within the hemodynamic simulation, the blood flow is governed by the *Navier-Stokes* equation for momentum conservation [81]. Thus, in addition to the vessel geometry, the simulation also requires flow measurements at each of the vessel's inlet and outlet planes. However, this type of data is typically not acquired in clinical practice. Instead, analytically described inflow conditions [82] or non patient-specific flow measurements from a healthy volunteer adapted to the vessel size and location [81] are used. Similarly, the flow division for the outflow boundaries from the selected vessel tree is based on their geometry instead of patient-specific measurements. Due to imaging limitation, very small vessel outlets are ignored for the simulation. However, these small vessels are assumed to have little influence on the overall intra-aneurysmal flow patterns [82]. For the simulation, vessel walls are commonly assumed to be rigid. This is acceptable due to the low amount of radial dilatations of cerebral vessels during the heart cycle. A more complex flow simulation including vessel wall dilatations would require additional information about local wall properties and thickness as well as intra-arterial pressure, which is not feasible in clinical practice [82]. Additionally, vessel walls are usually defined as *no-slip boundary*, meaning that the flow velocity directly at the boundary is zero. Similar to 4D PC-MRI flow measurements, the hemodynamic simulation results in a time-resolved flow field.

Flow Visualization 2.2.3

Flow visualization is an essential part of scientific visualization. It has many application areas, e.g. the visualization of aerodynamics for the automotive and aircraft industries, meteorology or oceanography. Flow visualization techniques can be broadly separated into two groups: *direct flow visualization* and *integration-based flow visualization* [83, 84].

Direct flow visualization techniques strive to present a visualization that is as close to the original data as possible. Common techniques to visualize two-dimensional flow data include color coding, wherein specific attributes such as velocity are mapped to


(a)

(b)

Figure 2.13: Circular (a) and turbulent (b) 2D vector field visualized using LIC [85].

Republished with permission of ACM, from "Imaging vector fields using line integral convolution"; Brian Cabral, Leith Casey Leedom; ©1993; permission conveyed through Copyright Clearance Center, Inc.

a color scale. To highlight regions within the data, contouring can also be used. The flow direction can be visualized using lines or arrow glyphs. These techniques can also be combined, for example by coloring the arrow glyphs to indicate velocity. Another common approach to directly visualize a 2D flow field is Line Integral Convolution (*LIC*), whereas an arbitrary texture is locally blurred according to the flow direction of a vector field (Figure 2.13) [85].

For three-dimensional flow datasets, additional challenges such as occlusion arise. The visualization of singular slices is an in-between solution to render three-dimensional flow data. In this case, established techniques for two-dimensional flow visualization may be used. For a true three-dimensional visualization, volume rendering can be considered as a extension of two-dimensional color mapping [83]. However, the lack of clear boundaries between flow structure may make it difficult to find an appropriate opacity mapping.

Additionally, flow data in non-cartesian grids can greatly increase the computational complexity for volume rendering and may even require a re-sampling of the data. As an alternative to volume rendering, iso-surfaces can be extracted from the flow field. If the dataset lacks clear boundaries, rendering iso-surfaces may lead to misinterpretations, as by nature the surfaces imply a clear boundary even when none actually exists within the data.

Integration-based flow visualization techniques perform integration on the flow field and visualize the resulting integral objects [83]. Each integration requires a starting point ("*seed*"). An inappropriate seeding strategy may cause critical features within the flow to be missed or other regions to become cluttered with integral objects [87]. Thus, a common challenge for integration-based techniques is the selection of seed points, lines or regions.

The following section gives a brief overview over some commonly used integral objects.

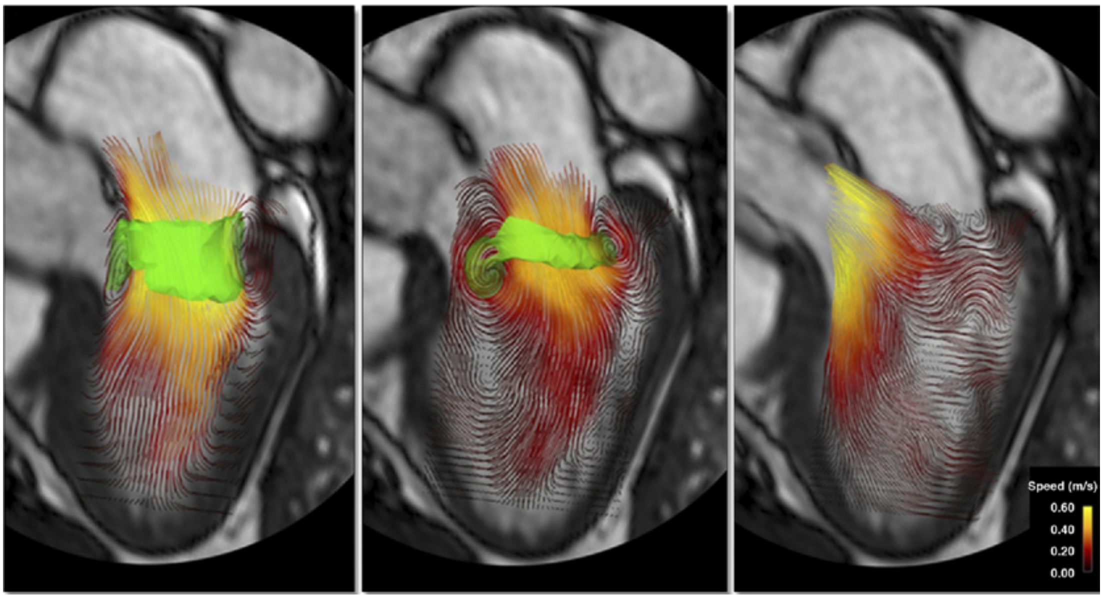


Figure 2.14: Streamlines representing blood flow in the left ventricle at 3 different points in time (left to right: early diastole, late diastole, ejection phase). A vortex ring is highlighted using an iso-surface (green) [86].

Reprinted from JACC: Cardiovascular Imaging, Vol 5; P. Sengupta, G. Pedrizzetti, P. Kilner, A. Kheradvar, T. Ebbers, G. Tonti, A. Fraser, J. Narula: "Emerging Trends in CV Flow Visualization", pp 305–316, ©2012, with permission from Elsevier.

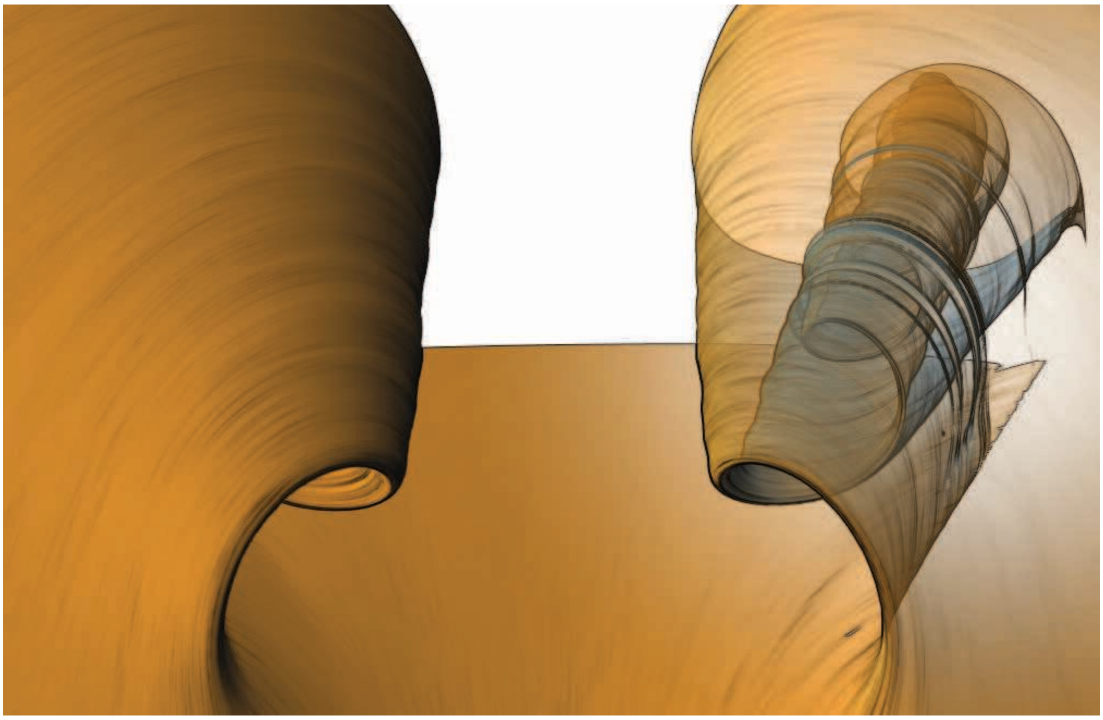


Figure 2.15: Stream surface visualization using lighting, textures, silhouettes and transparency to support the perception of surface orientation and reveal inset features [89].

Streamlines describe lines that follow the flow direction from a seed point in a single time step. One can imagine a streamline as the trace of a buoy that is moved by a constant stream. An example of a streamline-based visualization can be seen in Figure 2.14. Unlike streamlines, *pathlines* take time-dependent flow into account. Using the same analogy as before, a pathline can be seen as the trace of a buoy moved by a stream that changes over time. A fundamental difference between stream- and pathlines is that streamlines can never intersect each other (as the same position in the field cannot have different flow vectors), while pathlines can (as the same position in the field can have different flow vectors at different points in time). *Streak lines* are lines connecting multiple particles seeded at the same point in space, but different points in time. Keeping with the buoy analogy, imagine dropping multiple buoys one after the other into a stream from a pier, than taking a photo and connecting all of the buoys with a line. Similar to streamlines, streak lines also cannot intersect themselves or other streak lines. The differentiation between streamlines, pathlines and streak lines can only be made for unsteady, i.e. time-dependent, flows. In steady flows, these three types of lines are identical [84].

In addition to lines, flow can also be represented by *integral surfaces*. Stream, path and streak surfaces are an extension to stream-, path- and streak lines, but instead of seeding a *line* from a *seed point*, a *surface* is seeded from a *seed line* [88]. Similar to streamlines, stream and streak surfaces cannot intersect each other. Since there is no flow passing through them, they can be used to separate regions of similar flow within a flow field [88]. Using surfaces instead of lines increases the complexity and occlusion in the resulting visualization, thus benefitting from illustrative rendering techniques (Figure 2.15).

An important concept for the analysis of a two-dimensional flow field are *critical points*, i.e. points where the velocity magnitude is zero, as by knowing their *type* (Figure 2.16),

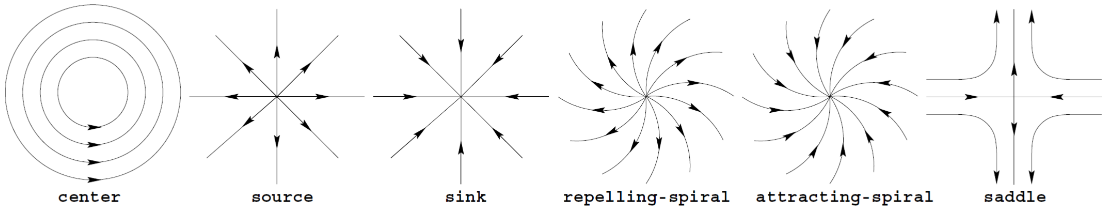


Figure 2.16: *Types of critical points in 2D [95].*

©2000 IEEE

one can infer the flow behavior around them and by extension the entire topology of the flow field [90]. Although the concept of topological visualization can be applied to three-dimensional flow fields, such visualizations are less common due to being more computationally demanding and having a tendency to quickly becoming cluttered for complex flow fields [91]. Thus, 3D flow fields are often visualized with specific features such as vortices in mind. Generally, a *vortex* can be considered as flow swirling around a core, however finding an exact mathematical definition of a vortex is challenging [92]. Vortices can be detected either by detecting the *vortex region*, i.e. the region in which the swirling flow occurs, or by detecting the *vortex core line*, i.e. the line that the vortex swirls around [93]. There exists many approaches for the detection of vortices, yet a universally reliable method has yet to be found [94].

Hemodynamic Parameters 2.2.4

Regardless if the flow field was measured using 4D PC-MRI or acquired from a hemodynamic simulation, it can be used to extract a set of additional quantitative hemodynamic parameters, with some of the most common ones detailed in the following.

- **Velocity**. For each point in the flow field, the time-dependent velocity $\vec{v}$ represents the local flow in x, y and z direction, usually denoted in m/s. Velocity is the basis for each of the other hemodynamic parameters and also used to integrate stream- and pathlines. In addition to being defined for each point in the flow field, both *velocity* and *velocity magnitude* can also be mapped to pathline points.
- **Stroke Volume**. The SV measures the amount of blood pumped out of the heart into the aorta or pulmonary artery. It is usually visualized as a time-dependent curve over a single heart beat [96].
- **Regurgitation Fraction**. The RF indicated the amount of blood that flows back from the aorta or pulmonary artery artery into their respective ventricle. Although a small amount of regurgitation occurs even within healthy subjects, higher regurgitation fractions indicate a pathology affecting the aortic or pulmonary valve.
- **Pressure**. The pressure p represents the amount of kinetic energy within the blood flow. The calculation of pressure is a mathematical part of hemodynamic simulations, thus it is directly available in simulated flow datasets. With measured flow data from 4D PC-MRI, the direct acquisition of pressure values is not possible. However, relative pressure maps can be calculated from the velocity field [97].
- **Wall Shear Stress**. The *WSS* encodes the *stress* on the *vessel wall* exerted by the tangential blood flow passing by. The directional WSS can be calculated

from the flow velocity close to the vessel wall. However physicians are often more interested in the WSS magnitude. Both areas with high and low WSS magnitude are associated with increased rupture risks of intracranial aneurysms [18, 98]. Due to the comparatively low resolution and signal-to-noise ratio (SNR), the WSS is less meaningful for measured 4D PC-MRI data [99]. Since the numeric values of the WSS are not suitable to compare multiple aneurysms in either the same or different datasets, the *normalized wall shear stress* if often used for this purpose. This value can be calculated by normalizing the WSS magnitude on the aneurysm surface with the average wall shear stress magnitude on the parent vessel.

- **Oscillatory Shear Index**. This parameter (OSI) is related to WSS and quantifies the alignment of the wall shear stress with the average wall shear stress vector over time. It is calculated using the instantaneous shear stress vector wss and the cycle period T, and yields values from 0 (strong alignment) to 0.5 (weak alignment). High OSI values therefore indicate areas where the wall shear stress direction changes strongly over time.

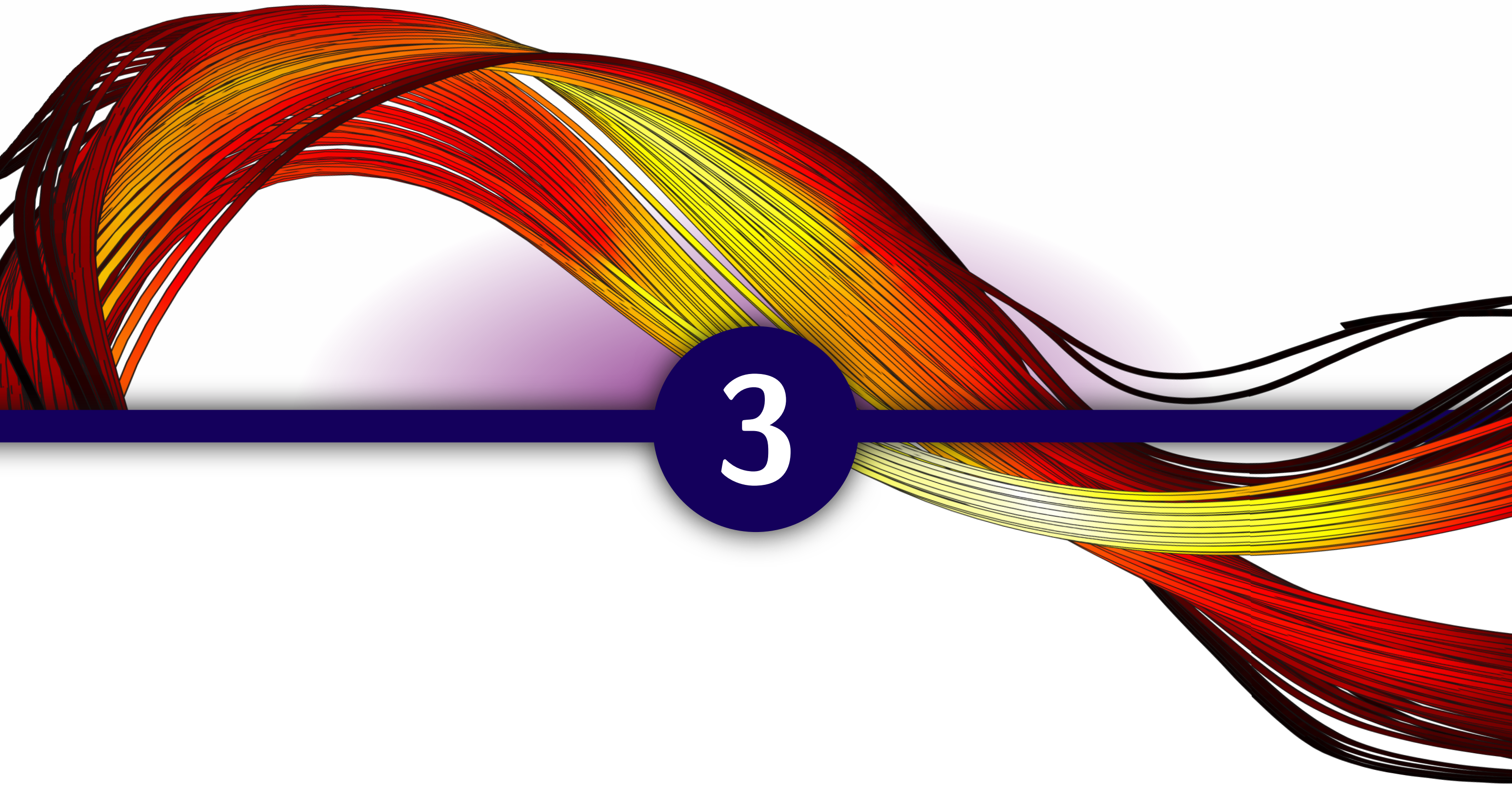
$$OSI = \frac{1}{2} \cdot \left(1 - \frac{\left| \int_0^T wss \mathrm{d}t \right|}{\int_0^T |wss| \mathrm{d}t} \right) \tag{2.1}$$

- **Residence Time**. This parameter is commonly mapped to pathlines and encodes the amount of time the flow particle represented by the pathline *resides* within an aneurysm. A more general variant of this parameter, turnover time (ToT), denotes the time a particle resides within a particular vessel region. Residence and turnover time are commonly denoted in s.

In addition to these quantitative parameters, physicians are also interested in the occurrence and characteristic of various qualitative flow patterns.

- **Vortices**. Vortical or helical flow describes a flow pattern that exhibits a circular movement around a vortex core line. Generally, the flow within vessels is considered laminar. Due to the curvature and branching of vessels, some natural vortical patterns occur even in healthy subjects. For example, slightly helical flow can be observed within the aortic arch [100]. However, vortical or helical flow patterns have been linked to various pathological changes of the vessel [101–104].

- **Intra-aneurysmal flow**. Each aneurysm has its own hemodynamic structure, based on the position of the aneurysm on the parent vessel [105] and the unique shape of the aneurysm [106]. For cerebral aneurysms, researchers are often interested in the *inflow jet*, which describes a blood stream that enters the aneurysm with a hight velocity, as well as the *impingement zone*, which is a region on the vessel wall that is being directly "hit" by the inflow jet [107].



3

Related Work on Hemodynamic Visualizations

4D PC-MRI Preprocessing and Segmentation 3.1

Segmentation is a fundamental part of many computer-aided diagnosis (CAD) systems, as the accuracy of results obtained from these systems is directly dependent on the segmentation quality of the underlying medical data. There is no universally applicable segmentation technique, as each combination of anatomy, image modality and overall acquisition goal presents their own unique challenges for a segmentation [108]. Thus, each application domain within the realm of medical visualization requires specifically tailored segmentation solutions. Within the domain of vessel visualization, various techniques have been developed over time to extract and segment the vessel geometry for different image modalities [109]. In the case of 4D PC-MRI, the acquired data consists of the time-resolved flow field and magnitude images. Performing a segmentation directly on these images is challenging due to the overall low contrast, low resolution, artifacts and non-homogenous brightness of the vessels. This section presents selected image processing and segmentation techniques that allow extracting the vessel geometry from 4D PC-MRI. The correction of image acquisition artifacts is not withing the scope of this section, however. A comprehensive overview of artifact correction techniques can be found within the work of Köhler et al. [72].

Preprocessing

To overcome the issues of low contrast and noise, visualization and segmentation of the anatomy is often based on the magnitude-based TMIP images. This follows the base assumption that any voxel containing flow with a high velocity in at least one time step of the magnitude images during the cardiac cycle is most likely part of the vessel anatomy [110]. However, the sole reliance on magnitude data discards the directional flow information present in 4D PC-MRI datasets. Especially when the data exhibits a low SNR, segmenting only magnitude images can easily lead to segmentation errors [111]. Therefore, various methods have been developed to improve contrast in the segmented images by incorporating additional or derivative information from the flow images. An example are phase contrast magnetic resonance angiography ($PCMRA$) images that weight the directional velocity images using the magnitude images [112].

Chung et al. [111] proposed the use of Local Phase Coherence (LPC), which is defined as the sum of dot products of a voxel's flow vector with those of the adjacent vectors. The result is an image highlighting areas with not just high flow velocity, but instead coherent flow (i.e. flow traveling with roughly the same speed in a similar direction). As voxels outside of the vessel anatomy contain only random noise, these regions do not form coherent flow and are therefore suppressed. A similar measure by Solem et al. [113], the Eigenvalue Coherence (EVC), incorporates eigenvalues of a local velocity tensor instead of dot products, which reveals its dominant flow directions. If there is only one dominant flow direction, the voxel is most likely part of the vessel anatomy. In case no dominant flow direction exists, the flow is assumed to be the result of random noise and the voxel can therefore be classified as background. An interesting case is the presence of two dominant flow directions, as it indicates that the voxel is part of the vessel wall. Here, the movement of the blood overlaps with the movement of the vessel wall, which is usually orthogonal to the blood flow direction, hence the second dominant flow direction.

Another way of quantifying the coherence of flow is the Finite Time Lyapunov Exponent ($FTLE$) [114]. The FTLE is a measurement for the separation of weightless particles placed at *similar positions* and being moved by a flow field over time [115, 116]. In areas with coherent flow, particles starting within neighboring voxels should reach a similar final position after a given amount of time. However, in the direct vicinity of the vessel boundary, where the coherent blood flow borders random motion caused by noise, a high rate of separation can be observed. Therefore, areas of high intensity in an FTLE image can be interpreted as an approximation of the vessel boundary [117].

Segmentation

Van Pelt et al. [118] used direct volume rendering to visualize the vessel geometry as context information. The vessel surface is visualized using raycasting on the TMIP data and clipped by an automatically positioned plane to show the inlying blood flow. This approach allows for fast exploration of 4D PC-MRI flow data, as it works directly on the acquired data by forgoing the need for a segmentation.

In many other applications, an explicit vessel segmentation is required to visualize the blood flow data. Hennemuth et al. [112] used watershed transformations with user-defined include and exclude points on PCMRA images to facilitate a segmentation. By labeling the include points, users can segment multiple anatomical structures at the same time. Similarly, Köhler et al. [99] used interactive graph cuts to obtain a vessel segmentation from the TMIP images. The user iteratively marks regions in the image as either belonging to a specific vessel or to the background until they are satisfied with the resulting segmentation. Another graph cut-based approach was presented by Gülsün and Tek. They first extract the vessel centerlines by applying a multi-scale medialness filter to the magnitude images [119]. Afterwards, the centerlines are used as input for a graph cut algorithm to facilitate a segmentation and reconstruct the vessel surface [120].

The previously mentioned techniques primarily focus on segmenting the vessel lumen to extract its geometry. Krishnan et al. [117] introduced an algorithm to detect the vessel boundary instead of the entire lumen, albeit not for the purpose of segmentation but as a stopping criterion for their pathline integration. For each voxel, the average flow velocity over time as well as the FTLE value are calculated. Regions with a low average flow velocity and a high degree of flow separation (denoted by a high FTLE value) are considered to be vessel boundaries.

Van Pelt et al. [121] used an active surface model to obtain a segmentation from 4D PC-MRI data. An initial surface is generated using the Marching Cubes algorithm on the TMIP image. This surface is then deformed by applying *internal* and *external* forces. The external forces draw the surface towards vessel boundaries, which are detected using the gradients of either magnitude, EVC or LPC images as a base. The internal forces work as a counter balance and prevent the surface from being deformed excessively from its initial shape. The authors compared the results from their approach on the magnitude, LPC and EVC images, with the latter two achieving superior results due to their incorporation of directional information.

Volonghi et al. [122] presented a semi-automatic segmentation technique to extract the vessel surface. The approach uses the Fast Marching Level Set algorithm and requires an approximation of the vessel centerline as input. To create this centerline, the user places markers within the vessel, which are automatically connected using cubic splines. During this step, a coarse approximation of the vessel surface obtained by using the

Marching Cubes algorithm with a fixed threshold on the magnitude images at the peak systole is shown to provide orientation for the user. This centerline is then used as an initial condition for a Fast Marching Level Set algorithm, which automatically extracts a refined surface.

Bustamante et al. [123] use a priori knowledge about the general shape of various vessels for a fully automated segmentation (Figure 3.1). In their approach, the aorta, pulmonary artery and vena cava were first manually segmented in a dataset obtained from a healthy volunteer and fed into an atlas. This atlas could then be registered to new datasets to facilitate a segmentation without further manual input. A coarse registration to TMIP or PCMRA data was achieved using affine transformations and then fine-tuned with a non-rigid registration. This registered atlas could then be further refined by fitting it to each individual magnitude image, creating a time-dependent segmentation. Due to no manual input being required for the segmentation, this approach achieves a high degree of reproducibility.

Another example for the use of a priori knowledge was presented by Bergen et al. [124]. For each spatial position in the flow image, the flow curves denoting the flow velocity in x, y and z direction were extracted. These curves were then matched to the typical flow curves of voxels one would expect within specific regions of the aorta, e.g. high systolic flow exclusively in x direction during the systole, to facilitate a segmentation.

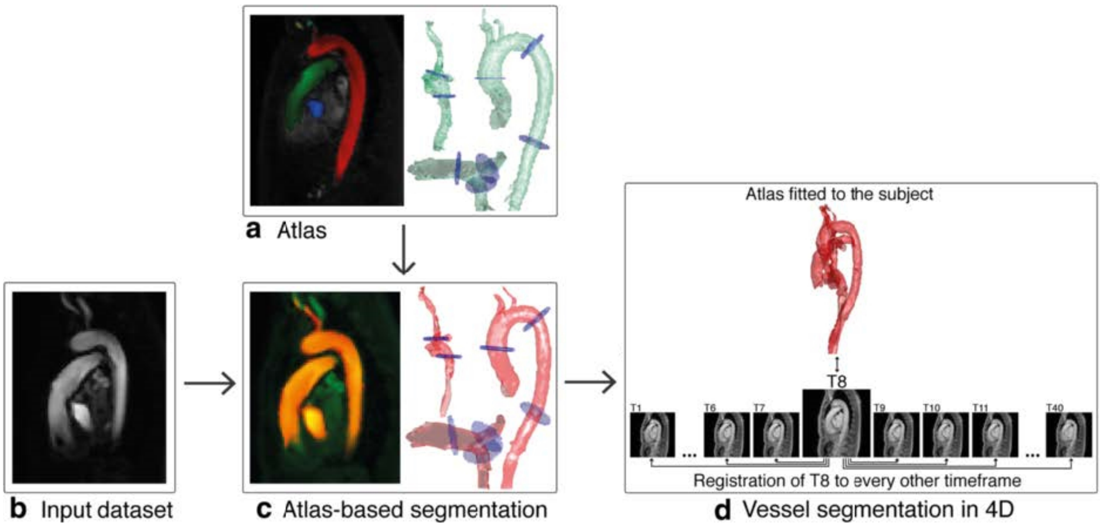

a Atlas
Atlas fitted to the subject
T8
T1
T6
T7
T9
T10
T11
T40
Registration of T8 to every other timeframe
b Input dataset
c Atlas-based segmentation
d Vessel segmentation in 4D

Figure 3.1: *Workflow for the atlas-based segmentation by Bustamante et al. [123]. A previously segmented atlas dataset is registered on a new input dataset to form an atlas-based segmentation. This segmentation can then be fitted onto each magnitude image to create a time-dependent segmentation.*

The image is taken from an open-access paper by Bustamante et al., which is published under the Creative Commons Attribution 4.0 International license. ©2015, Bustamante et al. URL: https://doi.org/10.1186/s12968-015-0190-5 License: https://creativecommons.org/licenses/by/4.0

Perception-oriented Vessel Visualization 3.2

Once the segmentation and extraction of the vessel surface has been achieved, the next challenge for vessel visualization is to support the user in perceiving and and interpreting the complex three-dimensional vascular structures. Thus, various methods to enhance the perception of both depth and shape in computer-generated 3D images have been implemented [128, 129]. Many of these techniques utilize colors or textures to convey depth. Ritter et al. [130] employed hatching to visualize depth relations in complex vascular structures and evaluated this approach in a study with 160 participants. Whenever two sections of the anatomy were overlapping, the posterior structure was hatched to simulate a shadow. Hansen et al. [131] introduced distance-encoding silhouettes, where the depth is mapped to the width of an outline drawn around all of the vessels.

Lawonn et al. [125] presented a combination of depth-dependent halos, support lines and the illustrative shadows by Ritter et al. to improve the perceptibility of depth. The support lines are cast from manually selected points of the vessel onto a plane, creating an effect similar to beams holding the model up, limiting the ability to freely rotate the vessel. In a later publication, Lawonn et al. [126] extended this approach by casting the support lines not to a plane below the vessels, but to a cylindrical support geometry surrounding the focal part ot the vessel tree. Instead of support lines, Lichtenberg et al. [127] employed glyphs to encode the depth of various points on the vessel tree. An advantage of this approach is that additional information can be encoded within the glyph, such as the distance to a risk structure. A comparison between three approaches can be seen in Figure 3.2.

Rheingans and Ebert [132] used distance color blending, a combination of intensity depth cueing and color modulation, to increase depth perception in volume models by mimicking the light-scattering effect of the atmosphere. Joshi et al. [133] later validated this method specifically for enhancing depth perception in vessel visualization. A similar method emulating real-world optical effects is Depth of Field (DoF), where objects are gradually blurred depending on their distance to a focal plane. Without using eye tracking, this focal plane needs to be positioned manually or using heuristics [134]. Grosset et al. [135] evaluated the effectiveness of various DoF techniques in a study with

(a) (b) (c)

Figure 3.2: Depth visualization using support lines (a) [125], support anchors (b) [126] and glyphs (c) [127].

25 participants. They found that DoF only supports depth perception when the focal plane is placed in the front of the scene.

Instead of emulating optical effects, the chromadepth technique by Steenblik [136] directly uses the optical properties of the eye lense to create a depth effect. The depth of each surface point in the image is mapped onto a specific color gradient. As light of different wavelengths is refracted at different angles in the lens of the eye, this gradient creates the illusion of depth in an otherwise flat image. Unlike other techniques, such as Anaglyph 3D, chromadepth does not necessarily require any special type of surface, glasses or other additional devices, although the effect can be strongly enhanced by diffraction grating glasses [137].

A variation of this technique called pseudo chromadepth (*PCD*) was introduced by Ropinski et al. [134] to increase depth perception in 3D angiography datasets. Instead of using the full range of colors visible to the human eye, PCD only uses a gradient from red (low depth) to blue (high depth), as a wider range of hues might distract from the shading used to convey shape. The high difference in wavelength between red and blue maximizes their chromadepth effect. Additionally, red is attention-grabbing and intuitively perceived as foreground, whereas blue – the color of the sky – is generally perceived as background. The effectiveness of this approach has been confirmed by multiple studies [134, 138]. A major downside of using a chromadepth or PCD color scale is that it completely occupies the color channel and thus prevents it from being used to map additional information about the vessel surface. Therefore, instead of using the chromadepth or pseudo-chromadepth scale directly on the vessel surface, Kreiser et al. [139] proposed using the empty space ("Void Space Surfaces") in between the vessels to map depth information (Figure 3.3).

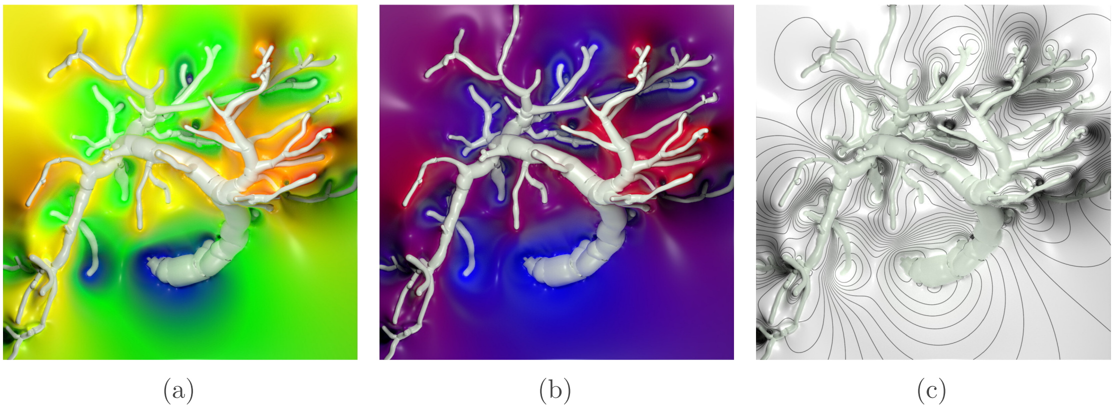

(a) (b) (c)

Figure 3.3: *Mapping of depth values to Void Space Surfaces using different color scales (a, b,c) and iso-lines (c) [139].*

Generation and Visualization of Pathlines 3.3

Although the vessel surface provides important context information and can be used to visualize hemodynamic parameters such as pressure or wall shear stress, the flow field itself needs to be visualized to obtain a comprehensive understanding of the blood flow. Geometry-based techniques, such as streamlines or pathlines, are the most frequently used methods for flow analysis [87]. The biggest challenge of these methods is to avoid clutter and occlusion without missing important features in the data. Interesting structures, such as vortices, are often hidden within other, more laminar flow. Therefore, a variety of methods have been developed to automatically or semi-automatically highlight important flow structures. These filtering methods can be divided into two groups:

Explicit filtering removes lines not satisfying a specified criterion. Lee et al. [140] proposed a view-dependent explicit filtering technique that resolves occlusion for pathlines. They calculate a 3D entropy field from the flow field and perform a *maximum entropy projection* to obtain a view-dependent 2D entropy map containing the highest entropy and its corresponding depth value for each pixel. Pathlines that only pass through low-entropy regions, but at the same time occlude pathlines with higher entropy, are filtered from the visualization. Specifically for the exploration of blood flow data, Salzbrunn et al. [141] introduced *pathline predicates*, a method of grouping vertices in a pathline based on their fulfillment of a user-defined criterion. By chaining multiple predicates together using Boolean algebra, complex filter criteria can be implemented. Born et al. [142] adapted line predicates to support the exploration of cardiac blood flow by designing a set of predefined predicates. Users could adapt and combine these predicates to suit their specific needs for exploring cardiac flow data. Köhler et al. [69] implemented line predicates to extract vortices from cardiac blood flow data. Although these vortices are reliably detected, this approach only allows for a global filtering of pathlines.

Implicit filtering describes rendering techniques to emphasize lines of interest without explicitly removing lines. One example is the work by Günther et al. [144, 145], who presented an implicit filtering approach which modulates the line transparency with respect to view-dependent occlusion and an importance criterion. The opacity of a line segment is reduced when it occludes another line segment of higher importance. Thus, in

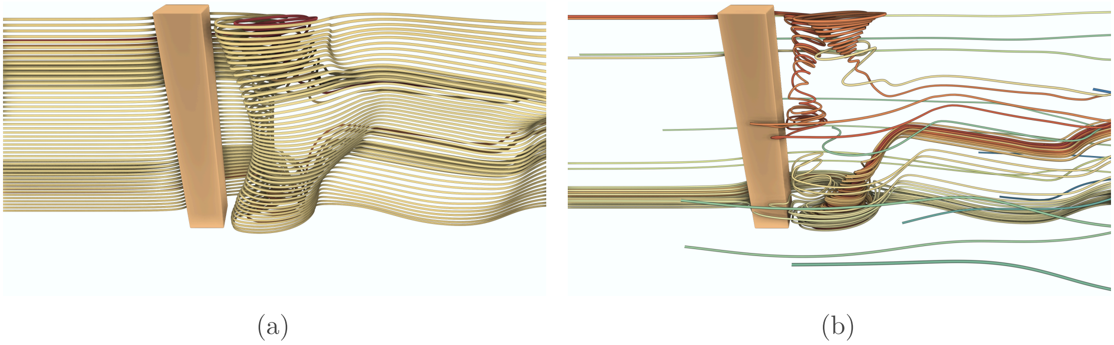

(a) (b)

Figure 3.4: Comparison between a regular seeding approach with filtering (a) and an evolutionary seeding approach (b) to highlight a vortex using the same quality criterion for the filtering and evolutionary algorithm [143].

Images courtesy of Wito Engelke.

contrast to the previously described approach by Lee et al. [140], occluding pathlines are not completely removed from the visualization. Examples for possible important criteria include line length, curvature, flow velocity or vorticity.

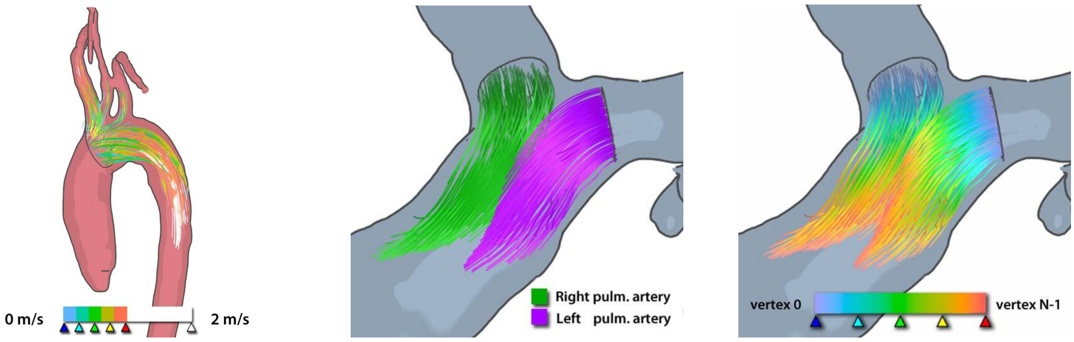
Both types of filtering approaches discard or hide a high percentage of the calculated lines, which is unfavorable in cases where the line integration is expensive. Additionally, since these techniques operate on pre-existing sets of pathlines, it is hard for them to emphasize features that are poorly represented in the unfiltered set. Therefore, some techniques, such as the previously mentioned explorative approach by van Pelt et al. [118], include dynamic seeding capabilities to overcome this issue. Particles, pathlines or integral surfaces can be created on demand in the region of interest by placing a cup-shaped seeding geometry within the vessel. Similarly, de Hoon et al. [146] use particles to simulate injecting ink into a vessel at a specific location. By using different integration and seeding options, these particles can mimic stream-, path- and streak lines. Additionally, the movement of individual particles can be modified based on the probabilistic distributions of noise on the original medical data, thus implicitly visualizing uncertainty. Another example for an explorative approach to filtering pathlines presented by Broos et al. , who used a user-defined transfer function to determine possible seed points from a flow field [147]. This function determines for each voxel the probability of becoming a seed point for a pathline. Multiple transfer functions can be combined to include multiple features. This approach effectively increases the density of pathlines in regions of interest and decreases the density of lines in all other regions to prevent occlusion and visual clutter.

Engelke et al. employed evolutionary algorithms to determine suitable seeding locations to achieve dense line coverage for specific flow features [143]. With this approach, they were able to achieve high-quality results with significantly fewer line computations compared to traditional filtering approaches (Figure 3.4). They also managed to overcome the problem of under-sampling, as their approach is not tied to a predefined sampling resolution. Esturo et al. [148] presented a method to find globally optimal stream surfaces by utilizing a domain graph. The approach is tied to a predefined resolution and thus unable to find solutions not contained in the graph. In general, adaptive methods follow a similar goal as evolutionary approaches by striving to spend most efforts in regions of interest. A good example in the context of flow analysis is the work by Barakat et al. [149], which uses adaptive refinement for an accurate computation of the flow map. The exploitation or refinement aspect of this approach is similar to that of evolutionary algorithms, which is facilitated by mutation. The exploration aspect differs, as the coverage of adaptive methods is determined in the first step by choosing a base resolution. In contrast, evolutionary algorithms try to achieve good coverage by continuous exploration, which allows them to work with a low population size from the beginning.

3.4 Combined Visualization of Pathlines and Vessel Surfaces

On the one hand, as previously mentioned, the vessel surface provides important context information and is therefore often included in visualizations of medical flow. On the other hand, it is also an additional source of occlusion. Therefore, *focus-and-context* visualization techniques that ensure the visibility of inlying important flow features while maintaining as much context information as possible are required. Viola et al. [150] introduced *importance-driven volume rendering*, whereas based on the viewing direction, less important structures are automatically suppressed (e.g. using transparency) in favor of more important inset objects. Gasteiger et al. [151] presented *Ghosted Views*, which employ an approximation of the Fresnel effect to modulate the opacity of vessel surfaces, thus showing the blood flow inside of a vessel without removing the entire front-facing part of the surface. In a subsequent study, Baer et al. [152] showed that in comparison to a surface with constant transparency, this approach allowed for a more accurate analysis of the aneurysm and its flow patterns. Glaßer et al. [153] presented a similar technique, which uses the Fresnel effect to highlight vessel boundaries. Lawonn et al. [154, 155] modified the approach of Gasteiger by mapping surface shape and distance to pathlines to the vessel surface using a 2D texture lookup table. Additionally, the surface opacity is modulated using a measure derived from smoothed *suggestive contours* instead of the Fresnel effect. Suggestive contours were introduced by DeCarlo et al. [156] and describe view-dependent contour lines added to three-dimensional renderings to emphasize their shape, similar to the contour lines an artist would draw.

Another focus-and-context approach by Gasteiger et al. [157] uses an elliptical "lens" manually positioned in screen space. This lens blends between two different visualization styles and can be used to clip away the vessel surface and reveal underlying flow in a specific focal region. However, due to lens existing only in screen space, camera movement may result in an undesired change of the focal region and require repositioning of the lens. Lawonn et al. [158] used an automatic cut-away technique that removes the vessel surface when it occludes any pathlines, allowing for a simultaneous visualization of blood flow and parameters on the vessel wall. This reduces the amount of surface area that needs to be culled, but prevents the entire flow course from being visible at one time.



(a) (b) (c)

Figure 3.5: Explorative forward (a) and backward (b and c) pathline seeding at a user-specified vessel cross-section [110]. The pathlines are colored according to flow velocity (a), seed plane (b) and distance from the seed plane (c).

Meuschke et al. [159] presented a combined visualization of hemodynamic flow and vessel surface information with a focus on near-wall flow. To prevent occlusion, the vessel surface is mapped onto a 2D plane and displayed alongside the 3D visualization. This approach focuses on correlating multiple surface parameters, such as wall shear stress and wall thickness, by mapping one parameter to the 2D plane using a color scale and an additional parameter to a 3D bar chart hovering above the plane. Neugebauer et al. [160] used the vessel surface as an input for implicit pathline filtering. They employed an automatic detection of potentially interesting surface regions based on the surface geometry, although manual selection of a region is also possible. Instead of filtering existing pathlines according to their distance to the selected region of interest, they dynamically seed new lines close to the region. Van Pelt et al. [110] introduced an implicit filtering technique by interactively selecting vessel cross-sections as seeding planes. By placing these planes into or close to regions of interest, the user can gain insights into both local (such as vortices) and global (such as splitting flow) flow patterns (Figure 3.5).

Oeltze et al. [162] used clustering to reduce visual clutter in simulated cerebral blood flow data by grouping pathlines together based on their geometry or attributes. Only a single representative for each of the resulting cluster is then visualized. In a later publication, Oeltze et al. [163] added dynamic seeding capabilities to better convey complex flow structures, such as embedded vortices. Meuschke et al. [161] classified flow patterns by mapping the aneurysm surface and pathlines to a hemisphere in order to make them comparable between datasets. The pathlines were clustered and a representative of each cluster was compared to a set of predefined pattern types to facilitate an automatic classification of the flow patterns (Figure 3.6). Another efficient clustering approach for blood flow data, using coherent structures, was presented by Englund et al. [164]. They calculate a *coherence map* for pathlines that are seeded from a 2D plane positioned by the user, detailing the amout of divergence over time of the pathlines seeded from similar positions. By finding local minima within this map (i.e. regions in the seed plane that produce similar pathlines) and clustering them together with their associated pathlines, distinct flow patterns can be extracted.

While clustering-based methods are well suited to give an overview of flow, they are less suitable to probe specific flow behavior. Additionally, the reduction of complex flow patterns to a single line may fail to capture their full structure and shape as well as

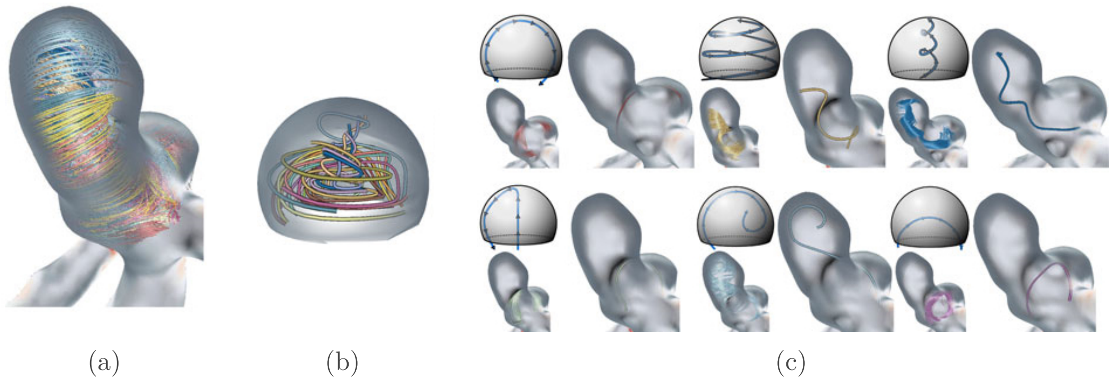


Figure 3.6: *Classifying flow patterns within an aneurysm (a) by calculating representatives and mapping them into a hemisphere (b). The resulting lines are then matched to several predefined flow pattern types (c) [161].*

their relation to the vessel surface. Targeting specific medically relevant flow patterns, Neugebauer et al. [165] used the ostium as a seeding plane for pathlines, increasing the pathline density and highlighting the flow entering or leaving the aneurysm. Instead of highlighting the entire flow within an aneurysm, Gasteiger et al. [107] presented a method to specifically extract and visualize the inflow jet and impingement zone in aneurysms, which are both correlated with the risk of rupture of an aneurysm. Van Pelt et al. [166] extended this approach to allow for a comparative visualization of multiple stent configurations in the same aneurysm. This allows estimating the effect of the different stent configurations, especially with respect to reducing the wall shear stress exerted on the aneurysm wall. However, since the resulting visualization is tailored to a very specific application, it is not suitable for a more general, explorative approach.

Flow Dataset Comparisons 3.5

The comparison of complex information is a key part of data analysis. Although there exist a set of general techniques, such as juxtaposition and explicit encoding, each specific application requires a specialized solution to allow for effective comparisons [167]. Verma and Pang [168] consider three approaches to comparative visualizations: With **Image-level comparison**, the comparison is performed on the output images of visualizations, i.e. by overlaying or displaying them side by side. **Data-level comparison** describes the direct comparison of raw data, requiring an application-specific metric. **Feature-level comparison** describes comparing features derived from the data, such as streamlines, pathlines or iso-surfaces.

Pagendarm and Post [169] present an example for image-level comparison by overlaying visualizations of features from the same flow field to highlight the spatial displacement of extracted features. Schneider et al. [170] implemented a feature-level comparison by extracting contour trees from two scalar fields and finding overlapping features in both datasets.

Various methods have also been explicitly developed for the explorative visualization of medical flow data [171]. So far, visual comparisons of unsteady medical flow data was achieved using image-level techniques based on strong abstractions that typically represent only some flow features. The framework developed by Van Pelt et al. [110] encompasses multiple visualizations. They use illustrative techniques, such as contour rendering, to focus the flow visualization on relevant aspects. These techniques include the use of measuring planes to directly visualize the underlying flow field. The overall flow structures, such as vortices or branching flow, can also be represented using pathlines. Born et al. [172] used various illustrative techniques inspired by hand-drawn anatomical illustrations to create simple visualizations that give an overview of 4D PC-MRI datasets. The depiction of flow is reduced to a set of minimal flow lines representing the main flow characteristics of the dataset. Although this simplified visualization improved the visual comparability of different datasets, it does not support the extraction of comparative measures, such as differences in flow speed or direction.

Angelelli and Hauser [173] presented a method to optimize side by side comparisons of flow in tubular structures, such as vessels. By straightening the structure, multiple representations of the same dataset, e.g. different time points or visualization parameters, can be juxtaposed (Figure 3.7).

In order to make the analysis of cardiac blood flow more objective and comparable, Köhler et al. automatically identified various blood flow patterns [69]. In a later publication, they developed a more abstract representation of vortex flow in cardiac vessels for the purpose of comparing multiple datasets by using circular bull's eye plots [174]. The temporal position of a vortex is mapped to the angle, the position on the centerline to the distance from the plot center, and the vorticity is mapped to color. This creates a highly compact spatio-temporal depiction of the general vortex structure in a single dataset, which can also be juxtaposed for comparative purposes (Figure 3.8). The main goal of this visualization is to enable the physician to quickly distinguish between pathological and healthy flow structures. A refined version of this approach, which shows the spatio-temporal position of singular vortices, was presented by Meuschke et al. [175]. Due to the high level of abstraction, both methods do not enable a more in-depth comparison between datasets.

In addition to these approaches, there also exist methods offering data-level or feature-level comparative visualizations of flow data for very specific use cases. Glaßer et al. [176] presented visualization and preprocessing techniques for the visual analysis of longitudinal studies of brain perfusion MRI data. The datasets are co-registered using the skull as a reference object and displayed using both 2.5D and 3D visualizations. The 2.5D visualization employs parallel coordinate plots. While this method allows for a comparative visualization of multiple datasets, it is strongly tailored to longitudinal studies with a single patient. A comparison between different patients or healthy volunteers is not supported.

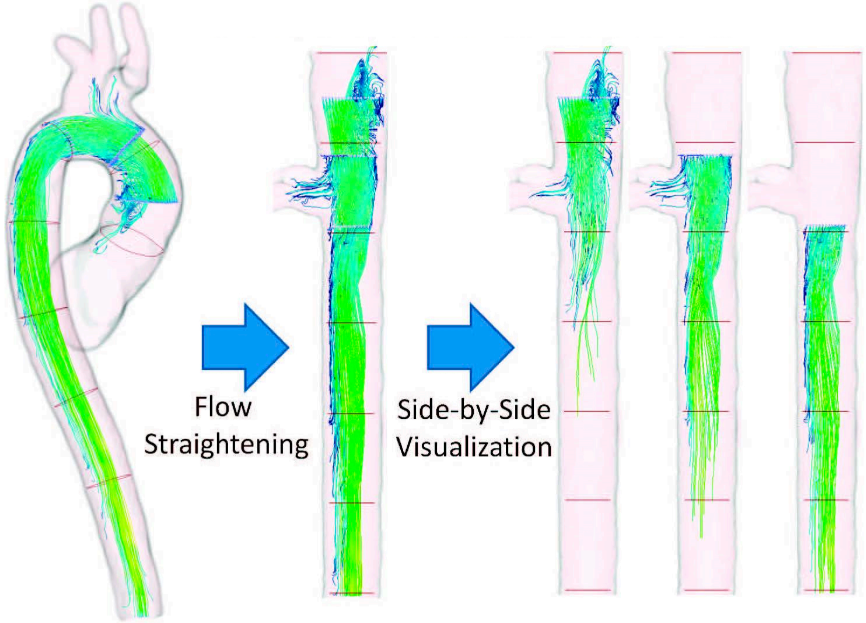


Figure 3.7: *Side-by-side flow visualization of multiple time steps in a straightened aorta [173].*

©2011 IEEE

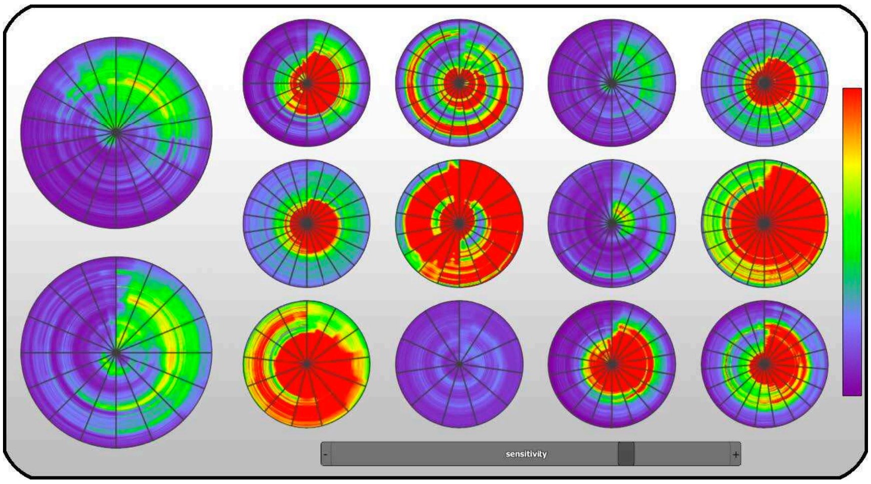


Figure 3.8: *Comparative view of bull's eye plots showing the temporal occurrence of vortices in multiple datasets (right). Two larger plots showing data from healthy volunteers are displayed as reference (left) [174].*

Reprinted by permission from Springer Nature: Bildverarbeitung für die Medizin 2015 [174], ©Springer-Verlag Berlin Heidelberg 2015

Placement within previous work of the Visualization Group 3.6

Within our Visualization Group, several PhD and Habilitation thesis focussing on the analysis of blood flow data have been written. Some of publications authored by former PhD students from our group have already been mentioned in the previous sections of this chapter. In the following, we briefly summarize their work and explain how our current approaches differ.

The first work on this topic within our group started with a cooperation with neuroradiologists from the University Hospital in Magdeburg and the Laboratory of Fluid Dynamics and Technical Flows from the University of Magdeburg. Matthias Neugebauer focussed his work on both segmentation and explorative visualization of cerebral aneurysms. In addition to visualizing the aneurysm and close regions of the harboring vessel, he added a volume rendering of the entire vessel tree to serve as context information [177]. Hemodynamic parameters of the vessel wall could be visualized using 2D planar maps positioned around the 3D surface visualization [178]. Potentially interesting flow structures close to the vessel wall could be automatically extracted and visualized [160]. A more in-depth visual analysis of the flow within the aneurysm is achieved by displaying color-coded streamlines seeded from and directional flow glyphs placed on the ostium, as well as manually placed sectional planes [165].

Rocco Gasteiger developed a surface shading that improves shape perception for the vessel wall, while still keeping inset features such as pathlines visible [151]. The impingement zone within the aneurysm could be visualized both as a surface conveying its shape and size as well as a simplified arrow glyph [107]. Using a lens in screen space, different visualizations (e.g. vessel surface, pathlines or iso-surfaces within the flow field) could be blended with each other to produce a customizable focus-context-visualization [157].

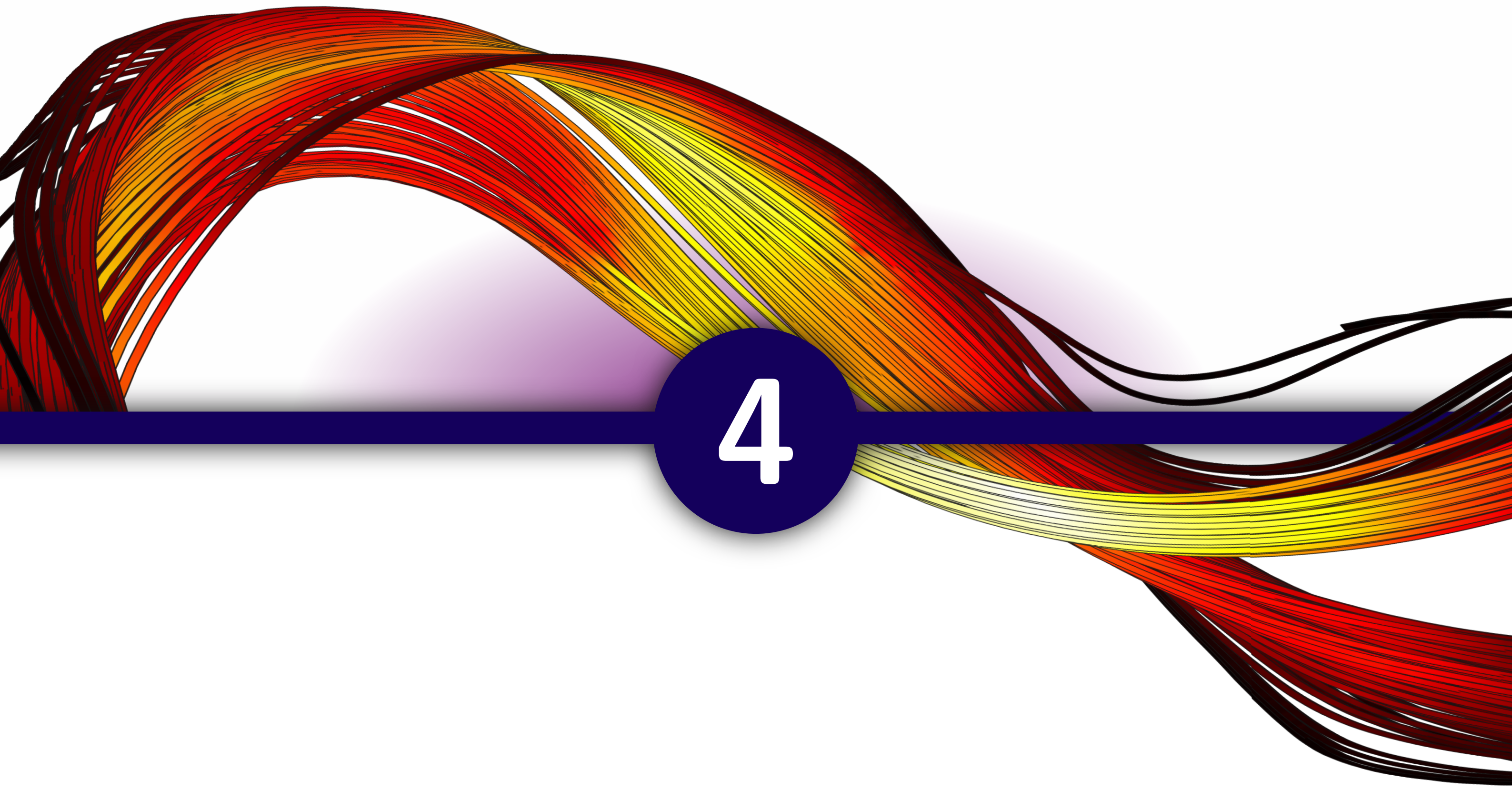
Kai Lawonn improved on the surface shading by Gasteiger et al. by adding more expressive shape cues and increasing the visibility of pathlines [154, 155]. Additionally, he developed a technique to visualize surface parameters simultaneously with pathlines by employing dynamic cutaways [158]. He developed several techniques to improve depth and shape perception of complex vascular structures [125, 126].

Steffen Oeltze-Jafra worked on clustering stream- and pathlines within aneurysms to identify different flow structures [162, 163].

An additional cooperation with radiologists from the University Hospital Magdeburg and the Herzzentrum Leipzig since 2011 led to additional research focussed on cardiac imaging. Benjamin Köhler worked on the detection [69, 179], visualization [180, 181] and comparison [182] of vortices based on 4D PC-MRI images within the aorta and pulmonary artery. Additionally, he extracted several hemodynamic parameters from the measured flow fields [96, 183, 184]. These functionalities were implemented into a single research tool, which was made available to the clinical research partners [99].

Monique Meuschke developed clustering and classification techniques for vortices within the aorta [175, 185]. Later, her focus switched to cerebral blood flow and worked on a clustering and classification system for flow structures within aneurysms [161, 186]. By using glyphs and 2.5D maps, she visualized complex hemodynamic surface parameters such as tensors or wall deformation [159, 187]. Additionally, she employed visual analytics techniques to compare multiple datasets [188].

Many of these previous publications focussed on detecting or visualizing specific flow features, such as vortices or the impingement zone. As I work with both simulated and measured data, my approach to the visualization of blood flow is to allow physicians as much freedom as possible to define features they are interested in. My work on explorative hemodynamic visualization (see Section 4.3) lets them find arbitrary flow structures within an aneurysm based on hemodynamic or geometric parameters of the vessel surface. Similarly, the evolutionary pathline seeding can be freely configured to produce different pathlines based on the user's specific research interest by adjusting its fitness function. The comparative visualization for the aorta using measured 4D PC-MRI data, while including a function to highlight vortices, uses normalized planes to find differences and similarities directly within the flow field in regards to flow velocity and direction.



4

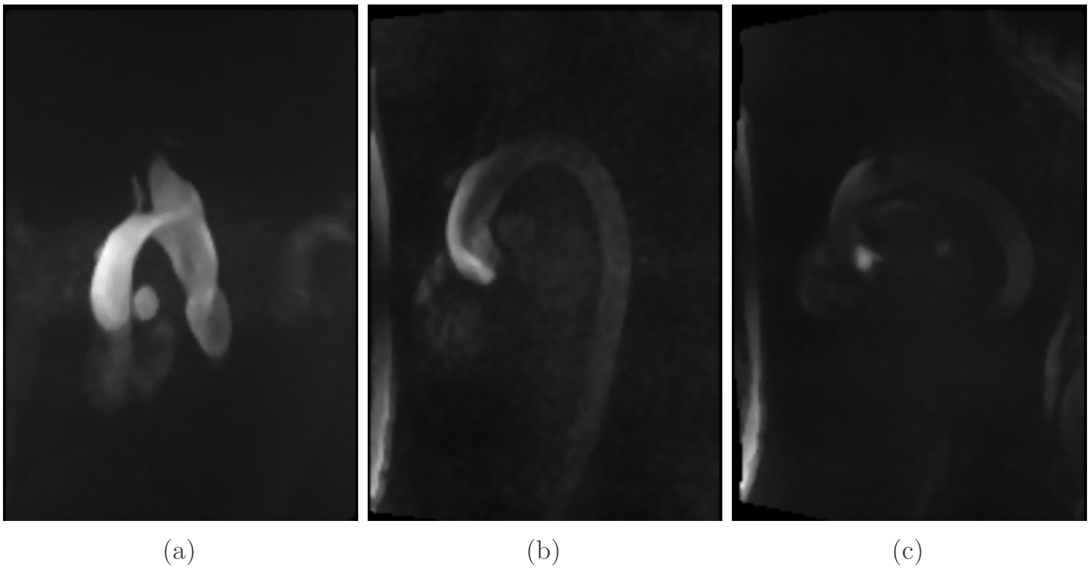
Exploration and Comparison of Blood Flow Data

4D PC-MRI Segmentation Using FTLE Fields 4.1

The content of the following section is based on:
Benjamin Behrendt, Benjamin Köhler, Daniel Gräfe, Matthias Grothoff, Matthias Gutberlet, and Bernhard Preim. "Semi-Automatic Vessel Boundary Detection in Cardiac 4D PC-MRI Data Using FTLE fields". In: *Eurographics Workshop on Visual Computing for Biology and Medicine.* The Eurographics Association, 2016

As mentioned before, an accurate visual representation and extraction of quantitative hemodynamic information hinges on having a good segmentation of the underlying medical data in most cases. Methods range from simple thresholding or region growing [76] to more sophisticated techniques such as level sets [190], watershed transformations [112] or graph cuts [99, 191, 192]. The segmentation of 4D PC-MRI data can be challenging, in particular when segmenting low-contrast images (Figure 4.1). In such cases, the segmentation usually requires extensive time-consuming and exhausting manual input. Since the differences in local contrast make the TMIP images unsuitable to base a segmentation on, we propose the additional usage of directional information. Flow inside of a blood vessel generally follows the vessel's course and is somewhat coherent, while any movement perceived outside results from noise and is therefore random. Thus, the presence of a general flow direction distinguishes the vessel anatomy from their surroundings, providing additional valuable information for segmentation purposes (recall Section 3.1).

This distinction is quantifiable using the Finite Time Lyapunov Exponent (*FTLE*) [114], which is a measure of the rate of separation of infinitesimally close trajectories in a



(a) (b) (c)

Figure 4.1: *Comparison of TMIP image slices showing the aorta with different contrast levels. While one image (a) can easily be segmented on image intensity alone, segmenting the others (b, c) is challenging due to the low contrast in the aortic arch and descending aorta.*

specified time frame of a dynamic system. The lack of coherent flow outside of the anatomy results in a high rate of separation in the direct vicinity of the vessel boundary, as this is where directed flow borders on random motion. FTLE fields approximate Lagrangian Coherent Structures (LCS), which are trajectory structures in a dynamical system that indicate the presence of physical boundaries or other major influences on the flow [115]. Therefore, areas of high intensity in an FTLE image can be interpreted as an approximation of the vessel boundary [117].

To improve the segmentation process specifically on 4D PC-MRI images with low overall contrast, we have developed an algorithm that combines FTLE and magnitude information into a new type of image. Using this image as a base for segmentation allows us to both significantly speed up the segmentation process and increase the segmentation quality in comparison to classical methods based solely on magnitude.

FTLE image calculation

The base for an FTLE field calculation is a flow map $\Phi_t^{t+\delta t}(v)$ [116], which can be calculated from the directional images. It maps each voxel $v = (x, y, z, t)$ in the four-dimensional dataset to the position a massless particle integrated from v at time step t would have at the time point $t+\delta t$. Parameters of this process are step size and integration time δt. The step size controls the number of sampling points for the integration process, thus increasing the step size can increase the calculation speed at the cost of accuracy. The optimal integration time varies with each dataset, as it is dependent on variables like overall flow speed and turbulences [193].

In areas with coherent flow, neighboring voxels in the flow map should reach a similar final integration position. Due to the coherent flow of the vessel in contrast to the random noise outside, the aorta's shape would be clearly distinguishable in this image. To obtain the actual FTLE value of each voxel v at time t (Eq. 4.3), its spatial gradient or Jacobian $J(v, t, \delta t)$ is needed (Eq. 4.1) [117]. It represents the separation of flow around v in all directions and is approximated by calculating the central differences with all neighboring voxels. By calculating its Euclidean or Spectral norm, it can be quantified as a single, scalar value. $\lambda_{\max}$ in Eq. 4.2 is the maximum eigenvalue of the matrix J^{TJ}. The logarithm in Eq. 4.3 is applied to account for the exponential growth of this term. Additionally, its numerical stability towards changes of the integration time δt is increased by performing a normalization [117].

$$J(v, t, \delta t) = \nabla \Phi_t^{t+\delta t}(v) \tag{4.1}$$

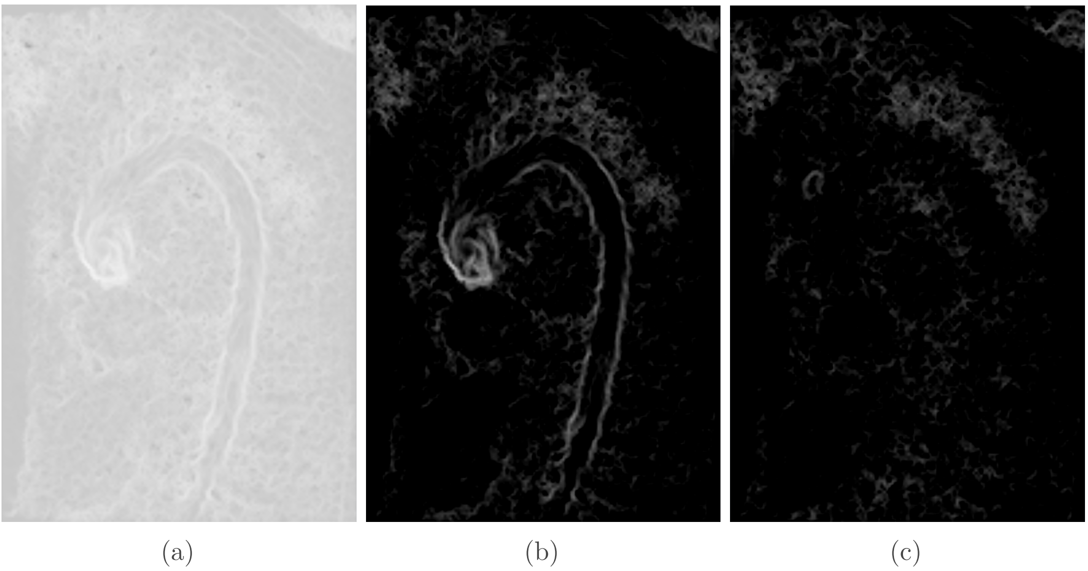
$$\lambda(v, t, \delta t) = \sqrt{\lambda_{\max}(J(v, t, \delta t)^T J(v, t, \delta t))} \tag{4.2}$$

$$FTLE(v, t, \delta t) = \frac{1}{|\delta t|} \log(\lambda(v, t, \delta t)) \tag{4.3}$$

Due to the absence of a heuristic to determine the optimal integration time before actually performing the integration, we opted for a fixed value of 20 % of the dataset's overall time frame (100 - 120ms). The FTLE calculation is performed for each voxel and time step, thus the resulting FTLE image is three-dimensional and time-resolved. Since our goal is to create a static segmentation, we can further increase the FTLE image contrast by sacrificing the temporal information and applying a temporal maximum intensity projection. As a measurement of flow separation, FTLEs are strongly dependent on the time frame they are generated over (defined by t and δt), so there is no need to calculate them over time frames without significant flow. Therefore, the diastolic phase can be skipped during FTLE calculation (Figure 4.2c). Since we are only interested in the ridges of the FTLE image, a normalization above the 80 % quantile was applied to the final image (Figure 4.2a and 4.2b).

In addition to boundaries, certain inset flow structures can also cause high FTLE values. For instance, strong vortices or inflow jets can sometimes be hard to differentiate from the actual vessel boundaries (Figure 4.3b). Also, FTLE images are sensitive to noise, which is likely to appear in air-filled areas. In order to increase the robustness of our approach against such structures, we use additional information from the TMIP and flow data to reduce or eliminate their effects.

Although the gradient of the TMIP images alone may not be sufficient to detect vessel boundaries (Figure 4.3a), it can still help to rule out noise that is detected as unwanted boundaries in the FTLE image. By normalizing both the TMIP gradient and FTLE image to an intensity range of 0 to 1 and multiplying them, we can enhance the contrast of the actual vessel boundary. In both the gradient and FTLE image, the actual boundary should have comparatively high values, while most inset flow structures are only visible in the FTLE image. This creates an image with high intensities on the vessel boundaries, but also a high amount of noise, especially in the lung (Figure 4.3d). As proposed by Walker et al. [194], we generated the STDEV image by adding up the flow's temporal standard deviation over all time steps for each voxel to remove these artifacts. Areas with an exceptionally high standard deviation are most likely to be air-filled regions and thus cannot be part of the vessel anatomy (Figure 4.3c). They can therefore be filtered out by multiplying our image with the inverse of a normalized STDEV image. In the resulting image, which we will reference as Enhanced FTLE (*EFTLE*), vessel boundaries are clearly visible with minimal noise (Figure 4.3e).



(a) (b) (c)

Figure 4.2: *FTLE slice generated for a time frame over the systole with (b, c) and without (a) normalization above the 80 % quantile, diastolic FTLE frame with normalization (a).*

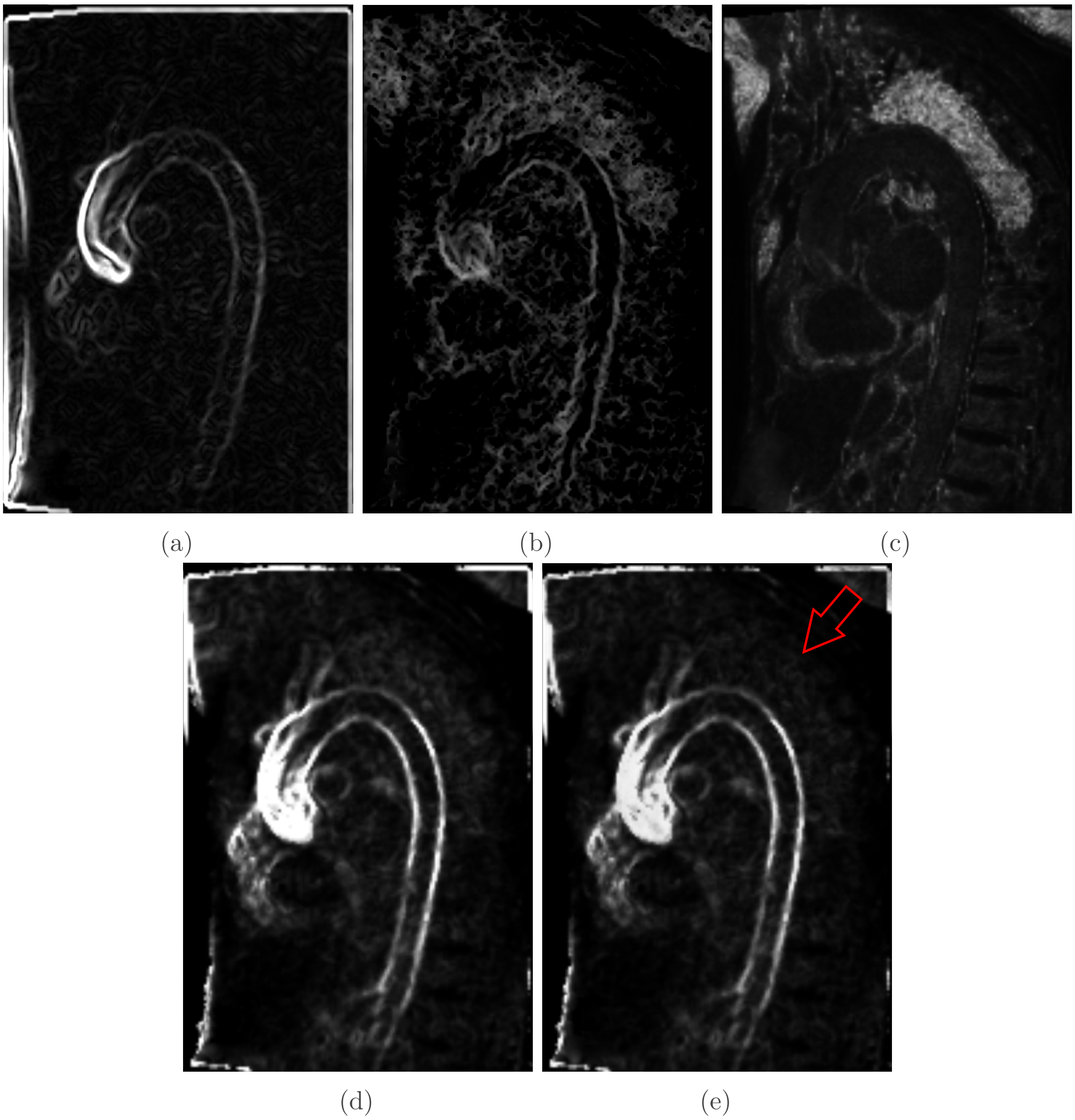


Figure 4.3: Slices from the input and output data; TMIP gradient (a), FTLE (b) and STDEV (c) input images. The last two images show the result excluding (d) and including (e) the STDEV image. The red arrow highlights an area, where including the STDEV reduces noise.

Implementation

After acquiring and preprocessing the dataset, our application generates the necessary derived images like TMIP and flow map using parallel processing on the GPU. Generating the flow map takes the most time, as it requires to integrate the flow for each voxel in every time step. To approximate the flow trajectory during the flow map calculation, we are using the fourth-order Runge-Kutta integration (*RK4*), with a step size of 1, which offers an acceptable compromise between accuracy and speed.

The actual segmentation is performed using the 3D graph cut implementation GridCut, with edge weights specified by the function $e^{-\alpha \cdot ||\nabla I||^2}$. I is the image intensity and α a tolerance parameter with an experimentally determined value of 1000 [192]. The user has to manually classify small regions using a paint brush tool as belonging to either the vessel anatomy or background (Figure 4.4). As GridCut supports 3D graph cuts, not all individual slices have to be painted onto by the user. After each manual input, GridCut is executed in a background thread to generate the segmentation, facilitating an interactive 2D segmentation preview. At any time, the user can also generate the 3D surface mesh from the segmentation to identify and correct regions with segmentation errors. This mesh is extracted using the Marching Cubes algorithm, decimated using the quadric decimation algorithm by Hoppe et al. [195] and smoothed using a low-pass filter, as suggested by Taubin et al. [196].

Results

To evaluate our method, we asked an expert radiologist to use it as a base to perform segmentations on. We used four clinical datasets (*A*-*D*, Figure 4.5) showing the aorta and pulmonary artery of healthy volunteers as well as patients with different pathologies. These specific datasets were chosen out of a larger database as representatives of different contrast levels. Each dataset consists of six images with a grid resolution of 132 × 192 × 15 - 26 for each of their 11 to 23 time steps, containing flow direction and magnitude in x, y and z direction. From each of these datasets, we constructed magnitude TMIP, STDEV and FTLE images and combined them into EFTLE images. All preprocessing tasks were performed before the radiologist started segmenting the images. On an *NVidia Geforce GTX980*, generating a flow map without the diastolic time frame took around one to three minutes, depending on the dataset dimensions.

The radiologist segmented all datasets twice, once based on TMIP and once on EFTLE images. To objectively evaluate our approach against classical methods, we compared both required input for and resulting vessel model from both segmentations. The quantification of user input consists of counting how many voxels the radiologist had to manually color in order to reach a satisfying segmentation. In order to make the results comparable between different datasets with varying resolutions and vessel volumes, we calculated the ratio of manually segmented voxels in TMIP and EFTLE

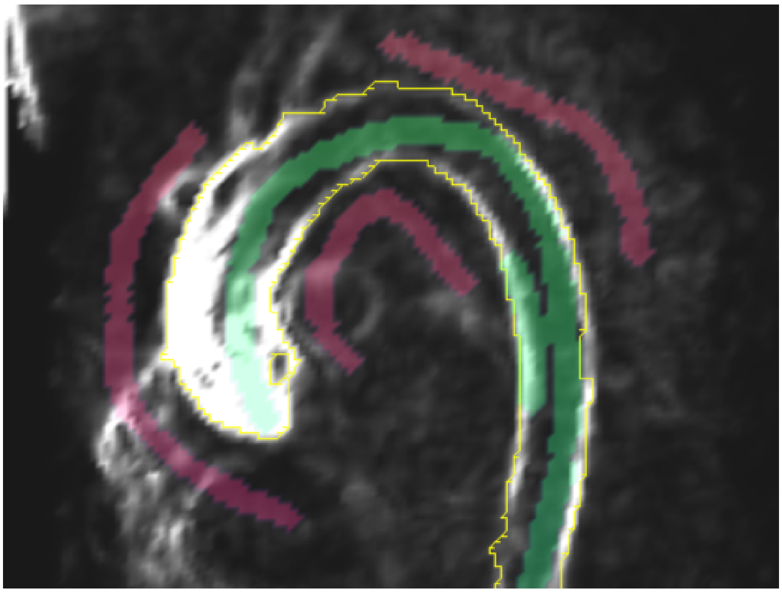


Figure 4.4: *Segmentation process using graph cuts. The user manually draws foreground (green) and background (red) regions into the slice images, which are then used as input for the graph cut algorithm, producing a segmentation (yellow outline).*

images. To ascertain that the user input for segmenting the EFTLE images would not also generate a good segmentation on a TMIP, the input for both EFTLE- and TMIP-based segmentations was then reapplied to the other image. We additionally generated Local Phase Coherency (*LPC*) images [111] and applied the EFTLE-based segmentation input to them to compare the respective segmentation results.

During an informal interview, the radiologist expressed that especially for the datasets with poor contrast, EFTLE images were much easier to segment because the anatomy was much clearer to see (Figure 4.6b). This proved especially useful for *Dataset B* and *Dataset C* (Figure 4.5), as the aorta does not align exactly with any of the image axes in most parts. The increased visibility allowed for a segmentation of additional parts of the ascending and descending aorta, as seen in Figure 4.7. The TMIP-based segmentation of the dataset in Figure 4.7a is missing parts of the descending aorta that were not sufficiently visible in the magnitude images. In Figure 4.7c, the radiologist was unable to segment parts of the ascending aorta around the left ventricle. Both of these areas were made possible to segment using EFTLE images (Figure 4.7b and 4.7d). Additionally, Figure 4.7d shows a better segmentation of the vessels branching from the aortic arch, which are important landmarks in many applications. A reoccurring problem with segmenting EFTLE images is that the segmentation can leak from the aorta onto the pulmonary artery, since the graph cut has problems separating them in some areas. On the low-contrast datasets, however, the effort to manually correct these errors was significantly lower than the effort to create a magnitude-based segmentation.

Figure 4.8 shows an overview of the ratio of foreground and background voxels the expert had to manually mark on our EFTLE images in comparison to TMIP images to order for the graph cut algorithm to produce a satisfying segmentation. Thus, a value below 1 means that less manual input was required for the segmentation using EFTLE images, a value above 1 denotes a higher amount of manual input. Due to the lack of a gold standard for the segmentations, we did not compute similarity measures such as the DICE coefficient. The two images with better contrast (*Dataset A* and *Dataset B*) exhibit higher ratios, showing that our method was less efficient on them. Due to the aforementioned leaking of the segmentation into the pulmonary artery, *Dataset A* has an exceptionally high ratio for manually marked background voxels. On images with lower contrast levels, our approach performs significantly better, as it reduces the amount of required manual input by 22 to 32 percent.

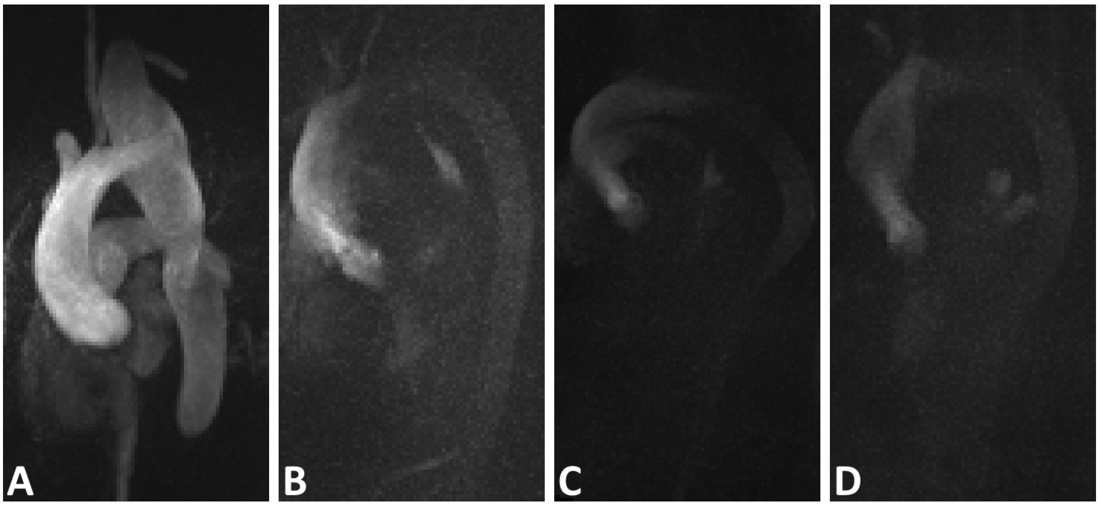


Figure 4.5: *Datasets used for the evaluation (*A-D*)*

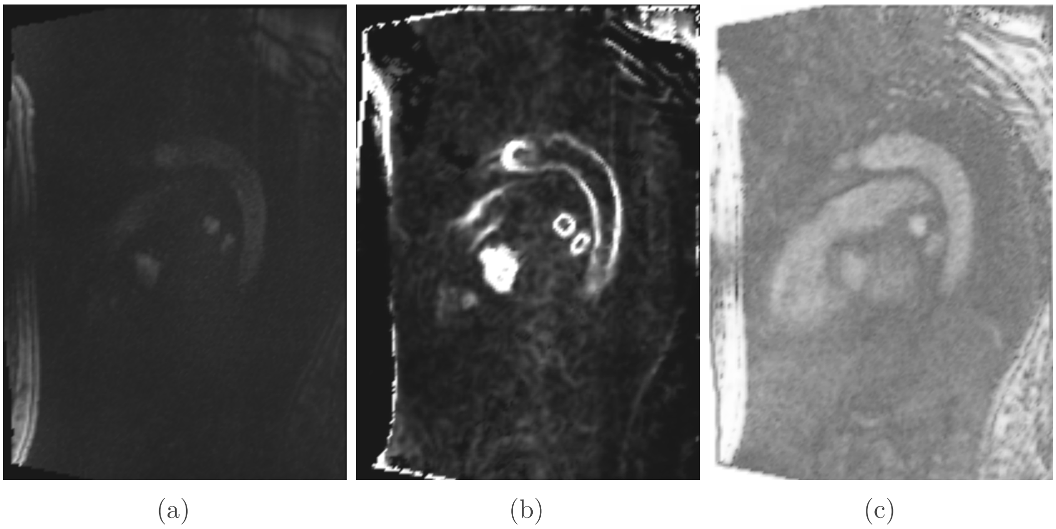


Figure 4.6: A slice rendered as TMIP (a), EFTLE (b) and LPC (c).

Additionally, we applied the segmentation input for EFTLE images to TMIP images and in reverse (Figure 4.9). On datasets with better contrast, exchanging the input produced mostly valid segmentations in both cases. Applying the EFTLE input to TMIP images generally causes smaller areas to disappear from the resulting segmentation. In the reverse case, additional undesired areas belonging to the pulmonary artery become visible (Figure 4.9b). The same things happen on the datasets with lower quality, although the effects are strongly increased. Here, applying the EFTLE input to TMIP images fails to produce a valid segmentation, because there is too little input for the graph cut to work with. This had to be solved by marking additional voxels as background, which still results in an unsatisfactory segmentation (Figure 4.9c). Applying EFTLE input to LPC images produced a seemingly valid segmentation for all four datasets. On closer inspection, however, the vessel segmentation turned out to be incomplete. The graph cut algorithm fails to automatically include many voxels near the vessel boundary, likely due to generally lower SNR of LPC images (Figure 4.10).

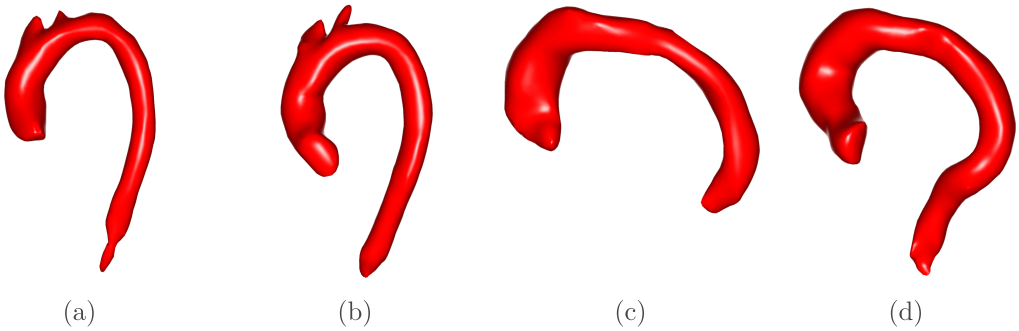


Figure 4.7: Vessel mesh extracted from the segmentation of Dataset B *(a, b) and* Dataset C *(c, d); the vessels generated from a magnitude-based segmentation (c, a) cover less of the anatomy than those generated from our method (d, b).*

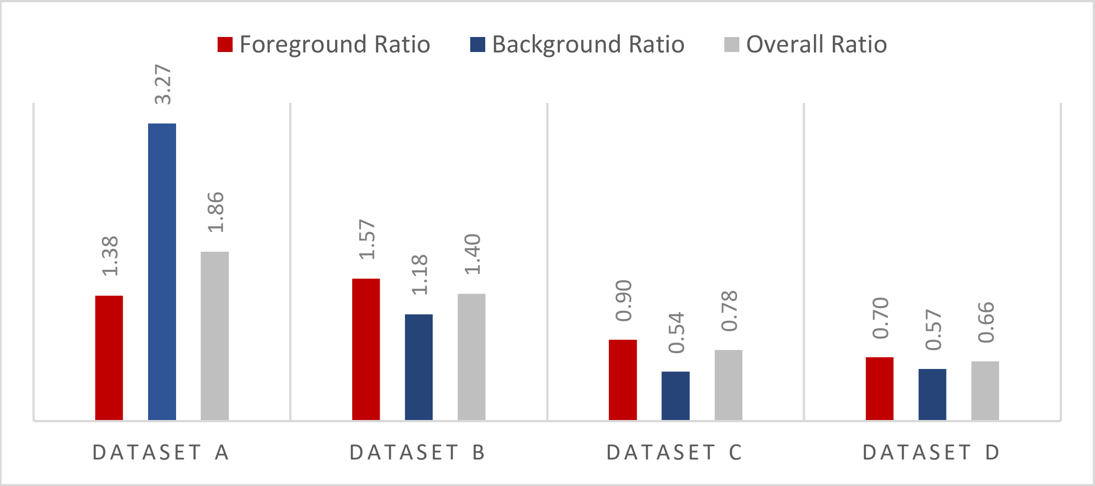


Figure 4.8: *Ratios of the voxels our expert had to manually segment on our EFTLE images in comparison to the TMIP-based segmentation.*

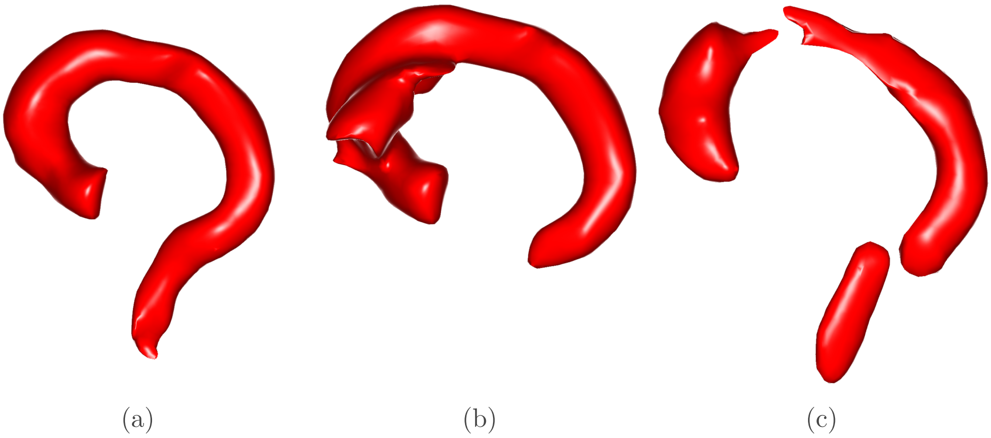


Figure 4.9: *Segmentation results after interchanging the input for TMIP and EFTLE images. Reference segmentation (a), TMIP input applied to EFTLE image (b), EFTLE input applied to TMIP image (c).*

Discussion and Future Work

We presented a method to aid the segmentation of vessels in low-contrast cardiac 4D PC-MRI datasets. This was achieved by combining magnitude-based images with flow coherency information extracted from FTLE fields. Although we only tested it with cardiac 4D PC-MRI data, our method should be easily adaptable for other regions of the human body, as long as the vessels are large enough to be captured by 4D PC-MRI. Similarly to LPC images, the resulting EFTLE images allow for a segmentation of the vessel even in areas with low magnitude contrast or signal-to-noise ratio. With the help of an expert radiologist we were able to confirm that our approach requires less input to generate a satisfying segmentation than an LPC-based method. While our approach works better on low-quality images, using the TMIP as a base for segmentation was preferable on high-contrast datasets. Therefore, users should be able to switch between these two options, depending on the quality of their datasets. Alternatively, a histogram analysis could be performed to automatically switch between EFTLE and TMIP segmentation.

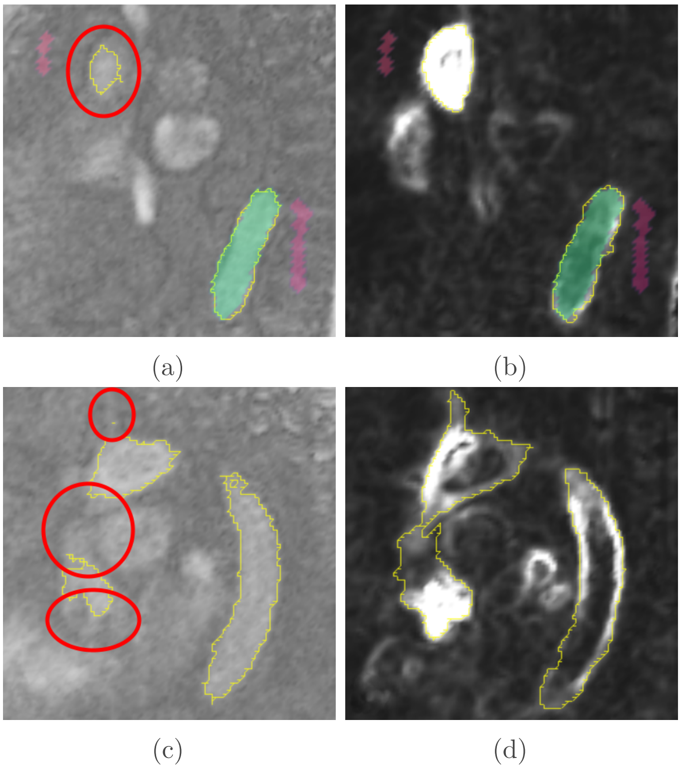


Figure 4.10: *Comparison between LPC- and EFTLE-based segmentations (yellow outline) using the same input; Red circles in the LPC images (b, d) indicate areas with incomplete segmentation compared to EFTLE images (a, c).*

Further improvements to our approach could be made by a more detailed exploration of the algorithm's parameter space, namely the step size, integration time and scale of the FTLE image. A dynamic adjustment of these parameters, based on the TMIP image contrast, could potentially allow our approach to perform well even on high-contrast images. Thus, further studies need to be conducted in order to find a heuristic for determining an optimal parameter set for each individual dataset.

Perception-oriented Vessel Visualization 4.2

Visualizing complex vascular shapes together with pathlines on a two-dimensional screen naturally comes with various challenges. Common issues are a lack of depth perception and overlapping between the vessel wall and inlying pathlines. In this section, we will present our solutions to these problems.

Enhancing Visibility of Blood Flow in Volume Rendered Cardiac 4D PC-MRI Data 4.2.1

The content of the following section is based on:
Benjamin Behrendt, Benjamin Köhler, Uta Preim, and Bernhard Preim. "Enhancing visibility of blood flow in volume rendered cardiac 4D PC-MRI data". In: *Bildverarbeitung für die Medizin.* Springer, 2016, pp. 188–193

This work is an extension of my Master's Thesis [198]. For the paper publication, I added the minimum structure thickness to suppress noise and conducted the expert interview.

As previously mentioned, a segmentation is a precondition for many vessel visualization techniques. However, to allow for a quick explorative visualization, it can be beneficial to use approaches which do not require a pre-processing or segmentation of the data. A use case for such an algorithm is the generation of preview images or videos for a dataset that has just been acquired. Direct Volume Rendering (*DVR*) is a visualization technique which allows displaying a volume dataset without needing a segmentation. The visualization is controlled by a user-specified transfer function, which can be manipulated on the fly to allow for fast exploration of the dataset. However, problems can arise when objects, such as pathlines, representing the intravascular blood flow are visualized in addition to DVR. Using a simple intensity-based transfer function, the pathlines would either be hidden behind the vessel wall or the volume rendering would have to be drawn entirely on a background layer, drastically reducing the perception of spatial relations between the flow and surrounding anatomical structures.

There are a variety of methods to solve overlapping problems, ranging from simply cutting out the entire context to geometric calculations identifying obstructing parts of overlapping features and removing them [150]. These techniques are often summarized as *smart visibility techniques*. However, they seem unsuitable for depicting blood flow obtained from 4D PC-MRI using path lines. Since these path lines occupy nearly all of the space inside the vessel's context anatomy, rendering geometric calculations to find overlapping parts would be superfluous. On the other hand, simply removing the entire context may reduce the viewer's spatial awareness, making it harder to estimate position and size of the focus objects.

Therefore, we propose an automated ray sampling algorithm to open up any context structure and show inset focus objects, achieving a similar effect to frontface culling on indirect volume visualizations. Our approach reduces occlusion problems between vessel wall and blood flow in a Direct Volume Rendering of 4D PC-MRI data. The visibility of focus objects inside the anatomy is guaranteed while spatial awareness is mostly maintained due to the presence of anatomical structures as context information. Furthermore, as our sampling technique does not make any assumptions about the transfer function, our approach can be combined with existing volume rendering techniques such as closest vessel projection.

Implementation

Our visualization approach encompasses two parts:

1. The blood flow, which is represented by pathlines and rendered as normal 3D geometry.
2. The vessel anatomy, which is important as context information and rendered using Direct Volume Rendering.

We decided to use a closest vessel projection [199] approach for our volume rendering, whereas the sampling along the ray is stopped once it hits the first local intensity maximum above a given threshold. Additionally, we use the local magnitude gradient at this position to calculate Phong lighting to support shape perception.

As our goal is to produce a fast, explorative visualization, the pathlines representing blood flow are not pre-computed. Instead, they are generated from particles with a uniform seed distribution inside the anatomy, which move according to the underlying time-resolved flow field and draw a trail behind them to form pathlines. To avoid placing particles outside of the anatomy, a mask defining valid particle positions is created by applying a threshold to the original TMIP. This threshold is linked to the transfer function used for the anatomy rendering, thus the mask only contains voxels that are not rendered completely transparent. This is similar to an approach by Stalder et al. [200], but instead of drawing the path lines on a layer in front of MIP visualization of the data volume, we spatially integrate both of these types of visualizations.

The composition of DVR anatomy and geometry-based path lines is accomplished by comparing the current depth during ray sampling with the OpenGL depth buffer generated when rendering the lines. The ray sampling is terminated prematurely once a line is hit, i.e. the current sampling depth exceeds the value stored within the depth buffer. Since all particles are placed inside the anatomy, their path lines are generally occluded, as shown in Figure 4.11b. Thus, the intravascular flow can only become visible if the vessel front is culled. If we were using a geometric surface, we could simply calculate the normals for each face and remove those pointing towards the viewer (*frontface culling*). A common way to replace surface normals in DVR, e.g. for the purpose of lighting calculations, is to employ gradients, as they point towards the outside of structures with high intensity (vessels) and high gradient magnitudes appear at structural boundaries. However, using such gradients to facilitate frontface culling would likely only peel off the outer layer of the voxels belonging to a vessel, as within the vessel, intensity gradient magnitudes are generally small and thus may be oriented arbitrarily due to noise.

Instead, our algorithm requires an approximated binary vessel segmentation. Unlike a segmentation used to extract the vessel geometry, e.g. using Marching Cubes, this segmentation can be very coarse and therefore be generated fully automatically. For the screenshots presented in this section, we used a threshold on the temporal maximum intensity projection (*TMIP*), which is set to the median intensity value. During ray casting, our algorithm stores two Boolean values for each ray (Algorithm 1). The first (b_1) represents whether the ray has already hit an anatomical structure, the second (b_2) acts as a switch to control whether encountered voxels are to be rendered or skipped (see Figure 4.11a). For each sampling point on the ray, we check if the current position is part of the anatomy by looking it up in the segmentation. This happens first at point P_1, where the structure is hit for the first time. Now, b_1 switches to *true*. After that, once we encounter a voxel that is no longer part of the anatomy (point P_2), b_2 is flipped to true and the sampling position is set back to a previous point on the ray (point P_3). Only now the intensity values read from the sampling position will contribute to the rendered image. Due to jumping back after leaving the anatomic structure, its back side will be rendered while any other part of it gets culled. The effect of this rendering method in combination with path lines can be seen in Figure 4.11c. By manipulating the distance between P_2 and P_3, the thickness of the rendered back sides can be adjusted. To prevent holes in the visualization, this parameter should be set to a distance that equals at least one voxel in every possible viewing direction.

It is possible for the user to specify a minimum structural size to increase the robustness or our algorithm against noise and other image artifacts. If the ray exits a structure without having traversed this given distance inside of it, the loopback will not occur and b_1 will be reset to *false*, effectively culling the entire structure. It should be noted that using a minimum structural thickness will not only remove artifacts, but also erode all anatomical structures. Therefore, in order to keep the visualization as faithful as possible, the minimum structural thickness should be at most two voxels.

Results

We have tested our algorithm with a set of 20 three-dimensional, time-resolved flow images of the aorta acquired from 4D PC-MRI scans using a 3 Tesla Magnetom Verio MR with a V_{enc} of 150 cm/s. They were acquired from both healthy subjects as well as patients with different pathologies. The datasets consist of six images for each time step, containing flow direction and magnitude in x, y and z direction. They have a resolution of 132 × 192, with 15 to 25 slices and 11 to 19 time steps.

In an informal evaluation, we asked an experienced collaborating radiologist to compare our visualization against one where the path lines were simply drawn in front of a MIP, as shown in Figure 4.12. The dataset in this example was acquired from a patient with a bypass and contains abnormal flow patterns in the form of vortices in the ascending

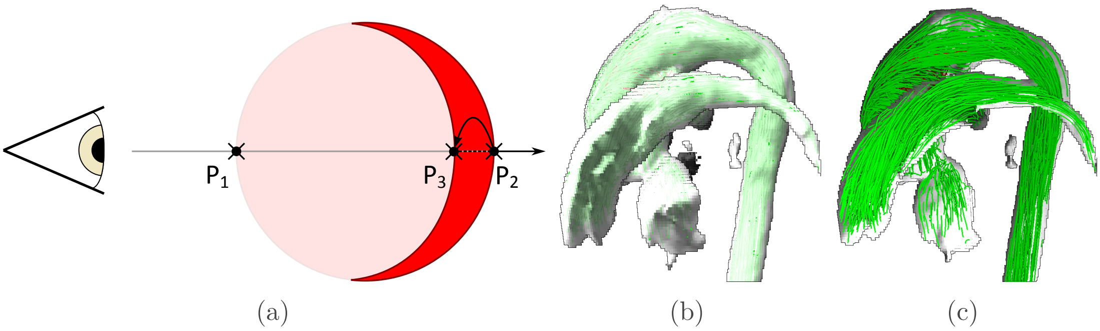


Figure 4.11: Principle of our algorithm. P_1 and P_2 are entry and exit points, respectively, whereas P_3 is the loopback point (a); flow visualization with loopback-based frontface culling disabled (b) and enabled (c).

Algorithm 1 Pseudocode for the processing of a single DVR ray

```
for all $ray$ in $rays$ do
    $b_1 \leftarrow false$
    $b_2 \leftarrow false$
    $d \leftarrow 0$
    $d_0 \leftarrow 0$
    while $d \leq$ LENGTH( $ray$ ) do
        $pos \leftarrow$ START( $ray$ ) $+ d \cdot$ DIRECTION( $ray$ )
        $s \leftarrow$ INTENSITYAT( $pos$ )                          ▷ Sample from image
        if $b_1$ and $b_2$ then
            ACCUMULATE( $s$ )                                        ▷ Apply transfer function
        else if $b_1$ and not INSEGMENTATION( $pos$ ) then
            if $(d - d_0) > min\_distance$ then                     ▷ Ensure min. structure size
                $b_2 \leftarrow true$                                ▷ Set second switch
                $d \leftarrow d - jump\_distance$                    ▷ Jump back
                continue
            else
                $b_1 \leftarrow false$                               ▷ Reset both switches
                $b_2 \leftarrow false$
            end if
        else if INSEGMENTATION( $pos$ ) then
            $b_1 \leftarrow true$                                    ▷ Set first switch
            $d_0 \leftarrow d$                                       ▷ Store current pos. on ray
        end if
        $d \leftarrow d + sampling\_distance$
    end while
end for
```

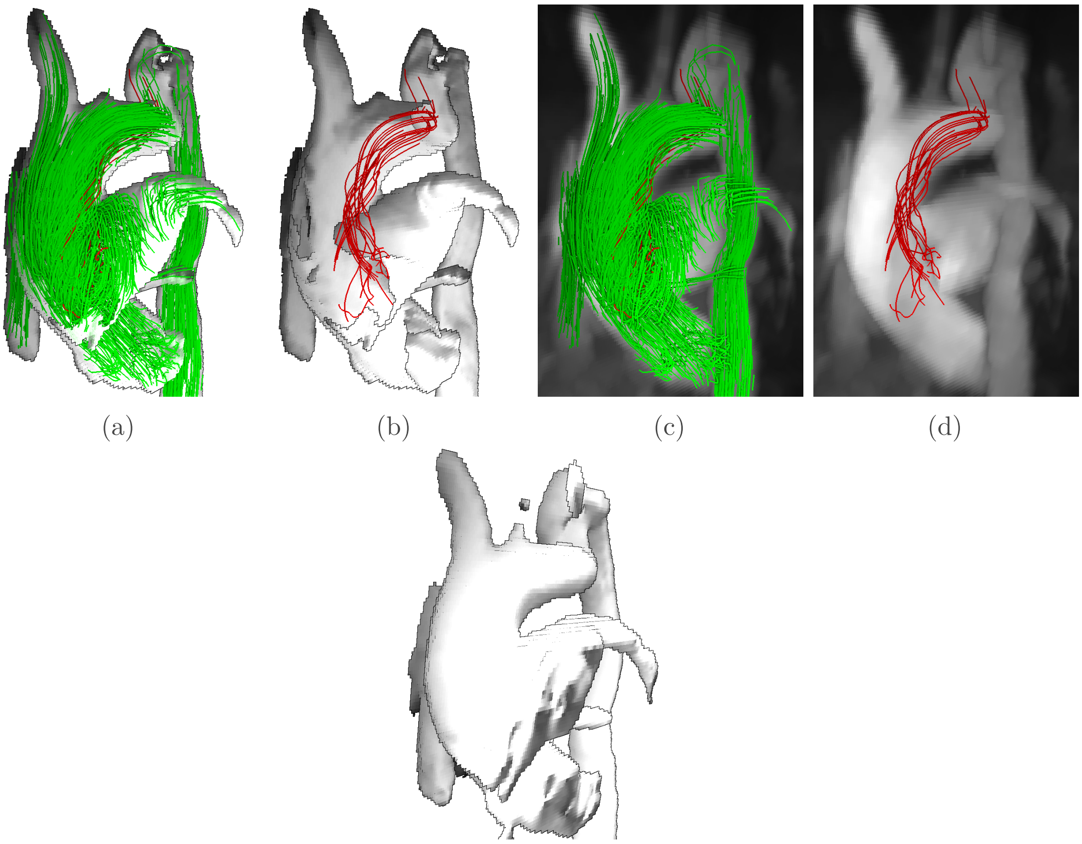
aorta and aortic arch. In both visualizations, the red highlights are best noticeable when hiding laminar flow (*green*). With only the vortex visible, the lack of spatial information in the MIP visualization makes it difficult to pinpoint its exact location. Using our approach, on the other hand, the vessel structure is still clearly recognizable, allowing for easier localization of the vortex.

The expert was asked to compare the visualizations with respect to the visibility of path lines and the ability to pinpoint their location inside of the vessel and confirmed that, in our visualization, the vessel's inner surface were clearer to see. It was also easier to judge the distance between path lines and vessel boundaries. To understand the overall vessel shape, however, the MIP visualization was more suitable, because even parts with lower contrast in the dataset were visible here. This was partially remedied by the ability to disable frontface culling at will, resulting in a more intuitive and complete model of the vessel. Still, it was noted that the ability to display the dataset using MIP should be kept as an option.

Discussion

We have presented a technique to solve overlapping issues when merging volume rendering with geometry visualization, emphasizing inset features while retaining spatial context information. Although we focussed on displaying blood flow in 4D PC-MRI data, the approach can be applied to any volume dataset as long as there is a way to obtain an approximate segmentation of the anatomy. Our method allows for a fast exploration and qualitative analysis of the dataset and containing flow patterns without the need of time-consuming manual segmentation. Since only the front sides of the vessels are removed, spatial information remains available.

Our approach does not make any assumption of the nature of the transfer function and sampling strategy used for volume rendering, besides that it needs to retain depth information to allow for a composition of the image with the geometry-based pathlines. It can therefore be used in conjunction with existing volume rendering approaches, such as windowing or closest vessel projection. Problems with our method arise from data quality as well as from using a TMIP to visualize the vessel anatomy. When two neighboring voxels have similar blood flow speeds, the TMIP does not generate a gradient between them even if they belong to different anatomical structures. Therefore, our algorithm will consider these intertwined structures as a single large structure and not properly visually separate them. These artifacts could be reduced by incorporating the flow information into the boundary detection, e.g. by using an FTLE or LCS image instead of the TMIP gradient. However, the increased processing time required to generate these images could be considered as a downside to an approach that is mainly supposed to deliver a quick preview of the data.


(a) (b) (c) (d)

(e)

Figure 4.12: *Blood flow inside the aorta and pulmonary artery rendered using our algorithm (a, b) and a MIP (c, d), in each case with (right) and without (left) laminar flow (green). Vortices are highlighted in red. A visualization of the vessel anatomy is provided for reference (e).*

Combining Pseudo Chromadepth Enhancement and Parameter Mapping for Vascular Surface Models 4.2.2

The content of the following section is based on:
Benjamin Behrendt, Philipp Berg, Bernhard Preim, and Sylvia Saalfeld. “Combining Pseudo Chroma Depth Enhancement and Parameter Mapping for Vascular Surface Models”. In: *Eurographics Workshop on Visual Computing for Biology and Medicine*. The Eurographics Association, 2017

Regardless whether vascular structures are displayed using DVR or surface rendering, the final result on screen is always a two-dimensional projection of complex, possibly intertwined or overlapping three-dimensional objects. Thus, a large set of techniques have been developed especially for vessel visualization, including surface and volume rendering, illustrative techniques and model-based techniques. Some of these techniques are carefully adapted to shape or depth perception by using special color scales [129]. The downside of using color scales to convey depth is that the color channel cannot be used to visualize parameters on the vessel wall. This is problematic in blood flow visualizations, as hemodynamic parameters of the vessel wall (such as pressure or wall shear stress) are often mapped directly to the surface using color scales.

Common techniques to convey depth using color include chromadepth or pseudo-chromadepth (*PCD*), recall [134]. In this work, we examine if this type of encoding can be combined with additional visualization techniques to increase depth perception. We present a framework for the improved visualization of vessels that features enhanced depth perception in addition to allowing surface parameters to be mapped to the vessel wall using color scales. Our approach uses different color scales on the vessel surface to create separate visualizations for depth and surface parameters, which are then combined using a blending mask. The generation of this mask is inspired by the Fresnel effect, which describes the reflection of a surface based on the viewing angle.

Both chromadepth and PCD assume that the color channel of the image does not contain relevant information and can therefore be utilized to increase depth perception [134]. While this is true for some applications, such as angiography images, it cannot be generalized for every kind of medical visualization task. An appropriate visualization for the exploration of hemodynamic data should therefore convey the general shape and depth of the vessel model, but simultaneously encode the aforementioned functional parameters as well. To reduce mental load, the physician should also be able to compare different regions on the vessel wall regarding their spatial relation and parameter values *without* having to switch between different types of shading. We present a technique that allows the use of PCD in addition to mapping data to the surface color of a model, which is described in the following.

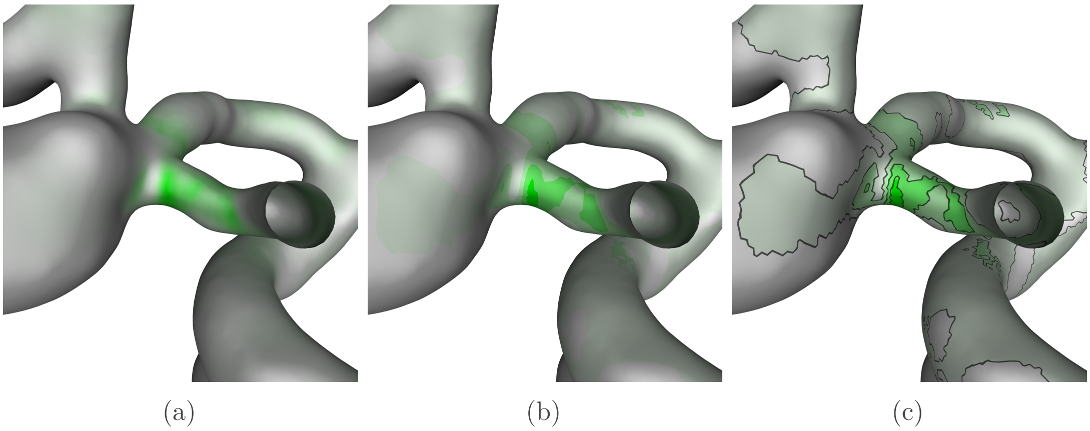
Method

We are applying our technique to cerebral vessel models with attached hemodynamic information. These vessel surface models are generated from 3D digital subtraction angiography data, showing cerebral aneurysms, by applying a threshold-based segmentation. The iso-surface is extracted and converted into a triangle mesh. This mesh is then visualized as a 3D surface model and illuminated using Phong Shading with a single headlight. For the extraction of hemodynamic parameters, the surface mesh is employed for computational fluid dynamic simulations. To map the hemodynamic parameters to the vessel surface, we tested two color scales: a color scale ranging from white to orange in a first study and a color scale from white to green for the final study. We decided against using hatching to convey the additional information, as it may also interfere with the Phong-based lighting we use to convey the shape of the vessels. Furthermore, hatching is not well suited to highlight small areas of interest.

When analyzing surface parameters on a vessel, physicians often look for “hot spots”. These are small areas with very high or low values, which can be perceived pre-attentively when they are encoded with color. To highlight regions with particularly high or low parameter values, we have chosen to discretize the color scale to five different shades (Figure 4.13b). To make these shades even more distinct, a black outline has been added to mark the transition between shades (Figure 4.13c).

Since the green color channel is not occupied by PCD, a trivial solution would be to map hemodynamic information to that specific color channel only. Such a visualization (Figure 4.14) would be unsatisfactory, as not only can the same parameter value have widely different associated hues depending on its location on the model’s surface, but the shade at a specific location can even change as a result of camera movement. It is very difficult for humans to mentally disassemble a composite color into its respective channels. Thus, the interpretation of such a visualization would be error-prone.

Our method displays the PCD color scale on the edges of the 3D model only, based on the current viewing direction. This type of shading is inspired by the Fresnel effect, which describes the amount of reflection and refraction of light on a surface in relation to the viewing angle. A flatter viewing angle on a surface increases the amount of light that is reflected, resulting in the surface appearing brighter when lighted (Figure 4.15a). A physically accurate calculation of this effect is quite complicated, especially when taking into consideration that, due to chromatic dispersion, the strength of the Fresnel effect also depends on the light components’ wavelengths. Instead, we use a simplified version of this effect to generate a mask for overlaying the PCD color gradient. Our Fresnel-Inspired PCD (*FI-PCD*) mask M_{PCD} is calculated similarly to the ghosted views introduced by Gasteiger et al. [151] using the following formula:



(a) (b) (c)

Figure 4.13: *Comparison of a smooth color scale (A), discrete color scale (B) and discrete color scale with additional boundaries (C) when visualizing wall shear stress on a vessel.*

$$M_{PCD} = f_{scale} * (1 - |2 * \frac{\arccos(\vec{I} \cdot \vec{N})}{\pi} - 1|)^s$$

$\vec{I}$ and $\vec{N}$ are the incident and normal vectors at the surface, respectively. The scaling factor f_{scale} can be used to adjust the effect strength. Similarly, s controls the steepness of the transition from surface to PCD color scale. We have empirically chosen a scaling factor of 1 and a steepness of 2. M_{PCD} is dependent on the angle between the normal and incident vectors, reaching its maximum value when they are orthogonal to each other. On spherical or tubular models, the Fresnel effect strongly increases the reflectiveness around the edges of the model (Figure 4.15b). Our final FI-PCD visualization comprises two images, both of them renderings of the vessel surface. The first one has the parameters mapped to its color (Figure 4.16a), the second is colored entirely according to the PCD scale (Figure 4.16b). For each pixel in the final image, the pixel's value in the mask M_{PCD} is extracted and used as weight for the linear interpolation between the two images (Figure 4.16c). For example, black M_{PCD} pixels yield the color-coded parameter value and white M_{PCD} pixels yield the PCD-based color-coding. The resulting FI-PCD visualization (Figure 4.16d) allows mapping a scalar parameter to any color scale, while PCD depth cues are shown only on the edges of the model. They are still clearly visible,

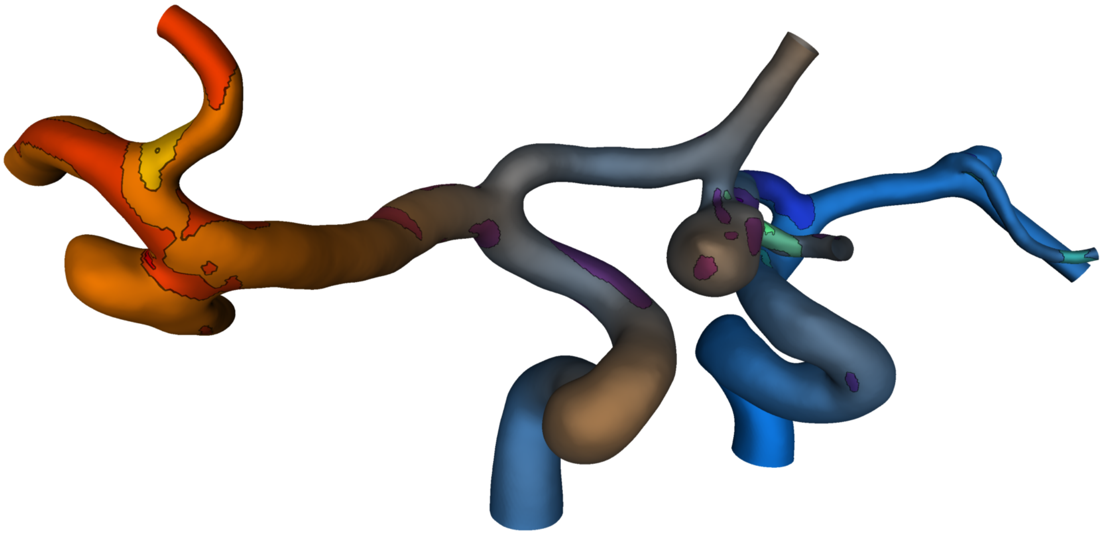


Figure 4.14: PCD shading where the depth is continuously mapped to the red and blue color scale and the scalar parameter is mapped to the green color channel using a discrete scale.

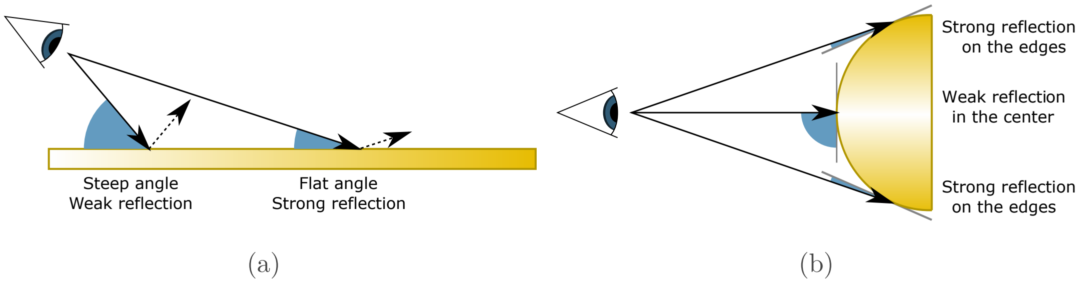


Figure 4.15: Simplified principle of the Fresnel effect; the amount of reflection on a reflective surface depends on the viewing angle (a). When applied to a spherical object, the edges exhibit strong reflections due to the shallow viewing angle (b).

while interference with the object's surface color is reduced. Usually, the physician would rotate the vessel in a way that the interesting areas are facing the camera instead of being relegated to the edge of the model. In addition to providing depth cues by hue, displaying the PCD scale at the edges also increases the perceptibility of overlaps, which is another important depth cue.

Implementation

Generation and composition of both images is performed mostly in the fragment shader. When rendering the surface, the attribute values for the surface are sent to the graphics card as vertex attributes. Additionally, the highest and lowest values of the attribute as well as the highest and lowest depth values from the previously rendered frame are stored in the fragment shader as uniform variables. Then, the attributes are interpolated between vertices, normalized to a $[0, 1]$ range and transformed into a color value by the fragment shader. The transformation is performed linearly in RGB color space between white (*#FFFFFF*) and orange (*#FF7F00*, pilot study) or green (*#00FF00*, final study). Next, the resulting color value is discretized into five distinct shades and used as surface color.

The boundaries between color shades are generated dynamically on a per-triangle base by analyzing the affinity of each vertex to a certain color class. For each triangle with different affinities at the edges, the fragment shader draws a black line separating these vertices. This approach allows for a very fast generation of dynamic outlines on the surface, without the need for any pre-processing or the creation of new geometry. Unfortunately, since the lines are always at the center between two vertices, they do not always line up exactly with the actual color transition. On a model with a decent triangle resolution, this effect is only noticeable when zooming in very closely to the surface.

The second image is generated by normalizing the current fragment's depth using the previously stored depth range and mapping the resulting value to the PCD color scale. Using the depth range from the previous frame allows us to draw the geometry using a single rendering pass, although it produces a barely noticeable flicker in the PCD color scale during fast animations. Afterwards, the M_{PCD} value is calculated and used to compute the composition of both images.

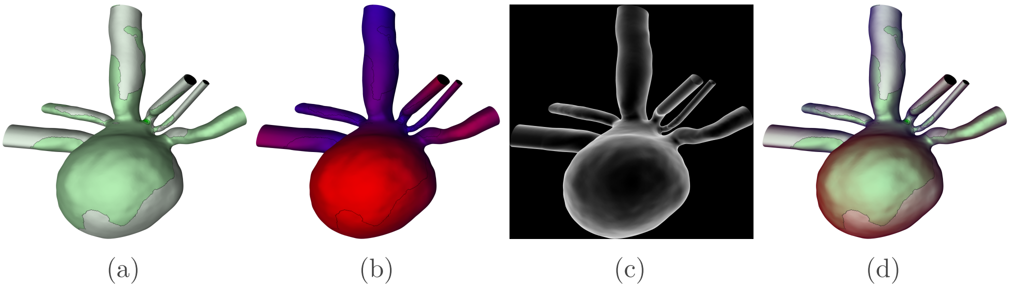

(a) (b) (c) (d)

Figure 4.16: *Composition of images to create the FI-PCD visualization: Surface color image (a), PCD image (b), Composition mask (c) and resulting FI-PCD image (d).*

Evaluation

Our evaluation consists of two separate studies; a pilot study with participants from the general public, and a final study with experts in the fields of medical engineering and flow simulation. All participants were shown 3D visualizations of cerebral vessel surfaces models. These models had one of multiple available attributes mapped to their surface, such as pressure or wall shear stress. Participants were shown two points on these datasets and had to select either the one closest to them or the one with the highest parameter. The datasets were shown with three shading styles:

1. A normal, Phong shaded visualization without any distinct depth enhancement
2. Using brightness as a depth cue, whereas triangles are reduced in brightness by up to 75% based on their distance.
3. Our implementation of FI-PCD.

All three visualization styles can be seen in Figure 4.17. For the purpose of this evaluation, a prototypical application was designed to automate the evaluation process as much as possible. Texts and descriptions in the application were shown in both German and English. When started, the test application presents the user with a few instructional pages. All of them include a “Continue” button that becomes enabled after five seconds and allows the participant to advance to the next screen. The first and second pages contain general information about the study as well as labeled example images for all types of visualizations used in this study. To prevent any bias, these images are always shown in random order. The actual study consists of two blocks, where the user has to select either the point closest to them, or the one with the highest scalar surface parameter. Thus, for two selected points on the surface, the user has to identify their spatial relation or ranking of scalar values (Figure 4.18). Additionally, they always have the option to click a button labeled “Not sure” if they cannot decide for one of the points. During each task, the application measures the completion time, rotation time and whether the user clicked the correct point or hit the “Not sure” button instead. For the rotation time, we counted the amounts of single frames that a rotation was performed

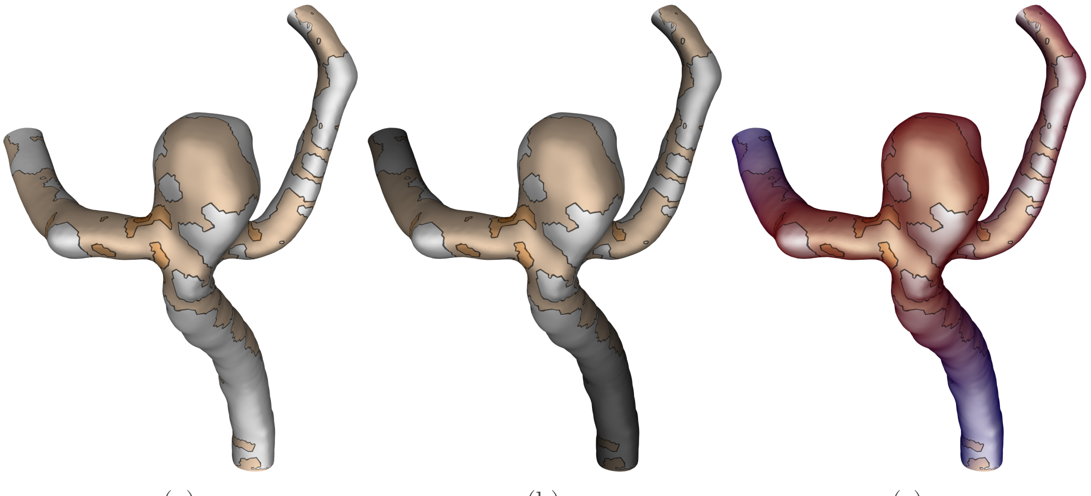


Figure 4.17: The different shadings used in the first study: No depth cues (a), distance-based brightness modulation (b) and FI-PCD (c)

in and converted them to a duration in seconds. Frames where the user kept the left mouse button pressed without moving the mouse (therefore not actually performing a rotation) were not included.

Each block is introduced by another instructional page, which is then followed by six dedicated tutorial datasets. They serve as a way to familiarize the user with the visualizations and tasks, therefore their measurements are excluded from the final statistic. Furthermore, a learning effect during the actual evaluation is prevented. After completing the tutorial for each block, the user sees a message explaining that the training part is over and asking them if they have any questions before proceeding. This was done to ensure they were properly prepared and did not have to ask questions during the time-measured evaluation. They were encouraged to complete each task as fast and accurately as possible due to the time measurement. “Guessing” the correct answer was discouraged in favor of using the “Not sure” button.

The order of blocks was consistent for all participants, starting with the depth judgment tasks and then switching over to the parameter judgment task. Each task consisted of 30 images in total, six of them being the training images. All users were shown the same images, although they were ordered randomly. The application ensured that the same dataset did not appear twice in a row with different shading. Participants did not receive any immediate feedback about the correctness of their answers during the study, but statistics about their general performance were made available to them afterwards upon request.

After completing the assignments on the computer, all participants were asked to fill out a questionnaire. In addition to age, biological gender and known visual disorders, participants were asked if they have experience with analyzing medical data or modeling 3D objects and whether they play 3D video games regularly. For visual disorders, we were mostly interested in those that impede the ability to perceive color or depth. Since there are many cases where people are unaware of their color perception impairment, we added a very abbreviated color blindness test using three Ishihara plates. Two of them had

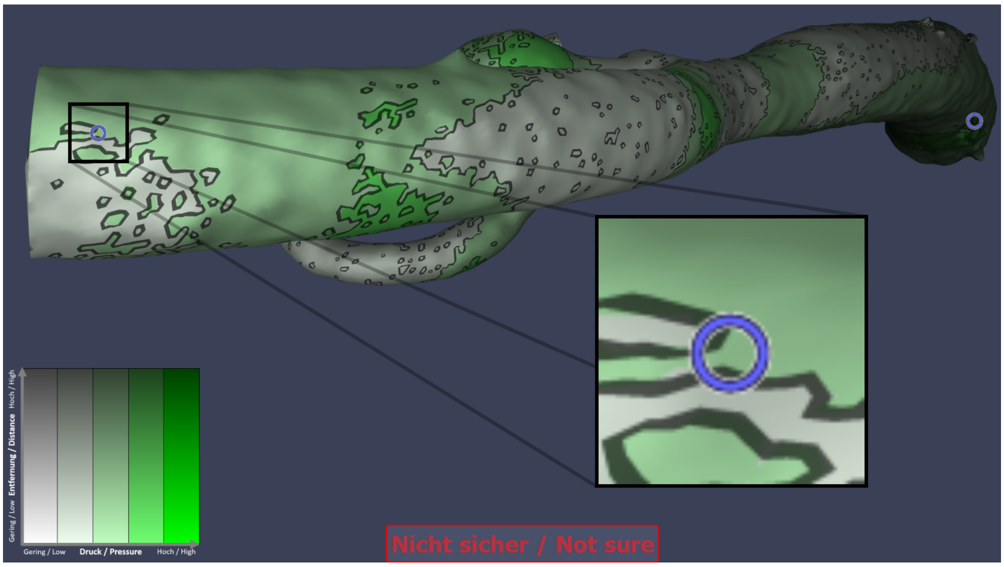
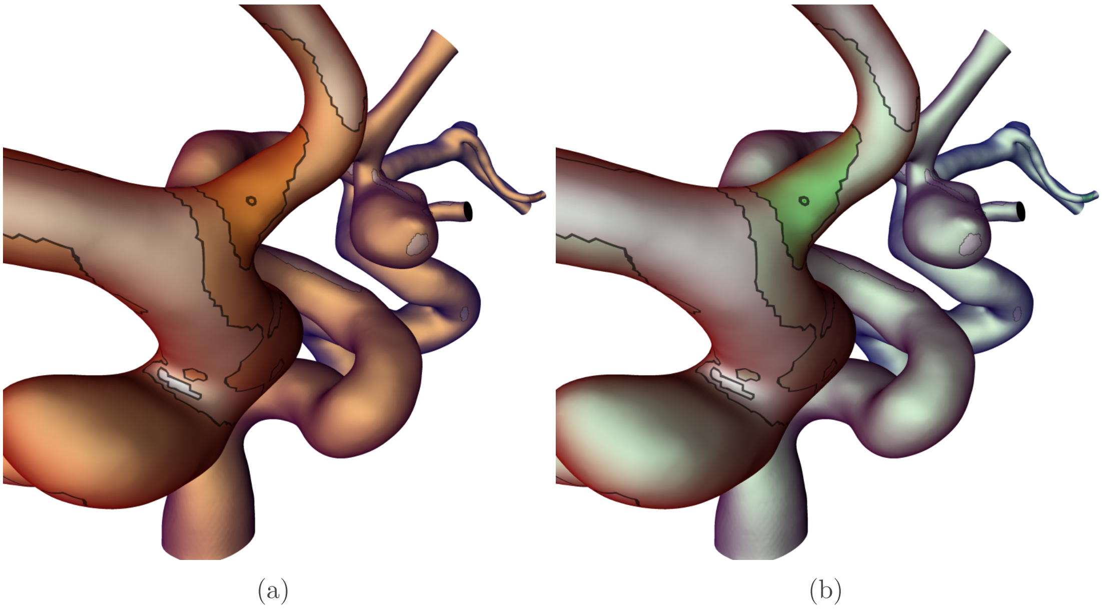


Figure 4.18: *One of the datasets with two marked points shown to the participants as part of the depth judgment task. The image used brightness-based depth cues and the green color scale from the final study.*

numbers encoded in them (42 and 6) while the last one did not. None of the participants who had not already denoted a form of color blindness in the questionnaire failed this test. At the end, participants were asked to rank the three types of visualizations according to their usefulness for perceiving depth and the surface attribute as well as their general aesthetic. They were also given space for any additional remarks. We expected the visualization without depth cues to perform worst in the depth judgment, but best in the parameter judgment task. As both brightness-based cues and FI-PCD would partially overlay the surface color scale, we expected them to perform equally well, but not as good as the visualization without depth cues. Since PCD has proven superior to brightness- or contrast-based cues by studies in the past (such as [138]), we expected FI-PCD to perform best in the depth judgment task.

The pilot study allowed us to identify several flaws in our technique. Before the final study, we corrected these problems by changing some aspects of both our visualization as well as the application. First, the color scale used to encode the surface attribute was changed from white-to-orange to white-to-green. The original orange scale was chosen due to aesthetic reasons. However, many participants in the pilot study noted that red areas from the FI-PCD shading interfered with orange areas from the surface attribute color scale. Since PCD only uses the red and blue color channel, green was chosen for the surface attribute to prevent color overlaps. A comparison between the two color scales in combination with FI-PCD can be seen in Figure 4.19. We also added a permanent legend for the used color scales in the bottom left corner of the screen. This was done in response to some participants in the first study confusing the meaning of some of the colors during the course of the study. The legend always encoded the surface attribute color scale in combination with the current depth enhancement color scale.

Pilot Study: For the pilot study, we took advantage of the popular open house day at our university as a means of finding volunteers. Visitors of this event were asked to participate in our study. A total number of 104 people from the general public volunteered to participate in the pilot study. Ten of them were later rejected due to vision impairments (i.e. various forms of color blindness or problems with depth perception),


(a)

(b)

Figure 4.19: Different color scales used in the first (a) and second study (b) in combination with FI-PCD.

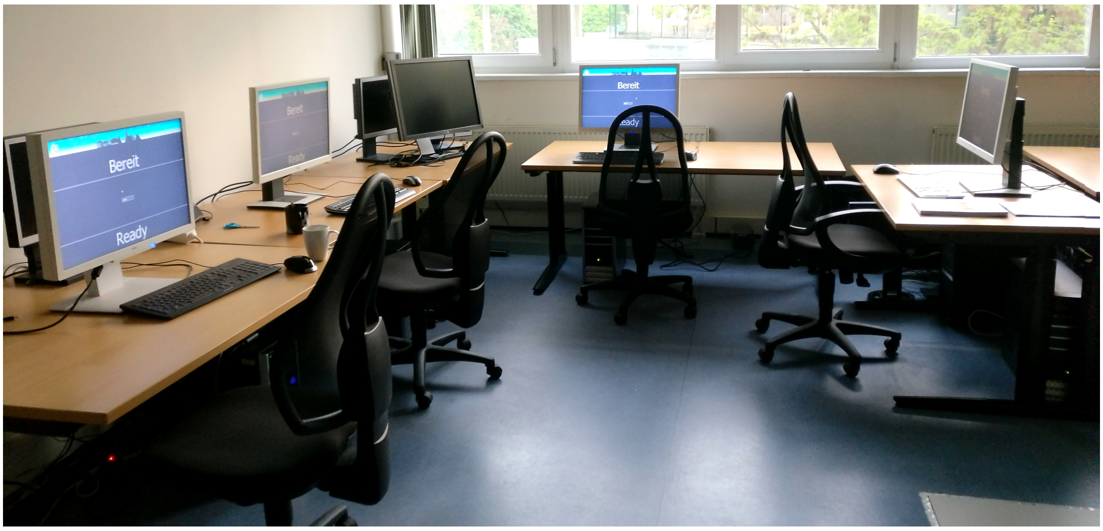


Figure 4.20: Experimental setup for the pilot study with all four PCs showing our applications stand-by screen. The two disabled monitors in the corner were not part of the study.

failing to comprehend the assignment or not filling out the corresponding questionnaire completely. The age of the participants ranges from 11 to 73, with an average of 28.6 and a standard deviation of 14. Out of the 94 participants that were included in the evaluation, 40 were female (42.6 %). In order to allow for a high number of participants, we set up four PCs for simultaneous use. They were positioned in a corner of the room to prevent distractions (Figure 4.20). To ensure comparability of the results between the different stations, we used PCs with similar hardware specifications and identical screens. All stations ran the application at a resolution of 1920 × 1080 with 60 frames per second.

Both verbal explanations as well as written instructions and other materials were made available to the participants. Additionally, they were given a short verbal introduction about the topic of vessel visualization in general and the study in particular. To keep any descriptions simple and explanations short, the different scalar attributes shown in the visualizations were always just referred to as “pressure”, despite also including other attributes such as wall shear stress. After that, they were instructed to sit down at one of the stations and follow the on-screen instructions from the application. Half of the participants were randomly selected to be given limited control over the camera during the study, whereas they can rotate the dataset by ten degrees in any direction. These participants were shown an extra paragraph in one of the instructional pages of the application explaining that they had the ability to orbit the camera. If they did not rotate the camera at least once during the tutorial, they were reminded to do so by a pop-up dialog.

Final Study: The results and feedback we received from the pilot study was used to enhance our prototype for the final study, as previously described. For this study, we directly approached several experts in the fields of medical engineering and flow simulation. Due to the lower number of participants in this study, we decided against splitting them into two groups. Therefore, we allowed all of them to rotate the camera. Eleven experts volunteered to take part in our final study. One person was excluded due to color blindness. The age of the included participants ranged between 22 and 41 (average of 29.1), with two of them being female (20 %).

The second study was performed on a laptop, as it took place at our participant's workplace. Despite having less powerful hardware than the PCs used in the first study, it was able to run the application at a resolution of 1920 × 1080 with 60 frames per second. For this study, the verbal introduction to vessel visualization was either omitted or kept very brief, since most participants were already familiar with this field. The instructions given by the application itself as well as the questionnaire remained unchanged from the pilot study.

Results

Pilot study: For each participant, we calculated four values from our measurement for each shading style. *Correctness* is the percentage of correct answers, e.g. how often participants selected the nearest point (during the depth judgment task) and the point with higher scalar value (during the parameter judgment task), respectively. *Certainty* denotes the percentage of answers where the user selected any of the points and not the "Not sure" button. *Duration* is the average time in seconds the users took for each image. *Rotation* is the average time the user spent rotating the dataset. For this value, we only included users who actually rotated the dataset.

The ability of the participants to pick the point closest to them benefited from having any form of depth cues enabled (Figure 4.21). Without them, they were only correct in 79 % of the depth judgment tasks. Brightness-based depth cues increased their accuracy to 90 %, whereas FI-PCD only increased it to 85 %. This is surprising, as we were expecting the FI-PCD to provide much better depth cues than the brightness-based approach. Although being reminded after each training session that they could rotate, only 35 of the 50 users with the ability to rotate actually made use of it. Three of them performed so little rotation that we assume this interaction to be accidental. This may have been a result of being overwhelmed due to unfamiliarity with 3D visualizations and interaction. Users that stated experience in 3D modeling or 3D video games rotated for an average of 0.3 seconds per dataset, whereas users with no experience only rotated for 0.19 seconds. The values for certainty and rotation are extremely similar for each of the three shading styles. The users generally rarely used the "Not sure" button in this study. The average duration was slightly higher for the visualization without depth cues (4.2 s) in comparison to brightness-based cues (3.8 s) and FI-PCD (4.0 s).

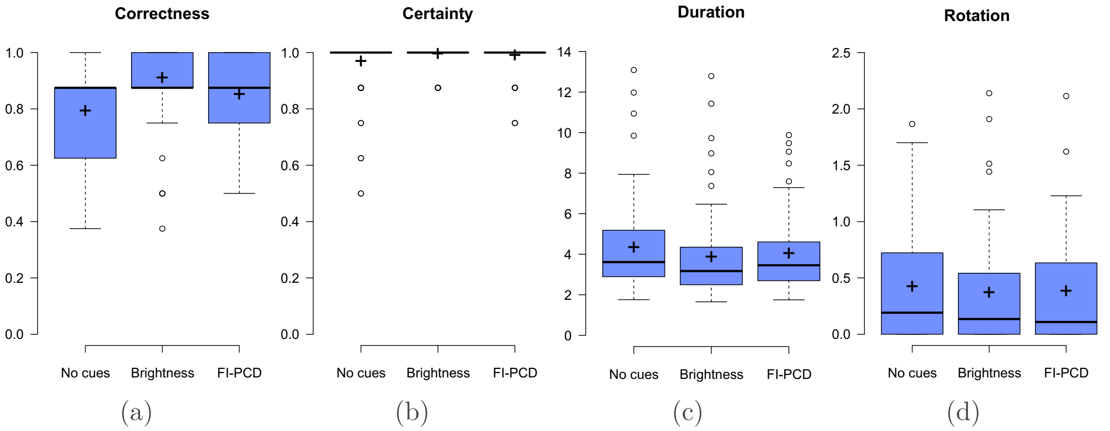


Figure 4.21: *Box plots showing correctness (a), certainty (b), duration in seconds (c) and rotation duration in seconds (d) for the depth judgment task from the pilot study.*

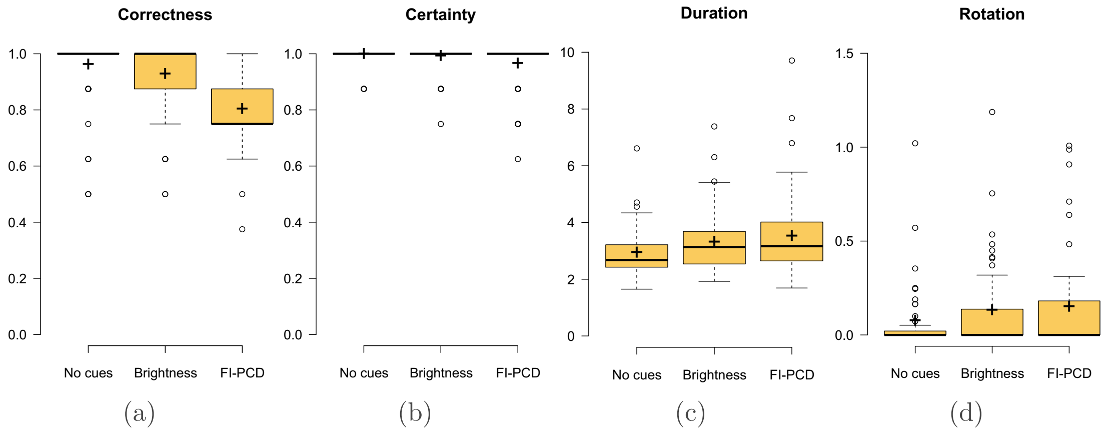


Figure 4.22: Box plots showing correctness (a), certainty (b), duration in seconds (c) and rotation duration in seconds (d) for the parameter judgment task from the pilot study.

For the parameter judgment task, the visualization without cues reached the best average correctness (96 %, Figure 4.22). Users also performed fastest, with an average duration of 2.9 s per image. This result was to be expected, as there are no additional color or brightness gradients added to the surface color. The brightness-based depth cues performed better than FI-PCD with regard to correctness (92 % compared to 80 %) and duration (3.2 s compared to 3.5 s). This is most likely due to our choice of color scale to encode the surface parameters in this study. Many participants remarked that the orange from the surface color scale was interfering with the red from the PCD scale, thus making it hard to distinguish them.

Interestingly, in order to interpret the colors of areas strongly affected by depth cues (i.e. those close to the edge of the vessel when using FI-PCD or those in the background when using brightness cues), users often resorted to "counting" color gradients. They would search for an area that was completely white and then count the boundaries they had to cross to reach the marked point. That way, they could tell which area represented a higher parameter value even without being able to distinguish the colors directly. Since this approach requires a path from a marked point to a white area that is visually uninterrupted, it was not possible in all cases.

Just as in the depth judgment task, the certainty for all shading styles is very similar. Rotation was used even more rarely in this task. Since the marked points were never obstructed by other geometry, there was little point in rotating the dataset to compare the surface coloring. We also analyzed the correctness with regard to whether the users made use of rotation during the tasks. The ability to rotate the view had very little effect on the results of the surface parameter task. The correctness of the depth judgment task increased when rotation was used on the FI-PCD images as well as those without depth cues. Since parallax movement is another important depth cue, this improvement is not surprising.

Final study: In our second study, the FI-PCD method reached better results. During the depth judgment task, users were able to pick the correct point in 94 % of the cases. With the brightness-based shading, they were able to choose correctly in 90 % of the cases, similar to the first study. Without any depth cues, the participants only reached 85 % accuracy. This accuracy is slightly higher compared to the first study, which may be a result of the users' increased familiarity with 3D vessel visualization.

The same trend is visible in the certainty plots. Overall, the duration and rotation plots from the second study show the same trends as those in the first study. Interestingly, users took longer for their decision and also rotated the view more when viewing the datasets with FI-PCD compared to brightness-based depth cues. This may be because the combination of PCD and surface color scale can no longer be perceived pre-attentively. As expected, users were able to judge the parameters best when no depth cues were present, reaching a mean correctness of 98 %. Brightness-based depth cues produced an almost identical result with a mean correctness of 96 %. FI-PCD shading had the strongest negative effect on the participant's ability to compare parameter values on the surface, although not as strong as in the first study. The mean correctness in this case was 90 %.

The average duration for each decision (from both tasks) was significantly higher in the second study (5.5 s) compared to the first (3.7 s). Similarly, the average rotation duration was also higher (0.3s compared to 0.2 s). This may indicate that in the second study, participants put more effort into the evaluation.

Discussion

With FI-PCD, we have introduced a novel rendering technique that combines pseudo Chromadepth with color-encoded surface attributes to visualize vascular anatomy in combination with scalar parameters. We have performed two studies to evaluate our technique. They have shown that FI-PCD can increase the perception of depth while maintaining recognizability of surface scales on the vessel surface. For the latter, a careful choice of color scale is required to avoid conflicts with the color gradients introduced by PCD. In our first study, we used an inappropriate color scale to encode surface parameters. This strongly reduced our method's ability to convey both depth and surface parameters at the same time. We were able to remedy this problem in the second study by choosing a different scale that relies only on the green color channel, which goes unused by PCD. This resulted in a higher increase of depth perception than classic, brightness-based depth cues.

We decided to use a discretized color scale instead of a smooth one. This reduces ambiguity between the surface color and PCD scale while at the same time highlighting areas with high or low values, which physicians are often interested in since their decisions are discrete as well. The highlighting was increased further by the introduction of outlines around the differently colored surface regions. This created a robust visualization that still allowed users to compare parameter values on the surface even when overlaid with another color or brightness gradient.

Both studies showed that overlaying the color channel with depth cues reduces the recognizability of the surface color scale. This effect was strongest when using FI-PCD. A likely explanation is that FI-PCD affects the color of both close and distant regions, whereas brightness-based depth cues only affect distant regions. Therefore, FI-PCD should be kept as an optional addition to any visualization that can be disabled in case an in-depth comparison between the scalar values of different surface points is required.

While we could show that FI-PCD can enhance the perception of depth, there are still issues that need to be improved on. Firstly, FI-PCD tends to distort the underlying color scale. This can be partly remedied by choosing a scale that does not interfere with the red and blue colors from PCD, such as our white-to-green scale. However, it would be interesting to see if this effect can be further reduced by using different values for the scaling factor and steepness in our FI-PCD formula. Reducing the scaling factor or increasing the steepness would make the PCD color scale less prominent in the visualization. Therefore, it may be possible to find a setting that results in a better trade-off between depth and surface color perception.

Explorative Hemodynamic Visualization and Comparison 4.3

The goal of many applications is not just to produce an effective visualization, but ultimatively to allow medical experts to gain new insights. This section details our work in the field of explorative visualization of hemodynamic data to allow medical researchers both to answer research questions and form new theories.

Explorative Blood Flow Visualization using Dynamic Line Filtering based on Surface Features 4.3.1

The content of the following section is based on:
Benjamin Behrendt, Philipp Berg, Oliver Beuing, Bernhard Preim, and Sylvia Saalfeld. "Explorative Blood Flow Visualization using Dynamic Line Filtering based on Surface Features". In: *Computer Graphics Forum* 37.3 (2018), pp. 183–194

After the original paper submission, we extended our application based on feedback from our collaborating physicians to allow for a more in-depth calculation of flow statistics and report generation.

Physicians are often interested in exploring blood flow patterns that manifest in specific hemodynamic features, such as changes in pressure or wall shear stress, on the vessel wall. Simply displaying them inside of the vessel anatomy using established smart visibility techniques would likely produce unsatisfactory results. To prevent occlusion between pathlines and vessel surface in the 3D visualization, the pathlines are often filtered by their parameters, such as velocity magnitude or vorticity. However, these filters require a priori knowledge of about the flow structures users expect to find. For an explorative approach, where the physician wants to figure out what kind of flow causes a specific phenomenon on the vessel surface, their usefulness is limited.

Thus, we present a set of techniques to interactively select and filter flow structures directly based on their effect on the vessel wall. By selecting regions on the vessel surface with hemodynamically interesting parameter values, such as local extrema in pressure or wall shear stress, underlying flow structures such as vortices are automatically highlighted. These highlights can then be further refined by filtering or modulating opacity based on parameters such as velocity, pressure or residence time. Our work was designed in cooperation with an experienced neuroradiologist to identify complex interactions between hemodynamic parameters in general, and combines the tasks of parameter visualization and pathline selection to create an intuitive and robust tool for explorative pathline filtering. The user's goal is to find local flow structures within the pathlines correlating to properties of the surface. A general workflow would therefore involve finding those surface features, selecting a subset of them for further investigation and extracting flow structures related to the selected features. The interaction should take advantage of the users' domain knowledge by allowing them to freely select regions they are interested in. Menial tasks such as manually drawing onto the surface to define this region should not be required, but kept as an option to increase flexibility. Based on this workflow, we identified the following key requirements for our application in cooperation with our clinical partners:

Req. 1 It should support finding medically interesting surface regions based on their morphology and hemodynamic parameters through the visualization.

Req. 2 The user should be able to easily select multiple surface regions and explore the associated local flow structures.

Req. 3 To support exploration, the user should have control about what kinds of flow structures are extracted.

Req. 4 The flow structures extracted from each selected feature should be visually distinguished.

Req. 5 The user should be able to further explore the extracted flow structures based on their own features.

To evaluate our work, we selected nine datasets, which yielded contradicting results using conventional analysis. The evaluation was carried out with two expert neuroradiologists and an expert of flow simulation. One of the neuroradiologists was involved in the design of our application, whereas the other is completely independent. In the evaluation section, we provide exploration results for these datasets as well as an informal qualitative evaluation.

During the evaluation, we show that our approach allowed for a systematic exploration and quantitative assessment of flow structures in aneurysms. Interesting structures such as vortices could be reliably detected and comprehensibly visualized, allowing the user to gain insights into the flow patterns both on a local and global scale.

Method

In this section, we will give an overview of the intended workflow we support (Figure 4.23). For each step, we will provide information about the user's interaction possibilities and explain our design decisions as well as the technical implementation. Starting from the vessel visualization, the structure is as follows:

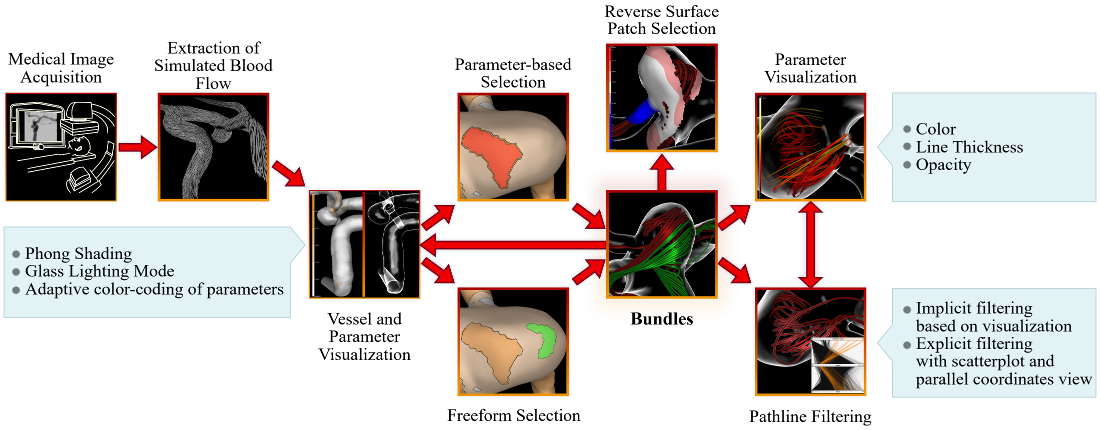

Medical Image Acquisition
Extraction of Simulated Blood Flow
Parameter-based Selection
Reverse Surface Patch Selection
Parameter Visualization
• Color
• Line Thickness
• Opacity
• Phong Shading
• Glass Lighting Mode
• Adaptive color-coding of parameters
Vessel and Parameter Visualization
Bundles
Freeform Selection
Pathline Filtering
• Implicit filtering based on visualization
• Explicit filtering with scatterplot and parallel coordinates view

Figure 4.23: Workflow for our application starting with the medical image acquisition.

1. Visualization of the vessel surface with mapped surface parameters
2. Selection of one or more interesting surface regions
3. Extraction of pathlines related to the selected regions
4. Further filtering of extracted pathlines

Vessel Visualization: Initially, the user is presented with an empty visualization of the vessel surface. We employ Phong shading with a single headlight to convey the vessel shape. To prevent the surface from occluding the inner flow that the user will eventually add, it is always rendered semi-transparently. The amount of transparency can be freely configured, but defaults to an empirically determined value of 33 %. While reducing the surface opacity does increase visibility of the inner flow, it also decreases the visibility of lighting effects and therefore reduces shape perception. We therefore decided to adopt a different strategy for applying lighting and transparency to the vessel surface that we call "glass lighting mode".

To emphasize the vessel boundaries even at higher transparency settings, we added a Fresnel effect to the lighting. Similar to the approach by Gasteiger et al. [151], the lighting intensity is then multiplied with the vessel opacity for each vertex, although we consider both the Fresnel and the Phong lighting for this. Regions with strong lighting therefore appear more opaque, highlighting the vessel shape and creating an effect similar to looking through a glass bottle. The glass lighting mode is enabled by default, but can be disabled by the user in favor of using traditional Phong shading with semi-transparent surfaces. A comparison between both modes can be seen in Figure 4.24. To prevent visual clutter from overlapping parts of the vessel, the user can set the backfaces of the vessel to be always fully opaque, despite the previously mentioned transparency setting. This is disabled by default to prevent the user from missing details in the flow that otherwise may be hidden. Figure 4.24 shows an overview of the effect of this setting both in the traditional as well as the glass lighting mode.

Since this visualization style produces multiple, overlaying transparent fragments for most pixels on screen, we employ Order Independent Transparency (OIT) [203] to ensure correct image composition. Instead of rendering fragments directly into a framebuffer and resolving overlays using a depth test, we write their color and depth values into a shader storage buffer using a linked list structure. Fragments with an opacity of 1 % or lower are discarded to reduce GPU load and conserve memory during composition, as such fragments barely contribute to the visualization. The final image is composed by a separate fragment shader that is applied to a screen-filling quad, effectively being executed exactly once for each pixel on screen. Using the linked list from the previous rendering stage, the shader gains access to all fragments for the pixel and is therefore able to sort them according to depth and to perform alpha blending in the correct order. This results in a correctly composed image generated entirely on the GPU without having to perform any pre-processing or ordering on the vertices prior to the rendering step.

Parameter Visualization: To add pathlines to the visualization, the user has to select at least one area on the vessel surface based on surface parameters. When the user is selecting these features on the vessel surface, naturally the surface is considered as the focus object. Therefore, it is now rendered fully opaque and allows mapping parameters using a color scale. As there are no pathlines rendered, the glass lighting mode is not available during this selection. Parameters of the vessel wall, such as pressure or wall shear stress, are displayed onto the surface using color scale. In order to highlight

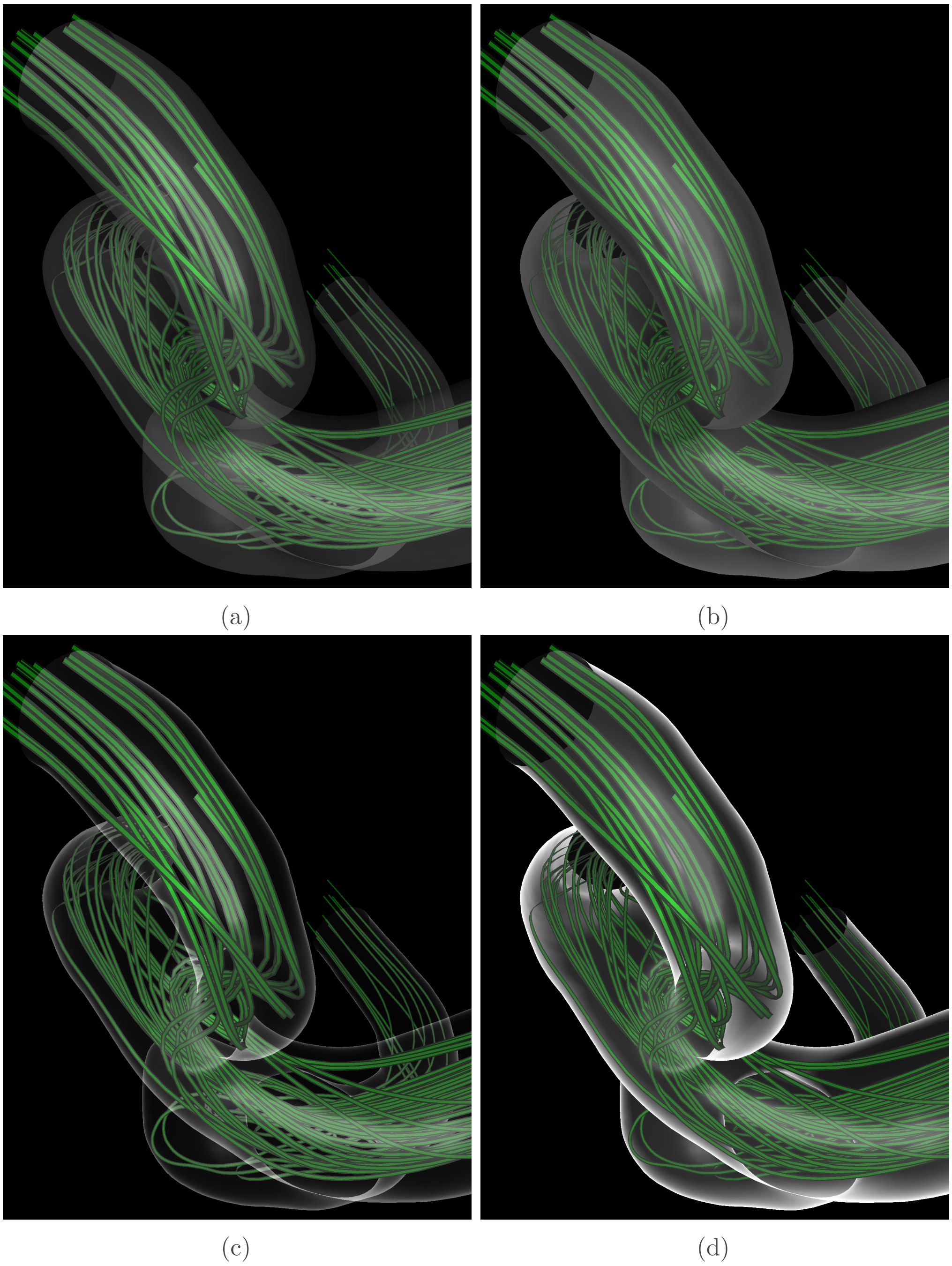

(a)
(b)
(c)
(d)

Figure 4.24: *Different rendering modes for the vessel surface in the pathline viewer; Disabled (a,b) and enabled (c,d) glass lighting, disabled (a,c) and enabled (b,d) opaque backfaces.*

interesting hotspots, which are characterized by local extrema of surface parameter values, we employ discretized color scales similar to the ones used in section 4.2.2. Once again, the boundaries are highlighted using black outlines to further emphasize the transition between shades. The user can freely choose from a set of predefined color scales and configure the amount of discrete shades. Figure 4.25 shows a comparison of different settings for the amount of shades. A higher amount of shades adds more details to the image, but can also lead to a cluttered visualization. Our clinical partners were interested in specific parameter ranges, i.e. areas with a normalized wall shear stress value below 20 %. To identify these regions, more than five different shades were rarely necessary.

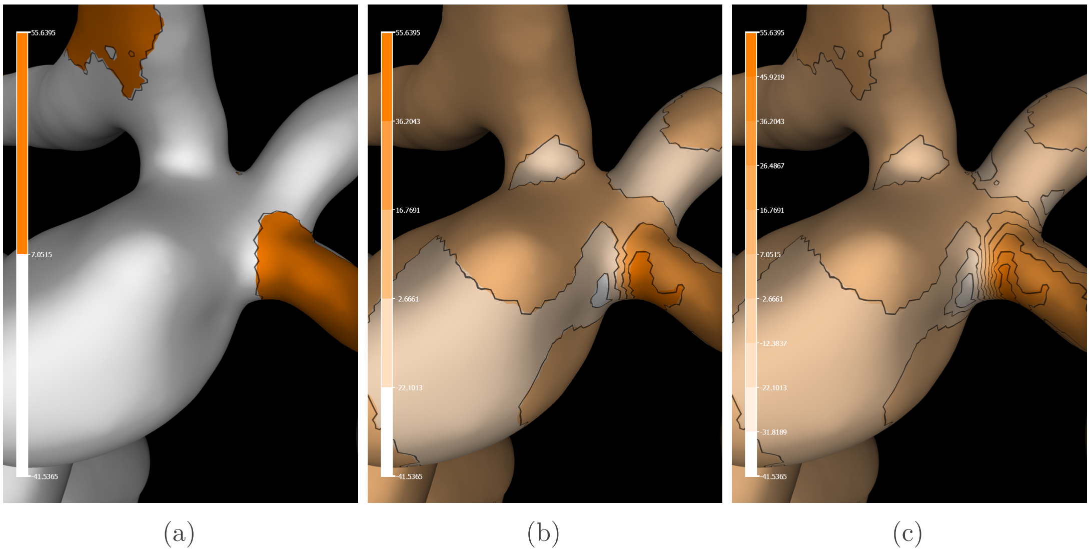


Figure 4.25: *Wall shear stress visualized on the vessel surface using a color scale with 2 (a), 5 (b) and 10 (c) discrete shades ranging from white to orange.*

Since the color scale and range settings need to be changed in real-time, they are applied entirely in the fragment shader. Parameters are normalized to a *[0-1]* range and uploaded to the GPU as vertex attributes. Settings, such as value ranges or the selected color scale, are stored as uniform variables for the shader, allowing them to change without the need to re-calculate the stored parameter values. To convey the meaning of the selected color scale, a color legend showing the parameter ranges for each shade is permanently visible on the left side of the image (Figure 4.25). Any change to the color scale, the amount of shades or parameter range is reflected on the legend in real time. The selected color scale's domain is initially determined based on the active parameter's value range and can later be adjusted. This is useful in case the parameter value distribution in the dataset does not fully cover the natural range of that parameter or if the user is only interested in a specific sub-range. The adjustment can be performed manually by simply entering new minimum and maximum values, or semi-automatically by basing the scale only on the currently visible surface area instead of the entire parameter range. Using the latter approach increases the detail dynamically when only a smaller part of the dataset is visible on screen. To achieve this, the fragment shader responsible for rendering the surface writes the parameter values it encounters into a buffer using atomic min/max operations. However, it may lead to overestimation of parameter differences, since smaller changes in the parameter value may lead to higher differences in the mapped color. To remedy this effect, the color legend will always show the entire parameter range, clearly indicating that the color scale currently only covers a part of the parameter range. At any point, the user can fixate the current automatically determined range to prevent it from changing as a result of adjusting the camera.

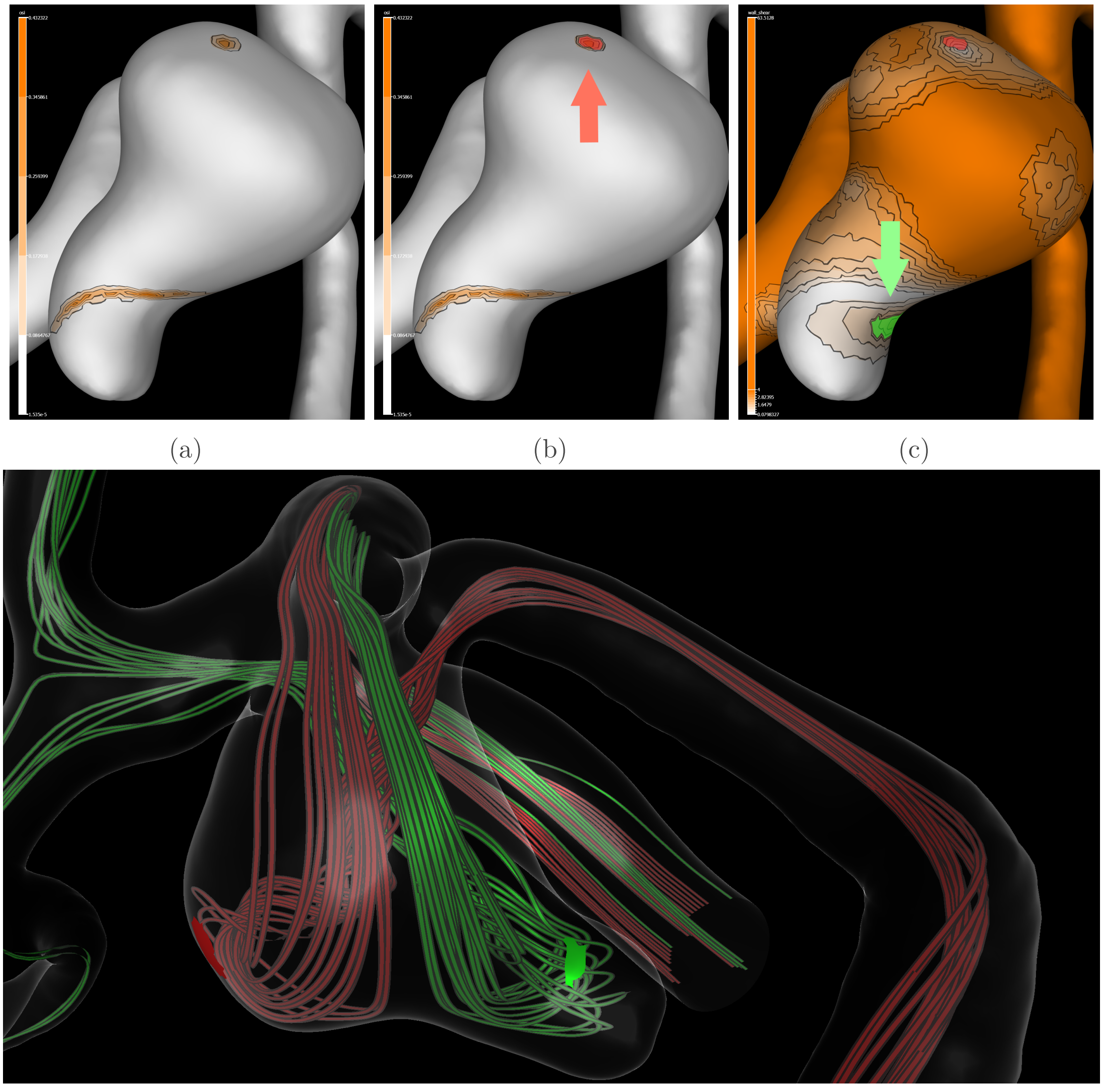
Surface Patch Selection: When the user clicks on the vessel surface, we determine the vertex closest to the cursor position in screen space. This is done by performing an off-screen rendering step that maps the IDs of all vertices onto the surface (without performing interpolation) and renders them into a framebuffer. Reading the value of that framebuffer at the cursor position returns the ID of the closest vertex. A simple way to select a feature on the surface would be to place a marker at the position of the closest vertex and then select all adjacent vertices in a specific distance. This type of selection is available in our software, but it is not the default setting. We decided against this approach as the primary method to select patches for several reasons. Using a distance threshold based on user input would add another step to the interaction, which we want to keep as simple as possible, according to requirement 2. It may also lead to confusion whether the distance threshold refers to the distance on the surface (resulting in a circular selection) or in 3D space (resulting in a spherical selection around the marker). Additionally, this type of interaction would limit the user to selecting circular or spherical sections of the surface.

Instead, we decided to allow selecting arbitrary regions on the surface. Unlike the approach by Neugebauer et al. [160], we base the region selection on hemodynamic instead of geometric features. We presume that a medically interesting region characterized by dissonant geometry would also be characterized by their hemodynamics. Therefore, we derive the selection shape directly from the surface parameter the user has enabled. We determine the color shade of the selected vertex and iteratively search for adjacent vertices with parameter values that would lead to the same color shade, effectively performing a flood fill on the surface (Figure 4.26b). Alternatively, the user can choose to also include “higher” or “lower” shades in the selection. This allows selecting arbitrarily shaped regions on the surface using a single click, whereas the parameter visualization itself works as a selection preview. We refer to these selections as “patches”. It is possible to change the active surface parameter during the selection. Previously created patches will remain unchanged, whereas the placement of new patches will be based on the currently active parameter (Figure 4.26c). We decided against using time-varying surface parameters, as they would likely make both the selection process and interpretation of the results more difficult.

By default, each singular selection creates a new patch. Additionally, it is possible to have multiple selections contributing to the same patch, even if the resulting surface is not coherent. If a selection based on surface parameters provides unsatisfactory results, the patch can manually be adjusted by drawing or erasing regions directly on the mesh. Visually, these patches are differentiated using predefined colors. Since the association of vertices to their respective patches is stored as flags in a 16 bit integral vertex attribute, it is possible for different patches to overlap. The total number of patches is therefore limited to 16. We assume this technical limitation to be unproblematic, as there would rarely be a situation where the user would need to create more than 16 different patches.

Pathline Visualization: After the user completes the selection of interesting surface regions, they have the option to select a distance from which to extract pathlines representing the associated blood flow. Only pathlines that come closer to the selected vertices than the distance threshold at least once during their course are selected. The chosen distance can be changed at any time, causing the pathline extraction to be repeated. The extraction is done by building a *KD tree* from the surface patch vertices and calculating the shortest distance for each pathline vertex from this patch. Only pathlines where at least one vertex is within the selected distance to the surface patch is included in the line bundle associated with that patch. We decided to use pre-integrated streamlines instead of dynamically seeding new lines close to the selected path for performance reasons due to the explorative nature of our application. Creating new pathlines with dynamic seeding would take significantly longer, whereas filtering existing pathlines takes at most few seconds and allows us to provide almost instant feedback to the user.

If the user has selected multiple patches on the surface, the distance threshold can be configured individually for each patch. The resulting pathline bundles are colored according to the patches they belong to (Figure 4.26d), making them visually distinct



(a) (b) (c)

(d)

Figure 4.26: *Workflow of selecting surface patches; surface visualization without any selection (a), selection of one patch (b, red arrow), second selection using a different parameter (c, green arrow); pathlines passing each of the selected patches highlighted with matching colors (d).*

(recall requirement 4). Pathlines belonging to different patches can be individually configured, such as by toggling their visibility or mapping parameters to their color, thickness or opacity. The patches themselves are visible on the vessel surface by default, but can also be individually hidden or rendered semi-transparently.

This is useful for extensive patches that may otherwise create occlusions. The pathlines are drawn as lines, then converted into view-aligned quads using the geometry shader. This allows the adjustment of the line on a per-vertex basis and also circumvents OpenGLs width limitation of line primitives. Alongside each vertex, we store the integration time point as a vertex attribute. This allows animating the flow by mapping the temporal distance of the time point stored for each vertex with the current animation time point to opacity. The temporal range for which vertices are visible can be adjusted by the user.

Pathline Filtering: To further refine the previously selected lines, pathline bundles can be filtered based on their parameters, such as pressure or velocity. This gives the user the ability to restrict the visualization of a previously extracted line bundle to a certain combination of features (requirements 3 and 5). One way of filtering the bundles is to map their hemodynamic parameters to line thickness or opacity, effectively reducing the visibility of lines with certain high or low parameter values. By mapping the residence time of a pathline in an aneurysm to opacity and thickness, for example, it is possible to highlight pathlines that stay inside the aneurysm for a larger amount of time (Figure 4.27). Instead of implicitly filtering pathlines using thickness or opacity, the user can explicitly select parameter ranges in a scatterplot or parallel coordinates view of the current pathline bundle. The scatterplot displays two parameters from the currently selected pathline bundle and allows the user to draw a selection rectangle. To allow filtering based on more than two parameters at the same time, we included a parallel coordinates diagram. The user can select which parameters are shown in this diagram and change their order. For each enabled parameter, they can interactively specify a range to filter pathline vertices.

Both the scatterplot and parallel coordinates diagram are synchronized. When the user performs a range selection on one parameter in a diagram, the selection is propagated to the other. In the parallel coordinates diagram, all lines belonging to a selected vertex are highlighted. To quantitatively compare two pathline bundles, they can both be plotted in a line chart (Figure 4.29). Here, one parameter of each bundle (such as speed) is plotted either over time or flow distance with respect to their average, minimum and maximum value as well as their 25 % quantile, median and 75 % quantile. Each of these metrics can be individually toggled by the user. Figure 4.29 shows the area between the average speed as well as the 25 % and 75 % quantile of two line bundles (red and yellow) and the complete set of pre-integrated pathlines (grey) plotted over the flow distance. The graphs use the same color as the pathlines in the 3D visualization.

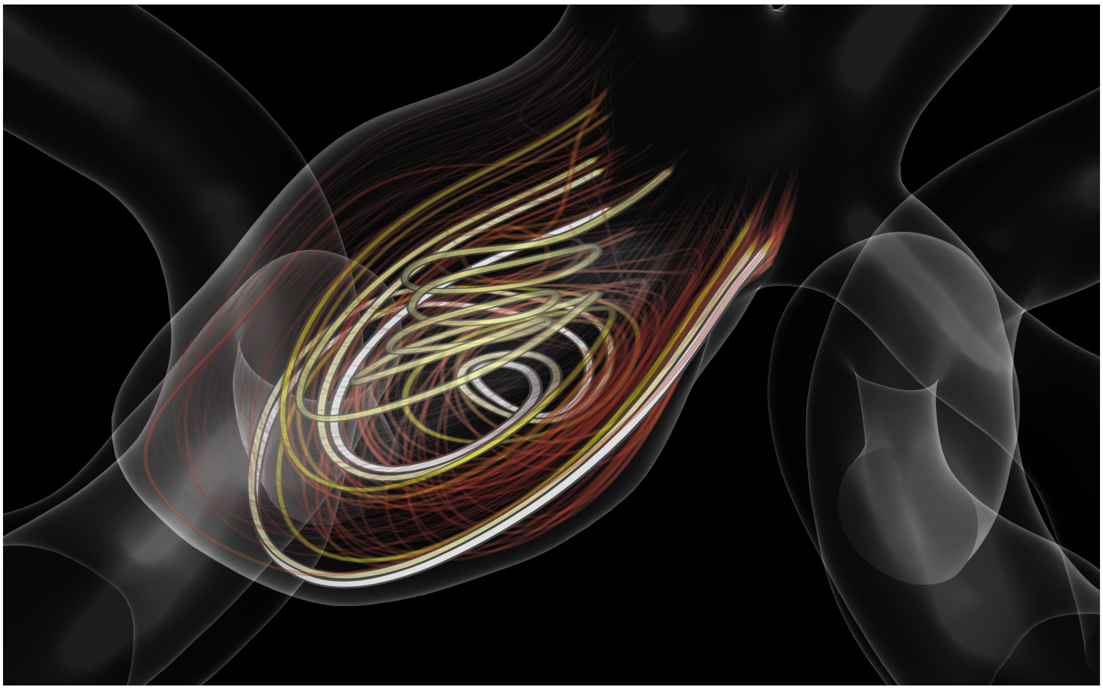


Figure 4.27: Visualization highlighting long-residing flow in an aneurysm by mapping residence time on line width, opacity and color (temperature scale) at the same time.

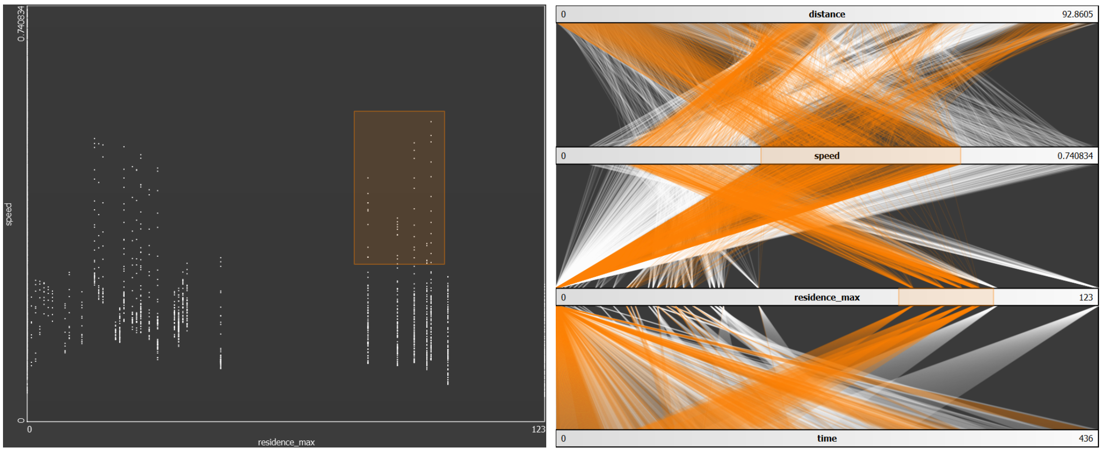


Figure 4.28: *Scatterplot and parallel coordinates view showing four parameters (integration time, residence time within the aneurysm, velocity in m/s and distance from the seed point) for each vertex of the same line bundle with synchronized selection. The scatterplot has residence time mapped to the x axis and velocity mapped to the y axis.*

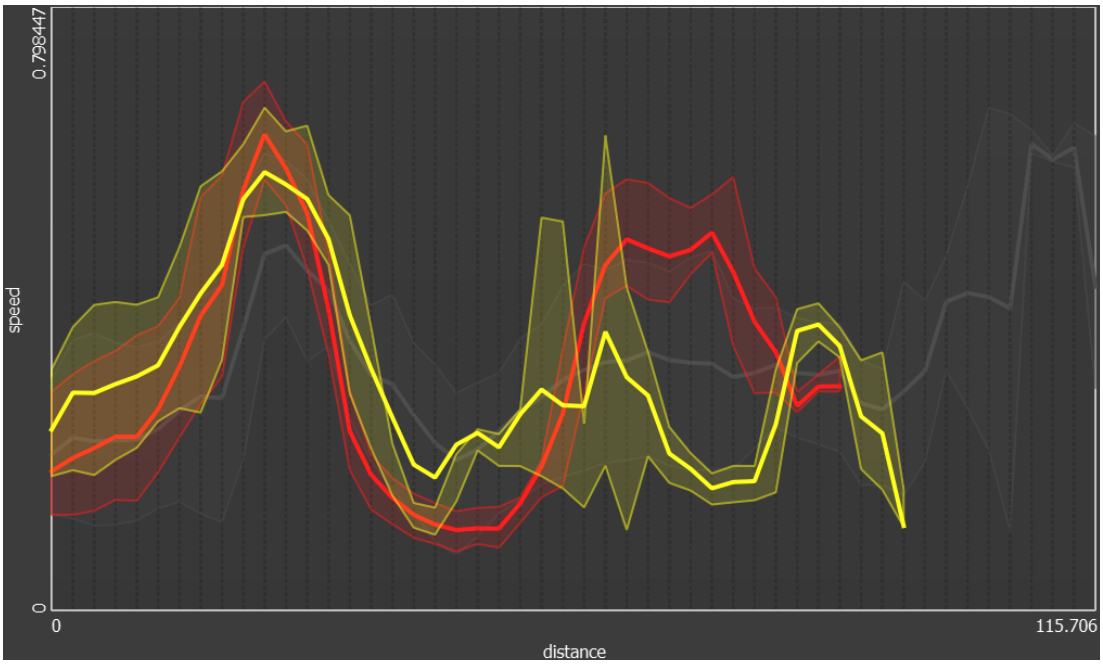
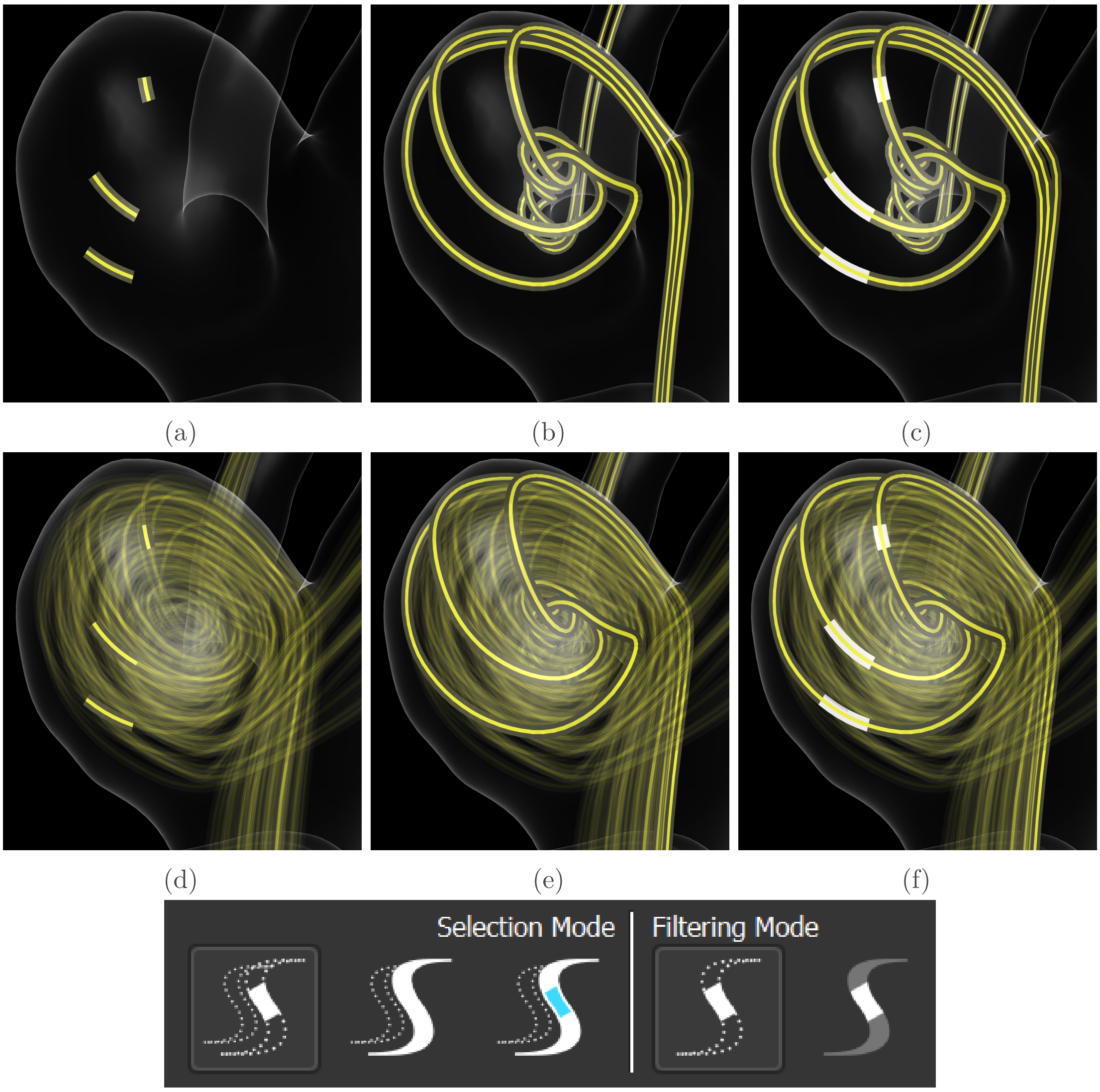


Figure 4.29: *Line chart comparing two pathline bundles with the global set of pathlines with respect to their speed and flow distance.*

Once the user has performed a selection in any of the diagrams, the 3D view will be updated accordingly. There are several ways in which highlighting or culling a certain parameter range can be performed (Figure 4.30). The first method is **vertex-based selection** (Figure 4.30a and 4.30d). In this mode, only vertices matching the selected parameter ranges will be kept. While this approach represents the user's selection exactly, it tends to produce very short line segments in some areas (Figure 4.30a). The **line-based selection** mode keeps an entire line if at least one vertex fits the parameter range (Figure 4.30b and 4.30e). This solves the problem of having very short line segments, but can lead to confusion as to which exact part of a pathline actually lies within the parameter range. The third mode is a **combination** of both previous modes. Like in the line-based mode, the entire line is kept. Additionally, vertices matching the parameter range are highlighted with white outlines (Figure 4.30c and 4.30f).


(a) (b) (c)

(d) (e) (f)

(g)

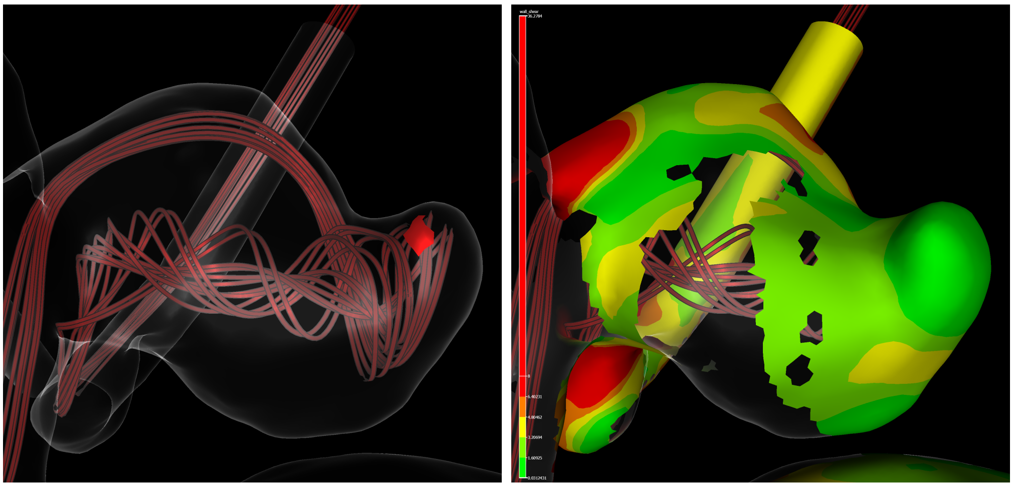
Figure 4.30: The six different combinations of filtering settings; vertex-based selection (a,d), line-based selection (b,e) and line-based selection with vertex highlights (c,f). Vertices / lines outside the selected parameter range are removed completely (a,b,c) or have their opacity reduced to 20 % (d,e,f). Icons used to represent these options in the user interface (g)

.

In addition to these three modes, the user can also choose how vertices or lines that do not match the parameter range are handled. They can either be removed from the visualization completely (Figure 4.30, a-c) or be shown with strongly reduced opacity (Figure 4.30, d-f). Completely removing them reduces visual clutter, but may also remove context information about the flow surrounding the selected areas. To support the mode selection and convey these rather technical options to the user in an understandable way, they are represented in the user interface by expressive icons (Figure 4.30g).

Reverse Surface Selection: The usual workflow involves selecting an interesting surface region and then extracting pathlines that pass this section closely. In some situations, however, physicians would not only be interested in which flow structures cause a specific surface feature, but also what other surface features the selected flow structure may pass. Therefore, it is possible to select additional surface regions based on their distance to an already extracted pathline bundle (Figure 4.31). This is implemented similarly to the way the pathline bundles are selected, except now the pathline vertices are written to the KD tree and compared against the surface vertices. By default, the same distance threshold is used, although this can be adjusted. The user can then map any parameter to the generated surface to look for other interesting surface features (Figure 4.31, right). Since we found the black outlines around the different shades of the color scale to be distracting in a view that also includes pathlines, we disabled them by default.

Flow Statistics Calculation and Report Generation: Originally, we designed our software to primarily support the visual exploration of flow structures. Based on feedback from our collaborating physicians, we added the option to calculate various quantitative measures concerning both the selected patches and associated pathline bundles, and export them in a standardized manner. Quantification of the exploration results allows physicians to perform statistical data analysis, thus further supporting them in gaining new insights. For the *patches*, we calculate the surface area and average values for all available hemodynamic parameters over the entire patch. For the *pathline bundles* associated to the patches, we calculate the average flow velocity over all pathlines that are part of the bundle. As the physicians are often specifically interested in the flow



(a) (b)

Figure 4.31: Original surface patch used to select a pathline bundle (a); additional surface regions extracted using the same bundle, with wall shear stress mapped to the color scale (b).

velocity within the aneurysm, we additionally compute the average flow velocity of the pathline sections between entering and leaving the aneurysm. The average velocity of pathline bundle sections that are close to the respective surface patch are also displayed. By default, this is calculated with ten distances between $0.1mm$ and $1.0mm$, although this can be freely configured. These values can be calculated individually for each surface patch and exported together with an automatically generated screenshot of the patch and pathlines as a PDF file. Additionally, the quantitative data from any pathline bundle used to create the scatterplot, parallel coordinates and line chart can be exported as a CSV file. This allows the user to employ an external application of their choice to perform further analysis.

At any point during the described workflow, the user can go back to previous tasks without losing any information. For example, if the pathline filtering pointed the users to possibly interesting surface regions they have not yet selected, they can return to the surface patch selection mode and add additional patches. The already existing patches, pathlines and filtering settings will remain unchanged.

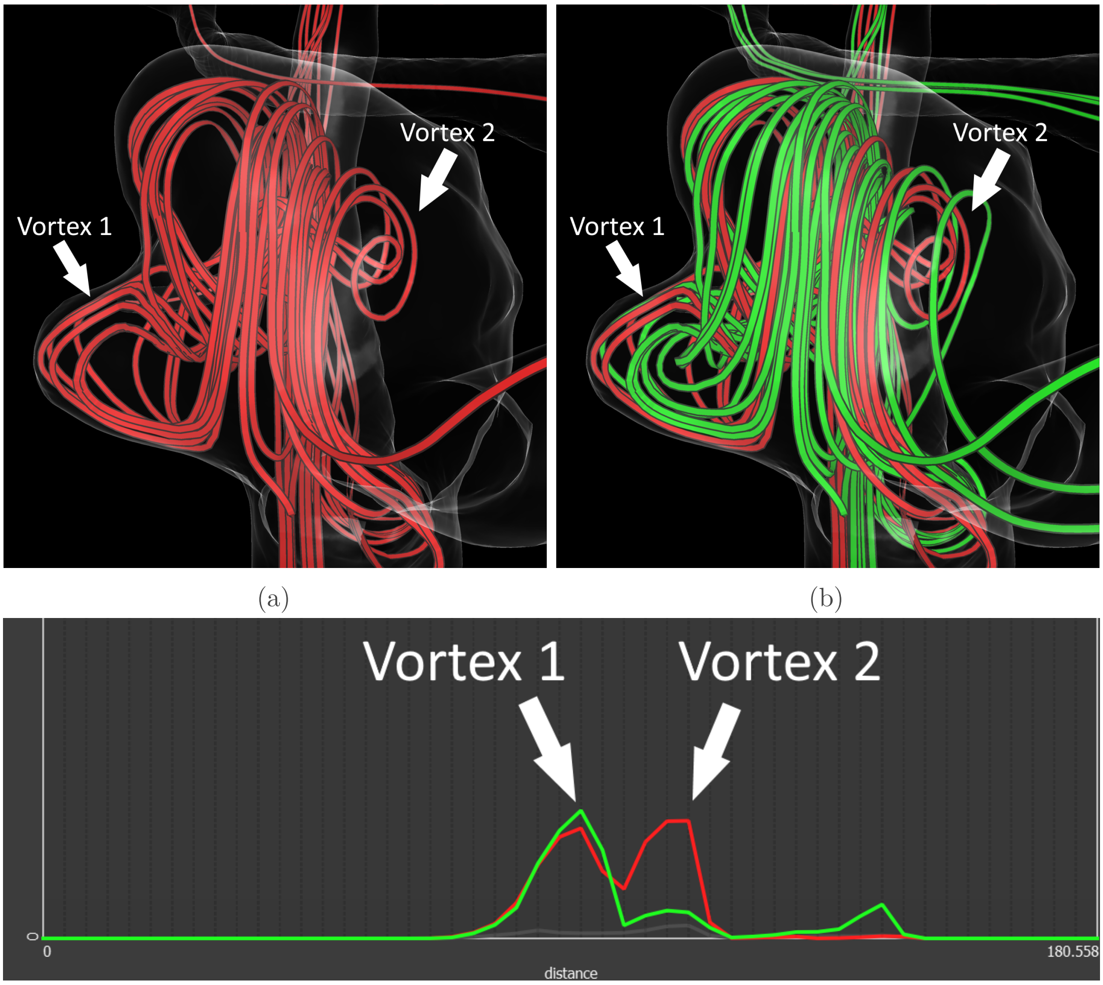
Results

To evaluate our methods, we asked two experienced neuroradiologists and an expert in flow simulation to apply it to nine aneurysms and recorded their findings. Two of these datasets were from a longitudinal study, acquired three years apart from each other. We also asked for general feedback in an informal interview afterwards. The first neuroradiologist and the flow expert were able to use the software themselves after a short introduction and demonstration on one dataset. The second neuroradiologist participated over the internet, using remote control. While she gave precise instructions on which patches to select, the actual interaction with the application was performed by us.

All three experts described our method as an advancement in the field of explorative flow visualization. They were able to quickly find interesting surface regions that almost always yielded interesting flow patterns such as vortices when selected. The color-coding proved especially useful for assessing which adjacent vessels a particular flow pattern drains into. According to the experts, a precise selection of specific flow patterns based on their relation to surface features has previously not been possible. They highly appreciated the visualization of splitting flow. Overall, our combination of interaction, visualization and filtering techniques allows for systematic exploration and qualitative assessment of flow structures.

The first neuroradiologist was primarily interested in patches with either high or low normalized wall shear stress or high OSI. His main goal was to correlate vortex structures in the flow with specific hemodynamic parameters on the vessel wall. To facilitate comparability between datasets, the expert used similar or identical parameter ranges for the placement of surface patches and the extraction of pathlines. In a few cases, the expert made use of the function to manually draw patches, for example when a patch would otherwise “bleed” into the parent vessel. The expert also used the line chart to determine if a pathline bundle contains more than one actual vortex structure. To correlate features from the line chart with the 3D visualization, the expert used vertex-based filtering (Figure 4.30a). He also used the line chart to determine if a pathline bundle contains more than one actual vortex structure by plotting the residence time over flow distance. Figure 4.32 shows an example of such a situation from the evaluation.

The red line bundle contains two different vortices, which is not instantly obvious due to the complex nature of the flow in the 3D visualization, but clearly visible in the line chart. With this information, the expert was able to add another patch that captures a flow bundle only passing through one of these vortex structures (green line bundle). Placing a single patch took the expert between 24 and 110 seconds, depending on the complexity of the vessel geometry and if he had to manually draw a patch. Since settings, such as mapped surface parameter, number of color scale shades or custom parameter ranges were reset to a default value when switching between datasets, he wished for a way to change the default values or create custom presets to accelerate the process of placing patches. Figure 4.33 shows two pathline bundles that the first neuroradiologist selected in an aneurysm. The red bundle was selected based on a local pressure minimum, the green bundle based on a wall shear stress minimum. Both the red and green vortices only appear after the blood flow hits the vessel wall. The flow decelerates when entering the aneurysm and accelerates when leaving it (Figure 4.33b). Although the aneurysm is located at a bifurcation, the flow from both vortices drains exclusively into only one of the adjacent vessels. The second and third dataset were acquired from the same patient at different points in time. The neuroradiologist was therefore interested in visualizing



(a)

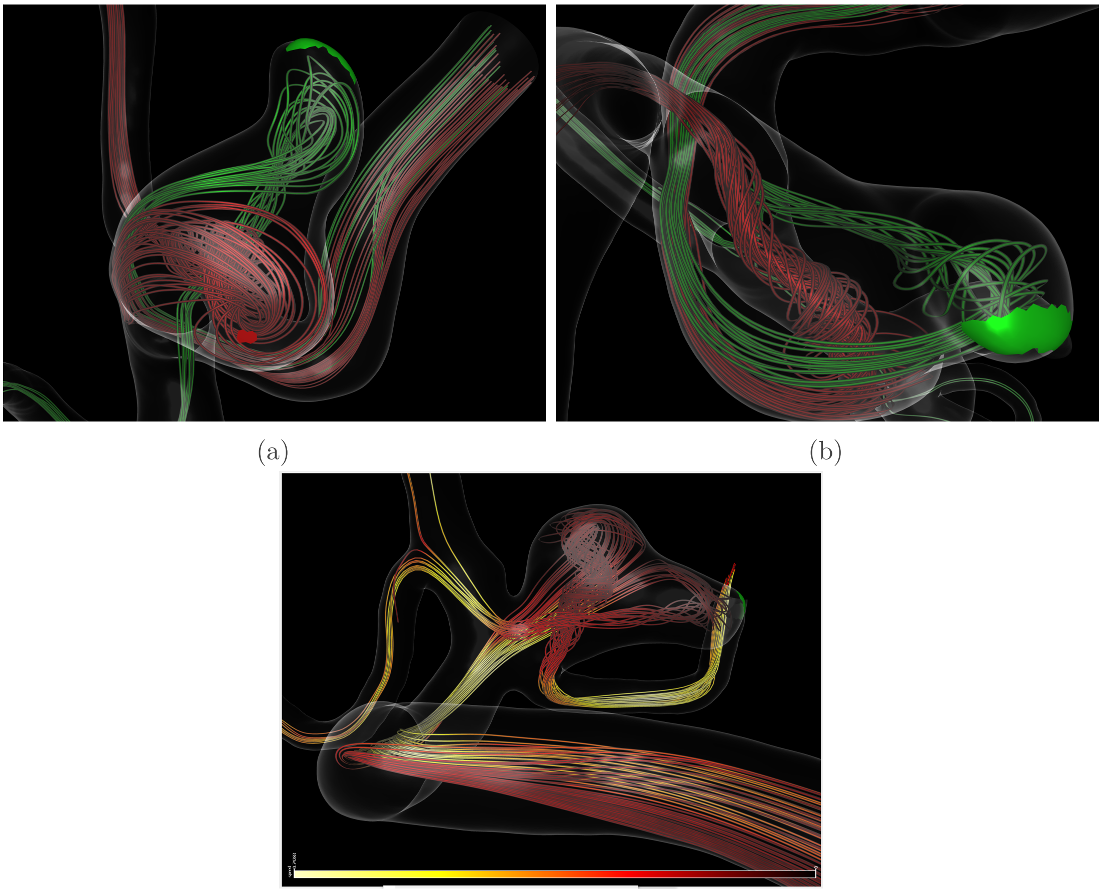
(b)

(c)

Figure 4.32: *Using the line chart plotting average residence time over flow distance to differentiate multiple vortices from a complex flow structure; the line chart (c) clearly shows multiple structures being present in a flow bundle (a) and helps creating a selection that only contains one of the structures ((b), green lines).*

the development of the aneurysm and flow. Since our application did not directly support the comparison of datasets, he improvised by running two instances at the same time and manually adjusting the camera to show a similar angle (Figure 4.34). Selecting a patch at a similar location in both datasets allowed for a qualitative comparison of the changes in flow patterns.

The second, independent neuroradiologist was primarily interested in visualizing splitting flow in aneurysms for the purpose of optimal flow diverter placement. She stated that highlighting the splitting flow structures can provide decision support for the placement of flow diverters. Usually, the physician wants to place a flow diverter to reduce pressure from the aneurysm without covering neighboring vessels completely since this would stop blood supply via these vessels. According to the expert, experienced neuroradiologists are often able to infer this information from the wall geometry alone. However, visualizing the splitting flow could be a valuable help to less experienced neuroradiologists. Since the expert had limited interest in correlating flow structures with surface parameters, she mostly placed patches based on geometric features, such as blebs or the aneurysm dome. When filtering pathline bundles based on their hemodynamic parameters, this expert preferred line-based filtering with reduced opacity for filtered lines (Figure 4.30f). We did not record the time she took for patch selection since the interaction was not performed directly by the expert.



(a)

(b)

(c)

Figure 4.33: *Flow selection performed by one of the neuroradiologists from three different perspectives; the right image has the flow velocity mapped onto the pathlines using a temperature color scale.*

Our method proved to be stable in respect of the parameter chosen for the surface patch selection. Interesting flow structures often manifest in multiple surface parameter changes, either in different locations or different parameters. For example, the red line bundle in Figure 4.33 could have also been selected using the OSI parameter. In cases where a selection either did not yield an interesting flow structure or resulted in multiple structures at once, the resulting pathlines usually contained clues about more promising surface regions that could be selected instead. The ability to manually draw patches onto the surface without regard to the underlying parameters proved to be useful when the experts wanted to select a region based on vessel morphology that did not fully correlate to any surface parameter. A point of criticism was that manually adjusting the surface color scales was often necessary. The color scale domain is initialized using the global parameter minimum and maximum, yet the physician is generally looking for local minima and maxima. These values may not always be visible initially due to the discrete nature of the color scale, therefore requiring manual adjustment. They requested various (possibly customizable) presets for these adjustments to be added to the application in order to save time when selecting surface patches with recurring parameter configurations.

All experts expressed their interest in being able to further quantify various aspects of their exploration results. An example for that would be the ability to measure the size and extent of detected structures. More complex measures, such as the amount of flow that passes through a certain structure or directly underneath a surface patch, would be desirable as well. Calculating additional quantitative measures for each line bundle would allow the physician to gain a deeper understanding of the flow patterns. Another requested feature was the ability to place a plane into the parent vessel of an aneurysm and map the attributes and spatial positions of pathlines passing through it. This would generate a flow profile depicting which regions of the vessel cross-section feed or drain into different flow structures. A potential use for this kind of information would be the optimization of stent placement.

Discussion

The feedback from all experts shows that our method can support the visual exploration of blood flow and its relation to surface features. According to their feedback, we were

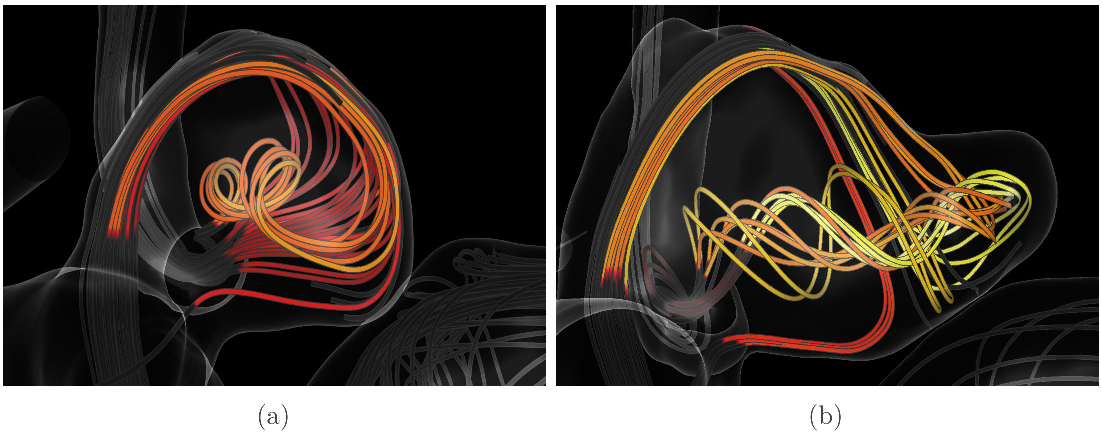

(a) (b)

Figure 4.34: Comparison of the same aneurysm acquired in 2011 (a) and 2014 (b); the residence time is mapped onto a temperature color scale.

able to fulfill the previously presented requirements. The use of a discrete color scale allows for a fast localization of extreme hemodynamic parameters. Unlike previous approaches such as [110, 157, 204], the local flow structures associated to an interesting surface structure can be visualized with only a few mouse clicks. Further exploration of the resulting pathline bundles is possible by either mapping their parameters onto a color scale for filtering them in real-time using a parameter scatterplot, parallel coordinates view or line chart. Both the extraction of pathline bundles as well as the additional filtering can be performed in real-time. Color-coding the selected patches and associated pathlines allows for an easy visual assessment of the entire course of a bundle, similar to van Pelt et al. [110].

We have presented a set of intuitive techniques to allow for an interactive exploration of local blood flow based on surface features. Both clinical and the flow simulation expert appreciated the local selection techniques to analyze blood flow characteristics in combination with surface parameters. In fact, both stated that they were missing this opportunity in their respective known tools. They also were interested in the visualization of the flow splitting and appreciated the presentation of the pathlines for the entire vessels. Furthermore, the sophisticated real-time filtering techniques including parameter-based filtering and usage of parallel coordinate views as well as the scatterplot could fulfill all of their requests regarding the selection of specific blood flow characteristics. At the moment, the selection of interesting surface regions is performed manually. Automatic suggestions for regions, similar to Neugebauer et al. [160], could potentially increase the reproducibility of our approach and prevent details from being missed. To facilitate such a selection, a measure for *interesting-ness* could be generated by calculating the local deviation of multiple geometric features and hemodynamic parameters. Although tailored to cerebral aneurysms, our methods can be easily adapted to other applications both in- and outside of the medical field. All that is required for our software to work is a surface model and a set of arbitrarily generated pathlines. Having quantitative parameters mapped to them extends the filtering possibilities, but is not strictly required for our application to be used. Problems could arise when working with complex intertwining surface models, since occlusions might hinder the user's ability to select certain parts of the surface. Possible solutions in this scenario would be to use semi-transparent surfaces in combination with an automated selection algorithm, as presented by Mühler et al. [205].

At the moment, our software is focused on the exploration of a single dataset. There are, however, many scenarios in which physicians would like to compare different datasets. For example, a physician may want to see how a treatment they have performed affected the blood flow in comparison to a dataset acquired before the procedure. To support these comparisons, further quantitative values in addition to the existing ones should be extracted, for example about the flow directly underneath a patch or the patch itself. Instead of simply showing multiple datasets side by side in isolated views, an integrated visualization would be desirable. This would require translating either the surface patches or the seed points for a selected pathline bundle to highlight how the flow has changed between datasets.

Evolutionary Pathlines for Blood Flow Exploration in Cerebral Aneurysms 4.3.2

The content of the following section is based on:
Benjamin Behrendt, Wito Engelke, Philipp Berg, Oliver Beuing, Bernhard Preim, Ingrid Hotz, and Sylvia Saalfeld. "Evolutionary Pathlines for Blood Flow Exploration in Cerebral Aneurysms". In: *Eurographics Workshop on Visual Computing for Biology and Medicine.* The Eurographics Association, 2019

In addition to the methodes described in this publication, we added a pathline clustering based on DBSCAN to automatically extract different flow structures within the set of generated pathlines. Another addition is the option to display all pathlines without occlusion in a 2D pathline overview plot.

In the previous section, we limited our prototype application to work with pre-integrated pathlines. This was to ensure that all interactions with the tool can be performed in real-time. However, this approach may be unsatisfactory in some situations. For example, if the users feel that a certain interesting area lacks details due to a low number of pathlines, they may want to dynamically add more detail by seeding additional pathlines in that specific region. Finding a suitable set of pathlines representing the most important flow patterns while avoiding visual clutter is challenging. Pre-computing sets of lines using dense seeding on a regular 2D grid in the inflow area of the respective vessel subsection followed by filtering is a typical solution. However, the seed points for interesting pathlines are often sparsely distributed in the seeding domain and can therefore easily be missed, even when seeding a high amount of pathlines. Thus, high computational costs are required for pathlines that will never be shown, while still having no guarantee to find the features of interest.

Recently, Engelke et al. [143] introduced evolutionary streamlines for the analysis of steady flows. They formulated the task of finding representative lines in a steady flow field as an optimization problem, achieving high-quality results while requiring significantly fewer line computations than traditional filtering approaches. They also managed to overcome under-sampling, since the approach is not tied to a predefined sampling resolution. We extend this method to evolutionary pathlines for unsteady flow analysis in complex vascular domains including pathologies by

- Extending the algorithm to support time-dependent data and produce pathlines instead of streamlines.
- Adapting the fitness function to specific requirements for blood flow analysis in cerebral aneurysms, including the ability to incorporate properties of the vessel surface into the fitness calculation, inspired by our previous work (recall Section 4.3.1).
- Embedding this technique into a framework designed to support clinical research and treatment decisions for cerebral aneurysms.

The generation of feature-sensitive sets of pathlines is integrated in an exploration framework for individual patients as well as the comparison amongst patients. A set of predefined fitness functions supports an automatic extraction of flow patterns of patient groups with high anatomical variations and aneurysm shapes, including aneurysm blebs or strongly lobulated shapes. 3D visualization of the pathlines gives the anatomical context required for improved treatment planning. Mapping properties of the evolutionary algorithm, such as fitness, onto the pathlines supports the comprehensibility of our approach. An easy specification of general optimization criteria for entire patient groups is possible, thus allowing simulated cerebral blood flow data to be batch-processed. Our method has been technically evaluated with respect to convergence behavior, stability of the results and computational efficiency. A case study shows the usefulness for the analysis of hemodynamic flow in cerebral aneurysms for treatment planning. The feedback of the domain experts suggests that the approach is superior to previous seeding and analysis strategies. Thus, the main contributions of our work can be summarized as:

- A new approach for analysis of medical blood flow data based on evolutionary pathlines.
- An efficient pathline seeding strategy to achieve high line coverage, detecting even sparse and highly localized features without relying on a complete field analysis.
- A technical evaluation of the algorithmic properties with respect to convergence, reproducibility, performance and line coverage, as well as an evaluation of the usefulness of the approach with domain experts.

Background

For pathline visualization, even a dense seeding at the parent vessel's inlet or the ostium cannot ensure to entirely represent the intraaneurysmal flow, whereas further increasing the amount of pathlines yields visual clutter. A characterization of blood flow patterns [106] as well as an aggregation into meaningful clusters [163] prevent visual clutter, but depend on pre-integrated lines and cannot account for strongly lobulated aneurysm shapes with blebs. Seeding directly in the region of interest and then

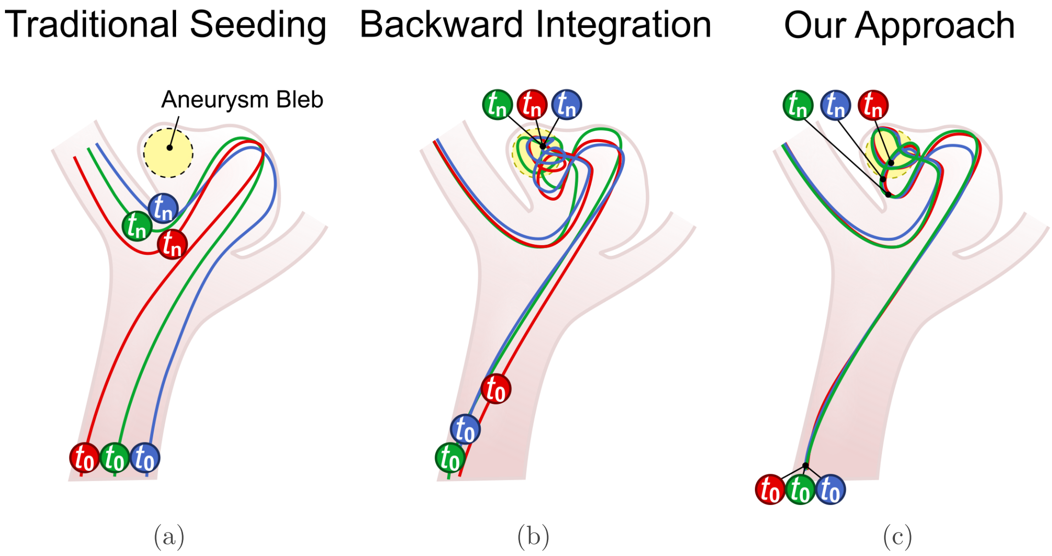


Figure 4.35: Illustration of line coverage using different seeding methods. Traditional Seeding (a) is under-sampling the aneurysm bleb. Backward Integration (b) leads to incorrect vortices due to large temporal differences. In contrast, our method (c) generates pathlines with identical start times and reveals complex vortices when searching for pathlines passing a specific area on the vessel surface.

performing both forward- and backward integration generates pathlines in the desired region. However, these pathlines arrive at the inflow plane at different points in time (Figure 4.35), which does not accurately represent the blood flow in the vessel and does not meet the requirements of the clinicians. In fact, it could prevent the extraction of complex structures like an embedded vortex. A similar problem arises when using two-pass integration, where potentially interesting seed points for forward-integration are determined by using backward integration, as seeding at a different time point (t_0) will produce a different pathline.

Evolutionary algorithms as meta heuristic for a guided search are based on the principle of natural selection by Charles Darwin [207]. They can be used for optimization problems, where an analytic solution does not exist and an exhausting exploration is not feasible. Application areas for evolutionary algorithms include parameter optimization (e.g. curvature of pipes) [208], packing and scheduling problems, and biological modeling. A recent book by Kruse et al. [209] discusses evolutionary algorithms in the context of computational intelligence. The core principle of evolutionary algorithms is to improve initial random candidates by using evolution-motivated mechanisms such as *selection*, *crossover* and *mutation*. Thus, a problem of the form:

$$\mathcal{H} = \{x \in \Omega | \forall x' \in \Omega : f(x) \succeq f(x')\} \tag{4.4}$$

can be optimized, where the goal is to find an element x in the search space Ω which optimizes a function $f : \Omega \to \mathbb{R}$ in Ω. This function is referred to as the *fitness function* and defines a quality measure for the candidate solutions and a comparison operator $\succeq$ in Ω. Furthermore, the specific implementation of the fitness function, selection, and especially the genetic operators (i.e. crossover and mutation) are highly problem-specific. Additional parameters for steering the algorithm are *population size*, *crossover probability*, *mutation probability*, and *mutation strength*.

Notation: We refer to a single solution candidate as *individual* I. The fitness of a specific individual I_i is referred to as $f(I_i)$. Furthermore, a *population* P consists of n_p individuals. Parameters for steering the evolutionary algorithms are percentages for *Elite Selection* p_e, *Mutation* p_m, *Crossover* p_c, and *Insertion* p_i. A single iteration of the evolutionary algorithm is referred to as a *generation*.

Initialization: In this step, the population P of size n_p is created, with each individual $I \in P$ representing one solution. The initial population contains only individuals with random genomes.
Evaluation: During the evaluation, each individual's fitness is calculated. With this, we obtain a value $f(I_i)$ with $i \in [0 \ldots n_p - 1]$.
Fitness-Based Sorting: After evaluation, the list of individuals is sorted according to their fitness. Subsequent steps, such as elite selection, mutation, crossover, and the calculation of convergence measures, rely on an ordered list of individuals.
Elite Selection: This step ensures that good solution candidates are kept between generations. Transferring n_e unmodified individuals to the next generation P' is referred to as elite selection. These individuals are selected from the ordered list according to I_i with $i \in [0 \ldots n_e - 1]$ and $n_e = p_e \cdot n_p$.

Crossover: The first genetic operator combines the genome of n individuals to create n offsprings. In its simplest form, with $n = 2$, each offspring inherits half of each parents' genetic information.
Mutation: The second genetic operator slightly alters each individual's genome, whereas the probability, type and strength of mutation are configurable. This step can be understood as exploration of the local surrounding of a solution candidate in the search

space Ω. A mutation is only imposed in subsequent generations if it is beneficial for the individual's fitness. Individuals are selected from an ordered list according to I_i with $i \in [n_c \dots n_e - 1]$ and $n_c = p_c \cdot n_p$, $n_e = p_e \cdot n_p$
Insertion: This step replaces a part of the population. For this, the $p_i \cdot n_p$ individuals in P with the lowest fitness are replaced by new individuals when inserted in P'. Similar to the initialization, the new individual's genome is randomly generated. This step adds to the genetic diversity and thus ensures further random exploration of the whole search space.

System Design

In this section we describe our method, consisting of *Pathline Generation* and *Pathline Visualization*. We start by providing an overview of the associated workflow for different types of users. The visualization and analysis system is composed of three layers: the *Technical Layer* that represents the computational backbone of the system, the *Advanced Layer* targeted at researchers and the *Clinical Layer* targeted at physicians. Additionally, there is a *Preprocessing Layer*, which details the data processing before being imported in our system (Figure 4.36).

Preprocessing Layer: Our software was developed for cerebral aneurysm surface models who are typically extracted from contrast-enhanced 3D rotational angiography. For the cases presented in this work, a threshold-based segmentation approach was applied to the 3D DSA images, followed by marching cubes to obtain a triangulated surface mesh using MeVisLab 2.7 (MeVis Medical Solutions AG, Bremen, Germany) [210], similar to the approach described by [76]. For the separation of an aneurysm from the parent vessel, we used a semi-automatic neck curve detection [211]. Based on the segmented aneurysm surfaces, hemodynamic simulations were carried out using STAR-CCM+ 13.04.

Technical Layer: This layer contains the main computational components, mainly the evolutionary algorithm with its integrated pathline integrator and the DBSCAN clustering algorithm. It is only targeted towards the developers and maintainers of the system and receives its configuration and data from the other layers.

Advanced Layer: This layer is targeted towards researchers in the medical domain and provides a variety of exploration options. It is designed to support the creation of non-dataset-specific presets for use in the clinical layer. Thus, it allows the configuration of the fitness function as well as the genetic parameters for the evolutionary algorithm. The *Fitness Function Editor* enables the user to create a fitness function using a weighted combination of configurable properties. Similarly, the *Genetic Parameter Editor* allows to configure the parameters for the evolutionary algorithm, such as the mutation rate or number of individuals. The fitness function together with genetic parameters can be saved as a named, dataset-independent preset for specific clinical use cases, e.g. extracting flow structures close to regions with high WSS or OSI. To support the comprehensibility of the evolutionary algorithm, it is possible to explicitly filter pathlines in the 3D view based on their overall fitness value. Comprehending why the evolutionary algorithm converges to specific pathline bundles can be a valuable aid in designing fitness functions for clinical use. For a more in-depth analysis, users may also view the previous generations of pathlines, thus allowing them to trace how the evolutionary algorithm generated the final result. Additionally, the advanced layer also allows to freely configure the pathline visualization with respect to mapping attributes to color scales or opacity. These settings can also be stored as dataset-independent presets.

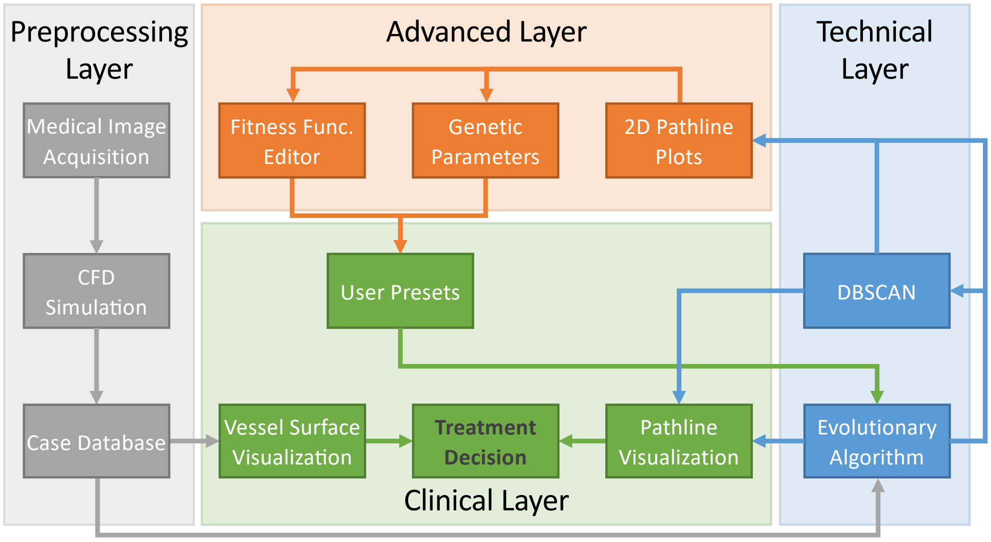


Figure 4.36: Overview of the workflows and layers within our framework.

Clinical Layer: This layer is reduced to a minimum of interaction, especially tailored to the clinical users and designed to support treatment decisions. It allows for a selection of datasets from the preprocessing layer and presets from the advanced layer. A manual configuration of the fitness function, genetic parameters or visualization settings is not available. Thus, the configuration for a clinical workflow needs to be prepared first within the Advanced Layer. The user can select multiple datasets to perform batch processing with a single preset configuration. Additionally, basic interactions like rotation and zooming in the 3D view are supported.

Pathline Generation

We use an evolutionary algorithm to generate a set of representative pathlines for the hemodynamics in an aneurysm based on a simulated unsteady vector field. The input to the evolutionary algorithm is the aneurysm geometry and the simulated blood flow. A plane is automatically fitted in the vessel inlet, which serves as seeding plane for the pathline generation. The evolutionary algorithm details, such as the highly domain-specific encoding of solution candidates and the genetic operators, have been specifically designed for our application.

Individuals: The genome of the individuals encodes a single seed point, which uniquely defines a pathline with respect to the integration settings. Thus, the search space Ω is defined by the integral lines' seeding domain. The individuals are stored as two-dimensional points $I_i = (x_i, y_i)$ in the local coordinate system of the seeding plane, with $x_i, y_i \in [0, 1]$. For the integration process, they are transformed into world coordinates. New individuals are always initialized with random values in their genome.

Mutation: In this step, we add a weighted displacement vector $\vec{d} \in [-1.0, 1.0]^d$ to the individual's genome; here, d is the dimension of the search space. Furthermore, we facilitate a generation-dependent weight which decreases with increasing generation count by a constant value, ensuring that with increasing average fitness of the population, solution candidates are altered less.

Evaluation: To evaluate the fitness of an individual, the entire pathline has to be integrated from its seed point. We use Shepard interpolation with four samples to acquire the velocity vector from the flow field. The integration is performed using fourth-order Runge-Kutta (*RK4*) and a fixed step size of 0.25, although, at the end of the integration, only every fourth point is kept to speed up the visualization and evaluation of the lines. Our collaborating physicians prefer a fixed seeding time to prevent visual confusion from pathlines seeded at different time points being visible simultaneously (recall Figure 4.35). Furthermore, this better mimics the clinical time-dependent data using contrast agent, which arrives in the vessel as a bolus. Therefore, we always seed at the first time step ($t = 0$). The fitness function $f(I_i)$ is assembled from a set of local properties $f_1..f_n$. The properties can be grouped into two categories:

- **Line properties** produce fitness information based on pathline geometry, such as length or curvature.
- **Surface properties** produce fitness information based on attributes of the vessel surface close to the pathlines, such as pressure, WSS and OSI.

Both length and curvature of a pathline can indicate the presence of vortex structures, as a pathline is often longer and has higher curvature when passing through a vortex. In cases where a separation of the aneurysm from the parent vessel is available, the residence time of a pathline within the aneurysm can also be used. Thus, only lines in specific regions are favored and thereby implicitly filtered in a spatial manner. To obtain surface properties in the context of a pathline, the closest vertex on the surface is efficiently determined using a KD tree for each line vertex. Then, the selected attribute from the surface vertex is normalized and assigned to the pathline vertex as a fitness indicator. Optionally, a distance penalty can be used to reduce the fitness indicator value after a certain distance from the surface to focus on near-wall features.

Aneurysm geometry: If the aneurysm in the dataset is segmented, all indicator values for pathline vertices outside a specific region can be disregarded, effectively preventing surface attributes outside of the aneurysm from contributing to the fitness of a line.

$$f_p(I_i) = \bigodot_{j=1}^{|I_i|} (v_{ij}) \tag{4.5}$$

$$f(I_i) = \sum_{p=1}^{n} (f_p(I_i) \cdot w_p) \tag{4.6}$$

Pathline fitness: Each local pathline property $f_p(I_i)$ is assembled from the per-vertex fitness indicators v_{ij}, depending on the number of vertices $|I_i|$ in the line (Eq. 4.5). Depending on the user configuration, $\odot$ is interpreted as *sum*, *avg* or *max*. Using the sum of the individual indicator values tends to prefer longer lines, as each vertex contributes to the overall fitness even if it has low fitness indicators. Using the average favors short lines, as each vertex with lower fitness indicators reduces the overall line fitness. Using the highest indicator eliminates the length of the line as a fitness factor. The final fitness value for a pathline is the weighted combination of all its pathline properties (Eq. 4.6). While individual weights (w_p) may be useful and add flexibility for advanced fitness function configurations, we do not explore this aspect and thus use the default value of 1 to weight all pathline properties. When using a fitness function $f(I_i)$ with respect to multiple surface attributes, normalization is important to avoid a bias, as the local fitness values may have different value ranges. Thus, we equalize the value ranges by normalizing all attribute values to the range $[0, 1]$. The user can further constrain this range, e.g. treating all values below or above a given threshold as zero. During the pathline integration, per-vertex fitness indicators and the overall line fitness are stored as attributes and thus are available for exploration and visualization purposes in the *Advanced Layer*.

Pathline and Context Visualization

Similarl to our previous approaches, pathlines are visualized as a strip of view-oriented quads inside a semi-transparent vessel model. The opacity of the vessel wall is calculated using a combination of the Fresnel and Phong terms to highlight both the vessel boundaries and the surface shape [151]. It is possible to map a scalar attribute, such as velocity magnitude, line curvature, or per-vertex or global fitness indicators onto the pathlines using a color map, opacity, or thickness (Figure 4.37).

2D Pathline Plots

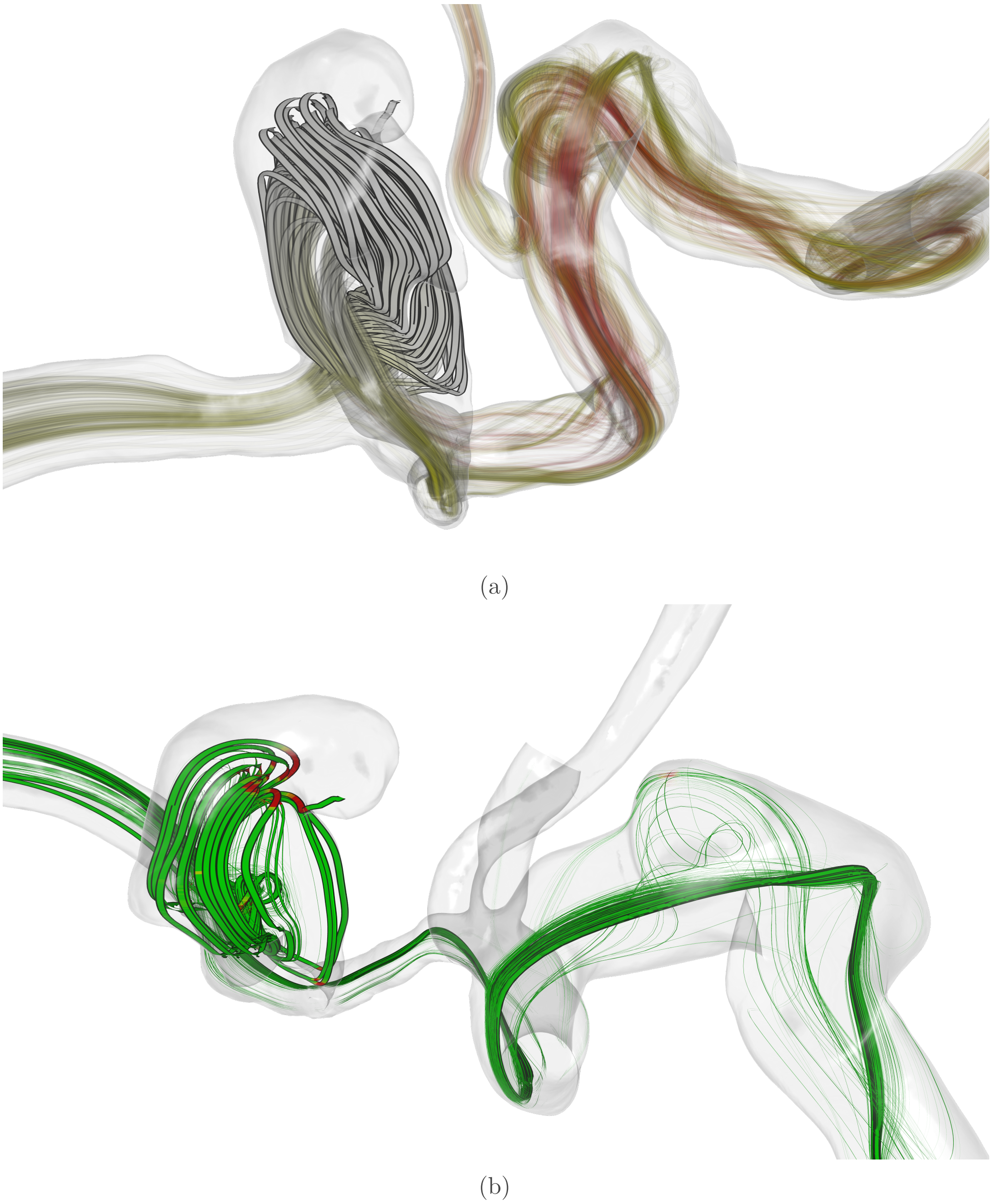
We implemented a 2D representation of each line set generated by the evolutionary algorithm to facilitate a comparative visualization for multiple sets of evolutionary lines. The lines are straightened and stacked onto each other (Figure 4.38). An attribute, such as flow velocity, can be mapped onto the lines using a color scale. The ordering of the lines in the diagram can be sorted according to fitness, overall length and length before the first aneurysm entry point. Additionally, the lines can also be clustered (see Section 4.3.2). Clusters are represented by colored bars on the left side of the plots. When batch-processing multiple datasets, the overview plots for each result are displayed in a grid view. The attribute color scale for each individual view is synchronized and displayed on top of the grid. The overview plots give insights to both the evolutionary algorithms and the clustering. When displaying multiple plots side by side, e.g. using different configurations for the evolutionary algorithm, they allow for a quick estimation of the differences and similarities between the generated line sets.

Clustering

The set of pathlines resulting from the evolutionary algorithm is often a composition of different flow structures. To automatically separate these flow structures, we cluster the pathlines based on pathline attributes along the line (Figure 4.39). We employ a density-based spatial clustering of applications with noise, i.e. the *DBSCAN* algorithm [212]. This approach was preferred against a hierarchical clustering algorithm as these algorithms require either manual interaction or an estimate of the number of expected clusters. DBSCAN requires a metric to measure similarity between individual lines. Due to the varying distance between line vertices as a result of the integration, a direct comparison of the lines is not possible. We therefore resample the lines so that the vertices of all lines have the same distance, with the longest line always having 64 vertices. We empirically determined 64 vertices to be an acceptable trade-off between computation time for the clustering and accuracy of the data representation.

To determine the similarity between two lines, we calculate the average difference s_v between the normalized attribute values $v_1..v_n$ at the first n vertices of both lines L_a and

L_b, whereas n is the number of vertices in the shorter of both lines (Eq. 4.7). However, this tends to group very short lines together with larger ones, as only a few vertices of the long line are taken into account. Therefore, we also calculate the difference between the normalized length l of both lines. The final similarity measure $s \in [0..\sqrt{2}]$ is calculated from s_v and the line length difference, ensuring that all lines in a cluster have similar length (Eq. 4.8).


(a)

(b)

Figure 4.37: Different parameter mapping options for pathlines. In (a), the residence time inside the aneurysm is mapped to the opacity and the flow velocity is mapped onto color using a temperature scale (white is slow). In (b), the per-vertex fitness indicator is mapped to color (green-to-red-scale) and the overall line fitness is mapped to thickness.

$$s_v = \frac{\sum_{p=1}^{n} |v_p(L_a) - v_p(L_b)|}{n} \tag{4.7}$$

$$s = \sqrt{{s_v}^2 + |l(L_a) - l(L_b)|^2} \tag{4.8}$$

Evaluation

In this section, we analyze our approach at first with respect to technical quality and performance, and discuss the method from an application point of view. First, we analyze our evolutionary algorithm with respect to its convergence and reproducibility to evaluate the *Technical Layer* of our workflow. Afterwards, we perform comparisons between our algorithm and a uniform seeding approach regarding the coverage of lines and its performance.

Convergence: The evolutionary algorithm empirically converges against an ideal result with respect to its fitness function. The convergence speed and the number of iterations needed to reach a sufficiently good result depends on the characteristic behavior of the fitness function within the search domain. There are two main parameters that control the trade-off between the quality of the result and the computation time. These are the population size and the number of generations. To determine a good default value for the number of generations and the population size, we have run the algorithm

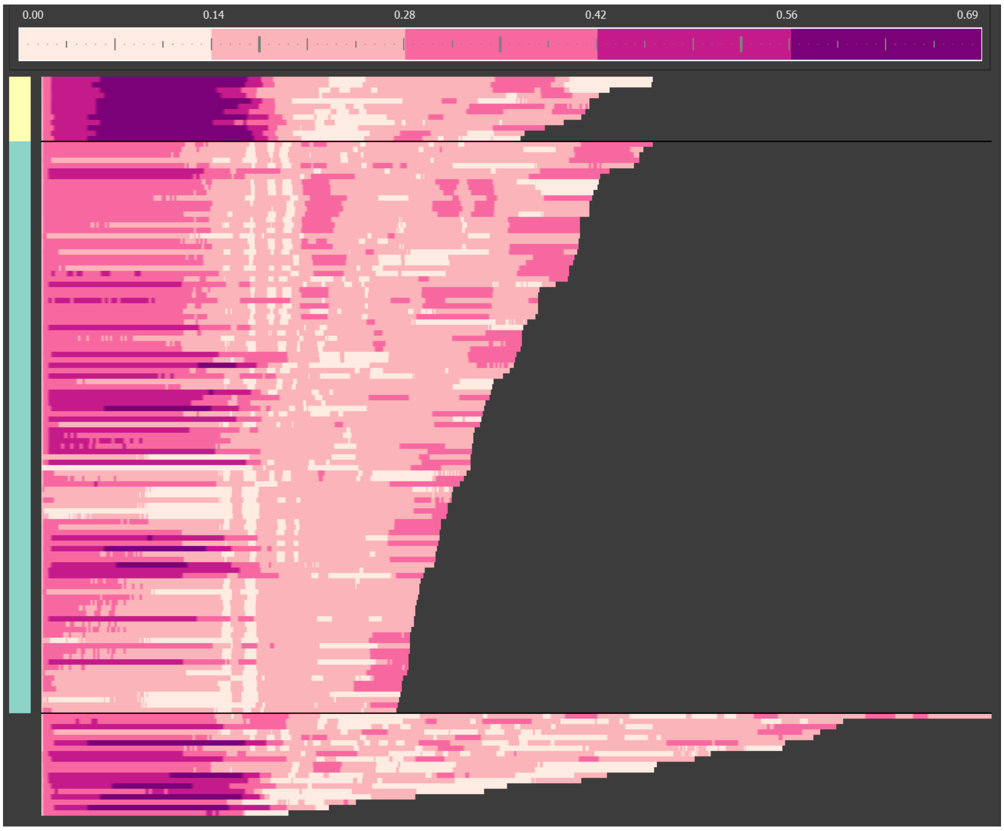


Figure 4.38: Two-dimensional overview plot for a line set generated by the evolutionary algorithm, with the flow velocity mapped onto the line color. The lines are sorted by length and grouped into clusters. Lines classified as noise by DBSCAN are grouped together at the bottom of the stack.

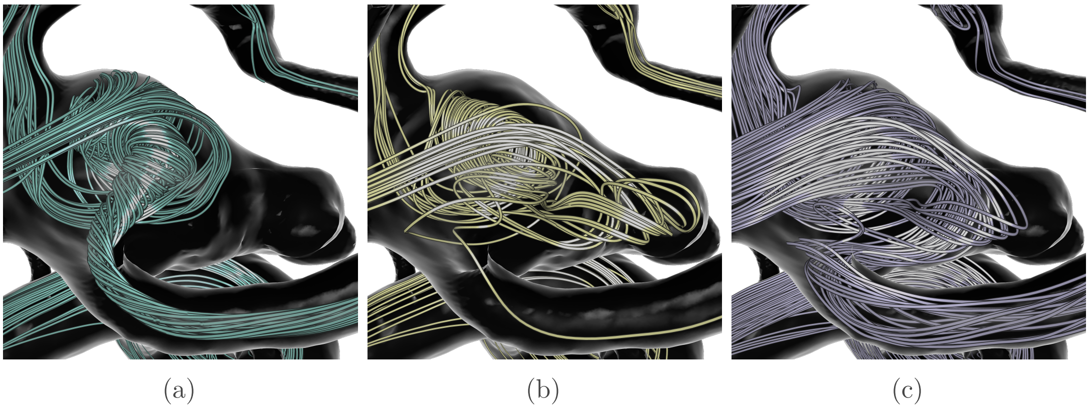
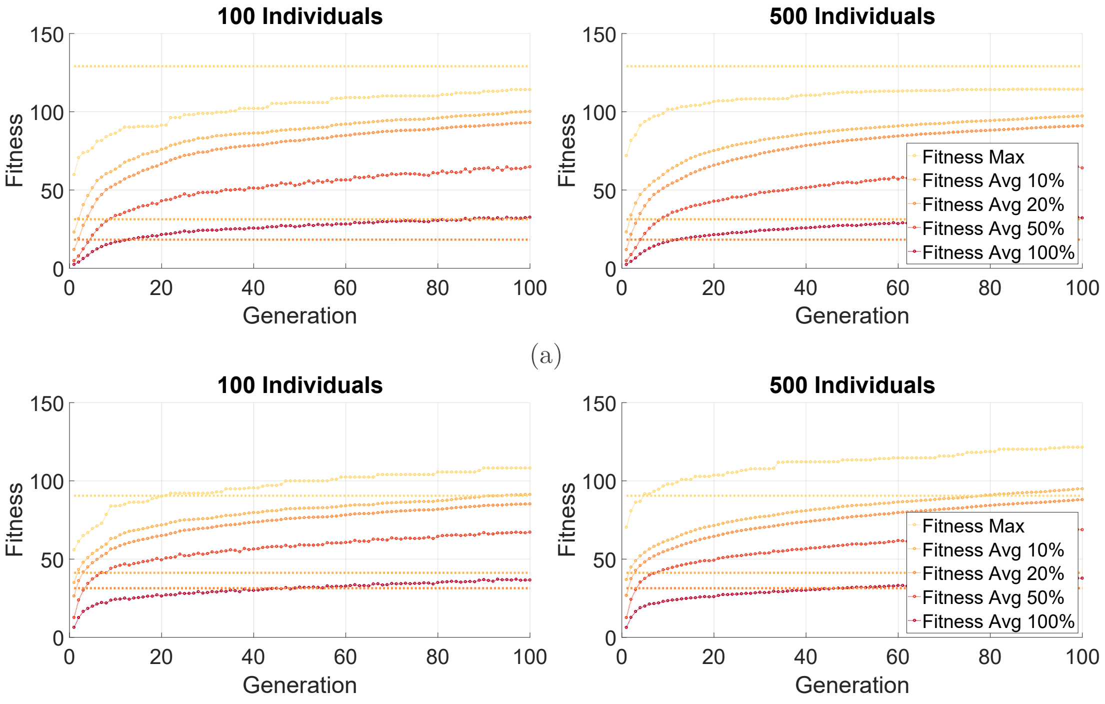

(a) (b) (c)

Figure 4.39: Clustering result: Pathlines forming a central vortex in the aneurysm (a), pathlines flowing into the bleb on top of the aneurysms (b) and pathlines leaving the aneurysm quickly without forming significant vortex structures (c).

on several datasets with different settings and have recorded the fitness values for each generation. Representative values are obtained by averaging the fitness values from ten runs for each population size.

The developments of the maximum fitness and different averages of the best lines are shown in Figure 4.40 for two datasets. The convergence speed and maximum fitness for the different population sizes is similar in both datasets, suggesting that the population size mainly influences the number of lines the algorithm produces. As a reference, the fitness reached by a uniform seeding approach for 1,000 line computations is plotted. In the first case (Figure 4.40a), the uniform seeding produces a single line with a higher fitness than all lines generated by the evolutionary algorithm. However, the 10%, 20% and 50% best lines by the evolutionary algorithm exceed those of the uniform seeding approach after only a few iterations. In the second case, the peak fitness value of the evolutionary algorithm soon exceeds the reference values (Figure 4.40b). After a few additional iterations, even the average fitness of the 10% best lines from the evolutionary algorithm comes close to the fitness of the best uniformly seeded line. This shows that better results as with the standard approach can be achieved even though areas with high fitness values are sparse and localized.

A fitness map shows the projected fitness of each individual on the seed plane for a single run of the evolutionary algorithm (Figure 4.41a). The projection is achieved by rendering a Voronoi diagram, where each seed point creates a cell that is colored according to the fitness of that point. With the given settings ($n_p = 200$, $p_e = 30\%$, $p_c = 0\%$, $p_m = 30\%$, $p_i = 40\%$), the evolutionary algorithm integrated $8,120$ pathlines in total. The map shows that the sampling rate automatically adapts to the fitness function. The resolution of the map is noticeably lower in areas with lower fitness, as the algorithm generally favours the exploration of high-fitness areas. To compare the result with fitness maps created using uniform seeding, we first use the same number of lines (Figure 4.41b). Then, we increase the number of lines to 10^6 lines to obtain a map that is close to the ground truth (Figure 4.41c). In regions with lower to medium fitness, the evolutionary algorithm is very close to the comparative run with the same number of lines. In areas with high fitness, it is almost identical to the densely sampled map (compare highlighted regions in Figure 4.41). The fitness values reached by all three approaches are summarized in Table 4.1. For both uniform sampling computations the fitness values are very similar. Integrating 10^6 lines leads to a better peak fitness due to a more exhaustive sampling of the seeding space. The evolutionary approach reaches



(a)

(b)

Figure 4.40: Average fitness curves (max fitness, top 10%, top 20%, top 50% and all lines) for 10 respective runs on two datasets a and b with identical configuration ($p_e = 30\%$, $p_c = 0\%$, $p_m = 30\%$, $p_i = 40\%$)*, but different population sizes (100 and 500). Horizontal lines indicate max fitness, top 10% and top 20% reached with uniform seeding of 1000 lines. Note that the best lines' fitness might be higher with uniform seeding compared to the evolutionary algorithm (see a), the averages are far below. Even after a couple of generations the evolutionary algorithm produces more lines with higher fitness than the uniform seeding approach.*

nearly the same peak fitness as the dense seeding, despite integrating significantly fewer lines. However, the 10%, 20% and 50% best lines have a significantly higher fitness than both of the uniform approaches.

To understand the influence of the population size on the convergence and thus the total number of lines that have to be computed, we recorded the fitness reached for identical configurations, but different population sizes (Figure 4.42). As expected, a greater number of individuals for each generation increases the speed at which the evolutionary algorithm converges, but also increases the computational effort.

Reproducibility: Due to the random nature of both the mutation and initialization of new points, two runs of the evolutionary algorithm with identical configuration will generate different line sets. To analyze the sensitivity of the results to this randomness, we performed multiple runs with identical configuration. Visually, the resulting pathline sets are highly similar (Figure 4.43). Additionally, we compare the fitness maps from 20 runs with different seed values and highlight regions with a relative fitness above a given threshold. The resulting images are thresholded and used to compute the mean pairwise Sørensen-Dice coefficient (DC) using the DC calculator by Tom Lawton [213]. DC is a summary measure of spatial overlap to compare the similarity of two binary images, where 0 means no overlap and 1 refers to complete overlap. For regions with a relative fitness above 0 (i.e. the seed regions that produce pathlines entering the aneurysm), the

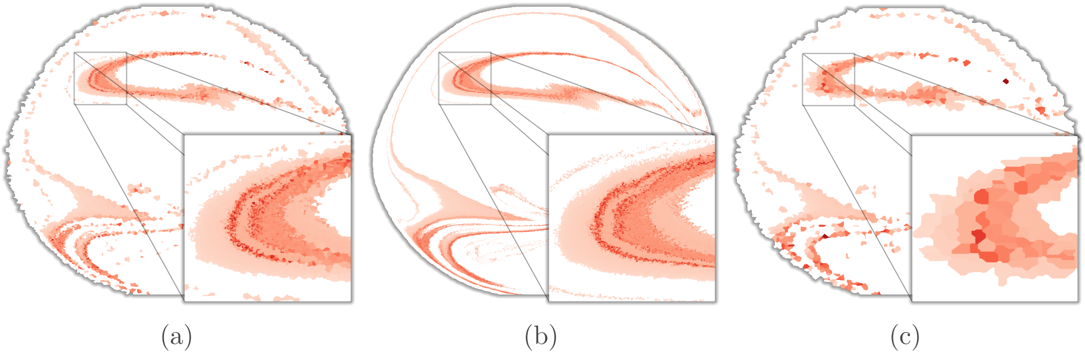


Figure 4.41: *Fitness map of the seed plane from the evolutionary algorithm and a total of 8120 line computations (a), a uniform sampling using the same number of lines (b) and 10^6 lines (c). Darker colors indicate higher fitness values, whereas in white areas the resulting pathline did not pass the aneurysm and therefore no fitness value is assigned. Seeding 10^6 lines takes multiple hours and thus is unfeasible in practical applications, hence uniform seeding approaches would likely miss the isolated and spatially confined maxima on the plane.*

	Ev. algorithm	$8,120$ *lines*	10^6 *lines*
Fitness Max	112.1	113.6	119.1
Fitness Avg. 10%	96.7	60.5	60.3
Fitness Avg. 20%	90.6	51.6	50.9
Fitness Avg. 50%	64	36.4	34.9
Fitness Avg. 100%	32.2	23.4	22.6

Table 4.1: *Fitness comparison between our evolutionary approach and uniform seeding with 8,120 and 10^6 pathlines, respectively.*

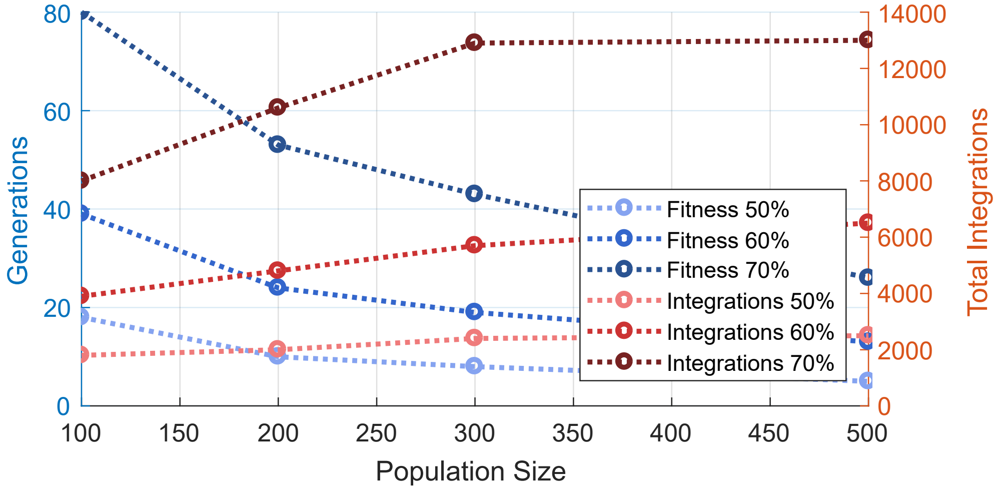


Figure 4.42: *Varying population size in comparison. Blue graphs show generations needed until the best 25 lines reach 50, 60 and 70% of the maximum fitness, respectively. Red graphs show total the amount of performed line computations.*

DC reached a value of 0.87. The DC for regions with a relative fitness above 25% and 90% of the maximum reached 0.72 and 0.42, respectively. Visual inspection shows that the differences are mostly located around the edges of the detected regions. Increasing the threshold lowers the overall size of the regions, thus increasing this effect and lowering the DC. Yet, regardless of the threshold, overlaying all 20 fitness maps shows that the detected regions are visually similar (Figure 4.44). Thus, we can conclude that, despite the variances in the results, the algorithm acts in a predictable and stable way. However, as exact reproducibility may be important in some use cases, our framework exposes the seed value for the random number generator to the user. Running the evolutionary algorithm twice with an identical configuration *and* random seed value will always yield identical results.

Coverage: Uniform seeding strategies on the inflow plane suffer from poor line-coverage in hard-to-reach parts of the aneurysm, especially in blebs. To demonstrate the ability of our algorithm to reach these areas, we compared it against ParaView (version 5.5.0) as well as our own line integrator using a uniform seeding strategy. In all cases, we seeded 300 individual lines. Both ParaView as well as our application used uniformly distributed seed points at the inlet for the uniform seeding approach. As we were not able to entirely replicate our own integration parameters in ParaView, the resulting lines are slightly different. The evolutionary algorithm was configured with $n_p = 300$,

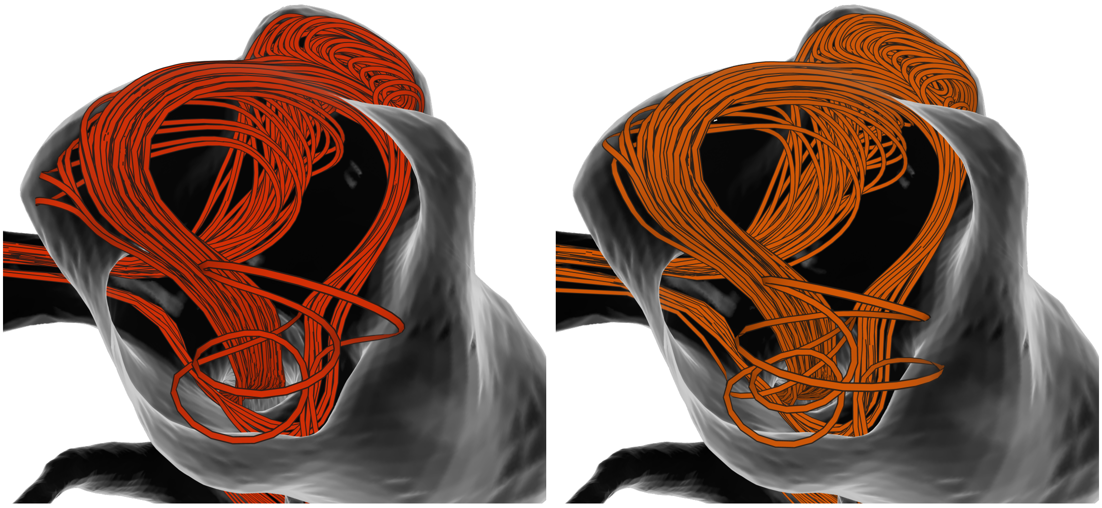

(a) (b)

Figure 4.43: Pathlines from two evolutionary algorithms using the same configuration, but different random seeds.

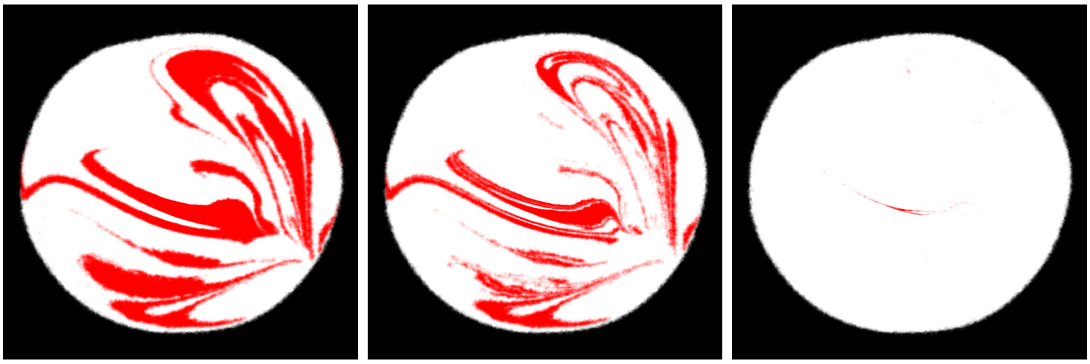

(a) (b) (c)

Figure 4.44: Average projection of the 20 thresholded seed plane fitness maps with the same fitness function, but different random seed values. Thresholds for the relative fitness were set to 0 % (a), 25 % (b) and 90 % (c)

$p_e = 30\%$, $p_c = 0\%$, $p_m = 30\%$, $p_i = 40\%$, using the total length of each line in the aneurysm as its fitness function and iterated over 30 generations. Both uniform seeding approaches produce similar results (Figure 4.45a and 4.45b), with the top part of the aneurysm nearly entirely empty. The evolutionary seeding, on the other hand, manages to capture the flow in the entire aneurysm (Figure 4.45c).

Performance: The most time-consuming part of the evolutionary algorithm is the evaluation of the fitness function, as it requires the full integration of a pathline and subsequent calculation of the fitness indicators for each pathline vertex. Figure 4.46 presents a comparison between the result from a uniform seeding approach with subsequent filtering and our evolutionary algorithm using an identical fitness criterion. For the uniform approach, $1,000$ lines were seeded in total, whereas 23 lines remained after filtering (Figure 4.46a). For the evolutionary approach, 460 lines were seeded in total ($n_p = 100$, $p_e = 30\%$, $p_c = 0\%$, $p_m = 30\%$, $p_i = 40\%$) over ten generations and the best 47 lines were used (Figure 4.46b). 53 lines with a fitness value of 0 were discarded. Despite seeding less than half of the amount of lines than with the uniform approach, the evolutionary algorithm produces a denser set of lines fulfilling the criterion. Integrating the 1000 lines using the uniform approach took 53 seconds (without performing filtering) on an *Intel i5-6500* with four cores clocked at 3.2 GHz, whereas the evolutionary algorithm took 45 seconds (including the evaluation of the fitness function). Integrating and evaluating pathlines to computing the densely seeded map from Figure 4.41c took eight hours. Since only small areas on the seeding plane result in lines with a high fitness value (Figure 4.41), the uniform seeding approach struggles to position a sufficient amount of seed points in these areas without strongly oversampling areas with lower fitness. Based on the results (Figure 4.42), we concluded that a population size of 200 or 300 is the best trade-off between fast convergence, total amount of line computations including evaluation and run stability, i.e. less variations between runs with different random seeds. While a

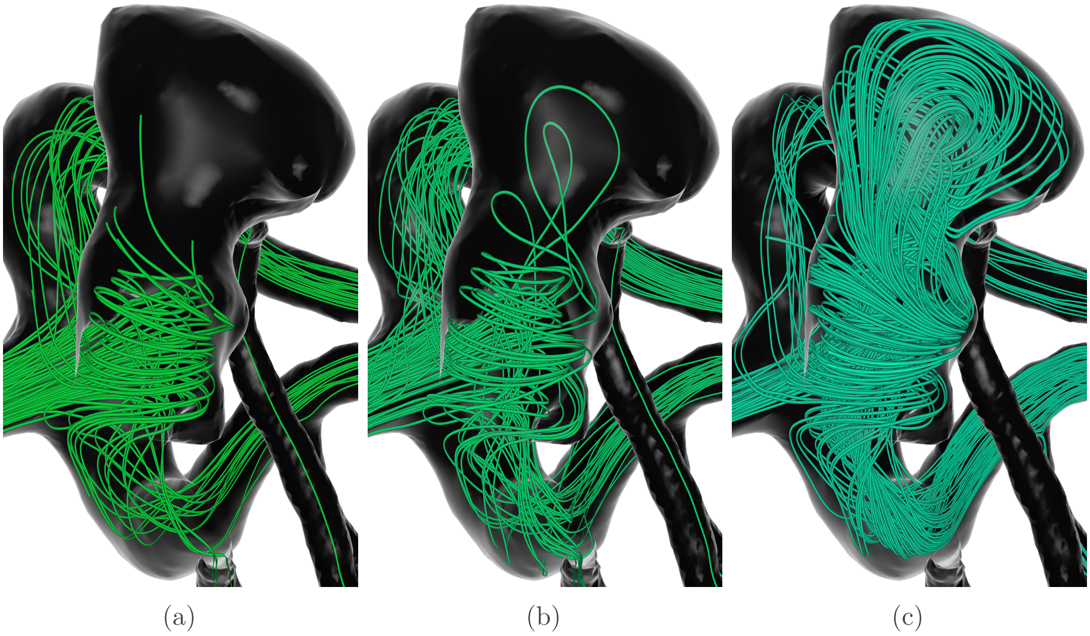


Figure 4.45: Comparison of expressive line sets for an aneurysm: We compare the output of a uniform seeding approach using ParaView 5.5.0 (a) and our in-house integrator (b) with our evolutionary algorithm (c). In both a and b, almost no lines reach the dome and therefore leave it heavily underpopulated, whereas the dome is densely covered in c.

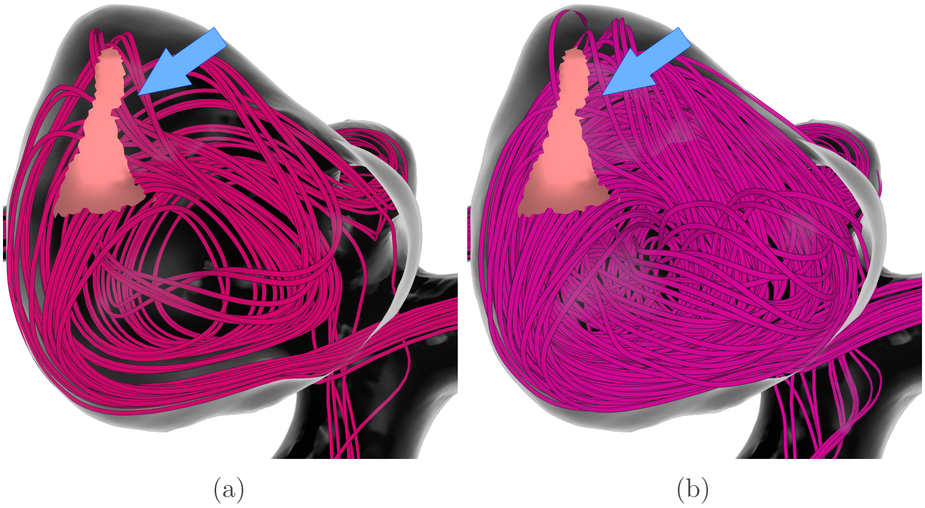


Figure 4.46: *Comparison of line coverage using uniform seeding methods (a) and our method (b) when searching for pathlines passing near a specific area on the vessel surface (arrow).*

population size of 100 also provides reasonable convergence speed and stability, the total amount of generated lines may be insufficient to facilitate a dense 3D visualization.

We performed a general evaluation with an experienced neuroradiologist (*Physician A*), who was also involved in the design process of our framework. Afterwards, we conducted a more specific evaluation of the *Clinical Layer* of our workflow with three additional neuroradiologists (*Physician B*, *Physician C* and *Physician D*), who were not familiar with our tool, to validate its clinical usefulness with respect to treatment planning. As suggested by Preim et. al, such a procedure is crucial to achieve meaningful evaluation results [214]. Our goal was to evaluate

1. which fitness function configuration produces the most satisfactory line bundles for this purpose.

2. whether the visualization of hemodynamic information using evolutionary pathlines influences the treatment decision.

To answer question 1, we asked *Physician A* to compare the resulting lines of different fitness functions. The comparison consisted of pathlines optimized for length and to pass by surface areas in the aneurysm with high and low wall shear stress (WSS) as well as high and low oscillatory shear index (OSI). The physician was able to easily identify differences between line bundles optimized with respect to high or low OSI. Interestingly, he was able to infer the location of high and low OSI or WSS values on the surface by comparing their respective line bundles. To evaluate the blood flow for treatment decisions, he preferred the length-based fitness function, as it ensured that the entire aneurysm was filled with pathlines. He did not find significant differences in the pathline bundles created using the OSI-based fitness function in comparison to the WSS-based fitness function. This could be explained with the fact that the OSI is derived from the WSS. Therefore, both attributes lead to similar line bundles.

In clinical practice, the placement of stents with or without coiling to treat aneurysms is carried out based on the patient-specific anatomy. To assess a benefit in treatment planning, we had to identify well suited test cases first. Therefore, we conducted a quick

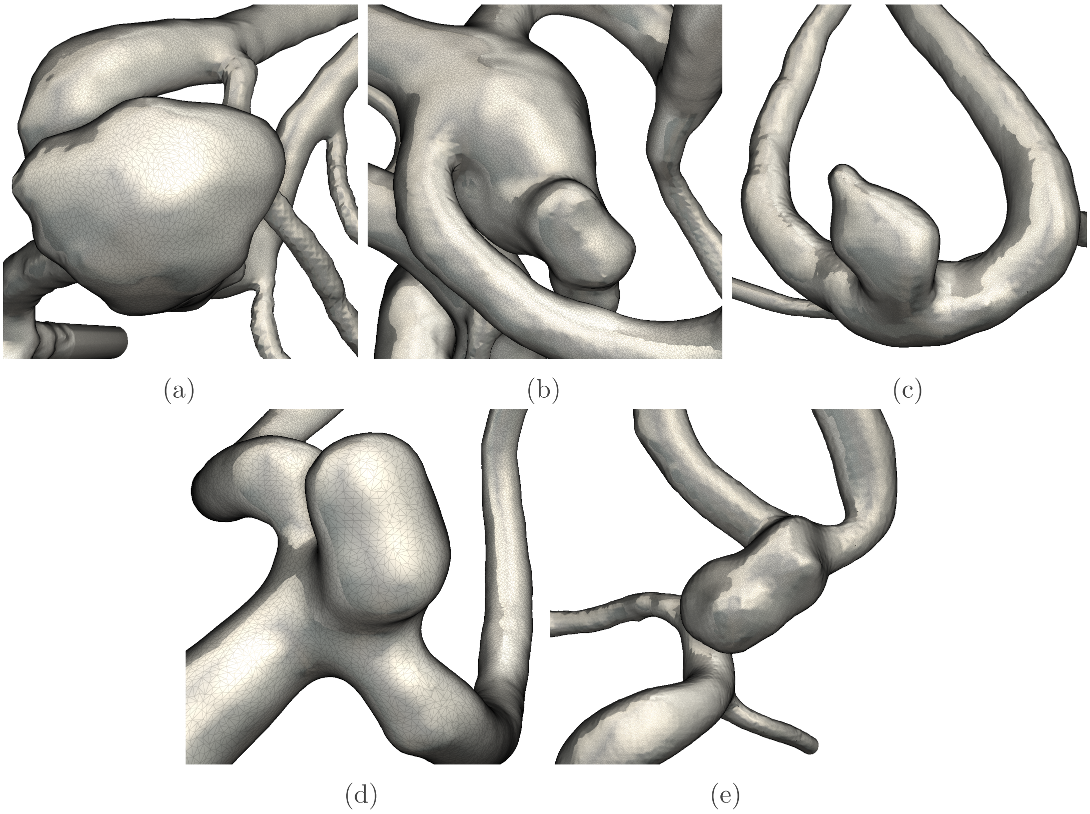


Figure 4.47: *Selected cases for our evaluation comprising four aneurysms at the anterior communicating artery: Case 1 – male, 36 years (a); Case 2 – male, 63 years (b); Case 3 – male, 86 years (c); Case 5 – female, 38 years (e) and one at the bifurcating middle cerebral artery: Case 4 – female, 59 years (d).*

survey of a database containing more than 100 cerebral aneurysms with *Physician A*, who had been treating these patients. We selected five cases with a challenging anatomy, i.e. aneurysms that are located at bifurcations with almost symmetrical outlet configurations (Figure 4.47).

To answer question 2, we presented the other experts (*Physician B-D*) with these five cases. They varied with respect to their work experience of endovascular treatment, including the hospital director (20 years of neurointerventional experience), a senior physician (9 years) and a novice physician (1 year). They were blinded to the applied treatment and patient outcome, but provided with clinical information such as the patients' sex and age.

The first visualization only depicted the vessel surface without any hemodynamic information. The physicians were asked about their treatment decision based on the given visualization. Afterwards, we presented them with the same dataset using our visualization techniques, including pathlines extracted by the evolutionary algorithm, and asked whether they would revise their decision. In both cases, they had full control of the viewing direction. For this evaluation, we chose a fitness function based on both line length and residence time of the flow inside the aneurysm, as it achieves a pathline coverage superior to uniform seeding approaches and covers the whole aneurysm. The results are provided in Table 4.2.

	Physician B	*Physician C*	*Physician D*
Case 1 - before	C & S	C & S	Y – S
Case 1 - after	C & S	C & S	C & S
Case 2 - before	Y – S	C & S	Y – S
Case 2 - after	Y – S	C & S	Y – S
Case 3 - before	C	C & S	Y – S
Case 3 - after	C	Y – S	C & S
Case 4 - before	Y – S	C & S	Y – S
Case 4 - after	Y – S	Y – S	Y – S
Case 5 - before	Y – S	C & S	Y – S
Case 5 - after	Y – S	Y – S	Y – S

Table 4.2: *Results of our evaluation for physicians B, C and D, when only seeing the 3D view for anatomical information (before) and after seeing our pathline visualization (after). Treatment decisions were: stent-assisted coiling (C & S), y-stenting (Y–S), i.e. implanting two stents, and coiling without stenting (C). Changes in treatment decisions are highlighted via color-coding.*

In four of five cases, at least one physician changed his or her treatment decision after exploring our evolutionary pathlines. These changes were motivated by an improved understanding of the intra-aneurysmal flow and its splitting into the outlets, including flow patterns. In three cases, these changes lead to more consistency in the final decision between the physicians. The third aneurysm was the most challenging one, as two physicians changed their minds and the final decisions are not consistent between them. On the one hand, their initial decisions were similarly inconsistent, but on the other hand, we selected challenging cases. Interestingly, the most experienced physician never changed his mind.

Discussion

We presented an adaption of the recently developed evolutionary lines to the domain of medical blood flow analysis by carefully adjusting the genetic operators to fit clinical and research requirements. We showed the usefulness of an evolutionary algorithm for seeding pathlines in cerebral blood flow data. Our approach shows significant improvements over existing seeding strategies, both in terms of computational effort and quality of results. Better coverage of hemodynamically interesting regions, such as aneurysm domes or blebs, is the major advantage over previous work. Thus, flow patterns can be identified in a more reproducible manner, which is essential to assessing the aneurysm rupture risk.

We mainly examined the feasibility of using evolutionary lines to visualize hemodynamics in cerebral aneurysms. The design of our fitness functions is tailored towards the examination of simulated blood flow in cerebral aneurysms, but could also be adapted to fulfil the specific research requirements for other blood vessels or data acquisition methods. An interesting extension would be the introduction of a normalization between the fitness value ranges of surface and pathline properties, in order to allow more combinations of properties and thus increase the flexibility of the fitness function. Further improvements, especially with respect to the conversion speed, could be achieved by an in-depth analysis of the parameter space of the genetic operators.

Currently, each individual in our evolutionary algorithm represents a single pathline. For the future, we plan to support additional types of structures, such as streaklines or line bundles. The evolutionary algorithm treats and optimizes each individual as a separate entity, therefore increasing the fitness of singular lines, but not guaranteeing diversity between the resulting lines. By using line bundles instead of singular lines as individuals, the coverage within the aneurysm could be further improved. A different solution to this problem could be the usage of seed plane fitness maps, which are generated as a side-product of the evolutionary algorithm, for a final seeding step.

VICTORIA - An interactive online tool for the Virtual neck Curve and True Ostium Reconstruction of Intracranial Aneurysms 4.3.3

The content of the following section is based on:
Benjamin Behrendt, Samuel Voss, Oliver Beuing, Bernhard Preim, Philipp Berg, and Sylvia Saalfeld. “VICTORIA - An interactive online tool for the VIrtual neck Curve and True Ostium Reconstruction of Intracranial Aneurysms”. In: *Proc. of Bildverarbeitung für die Medizin (BVM)*. 2020, pp. 209–214

To evaluate these quantitative values of intracranial aneurysms, such as wall shear stress, the separation of the aneurysm from its parent vessel is required. Furthermore, therapy planning requires a detailed knowledge of the individual aneurysm neck size to select an appropriate treatment strategy and device, respectively [216]. In addition, a virtual separation of the aneurysm from the parent vessel allows for the extraction of parameters from a 3D model rather than 2D-projected images, which are used in clinical routine and suffer from increased user as well as viewing angle dependency [217]. The separation of the aneurysm from the healthy parent vessel is often realized using a planar cut-plane [218, 219], which might be error-prone for complex aneurysm shapes. A fundamental problem of this concept is the missing ground truth as studies report strong variations for this procedure between medical experts, yielding increased interobserver variability for subsequent evaluations. This might lead to insufficient analyses and in consequence to unreliable conclusions.

To make further steps towards consensus, we initiated the *VICTORIA* (*VI*rtual neck *C*urve and *T*rue *O*stium *R*econstruction of *I*ntracranial *A*neurysms) project. We developed a web-based solution, combining a client based on HTML and JavaScript and a server part utilizing PHP and the Matlab Runtime environment. Within this study, participants are requested to identify the neck curve of five virtual intracranial aneurysm models and manipulate the resulting ostium surface to model the wall of the original parent artery. For the definition of these neck curves and ostia, highly experienced physicians are required. Since their availability for user studies is limited due to the clinical work load, the application is available over the internet, solely requiring a up-to-date web browser. Thus, we can gather expert-knowledge from physicians as well as biomedical engineers. The study has been successfully launched at `https://VICTORIA.cs.ovgu.de/`.

Method

Within the VICTORIA study, participants are requested to identify the neck curve of patient-specific IA models extracted from 3D digital subtraction data. The study is conducted using a specialized application, consisting of two parts: a web-based client written in HTML and JavaScript, and a server written in PHP and using Matlab runtime environment. Thus, the VICTORIA study can be accessed via the internet using a web browser. Between client and server, the data is exchanged in the JSON format, which is highly flexible and well supported by both browsers and PHP. The server stores user-submitted data in a relational database, thus allowing for easy sorting and filtering of the data (Figure 4.48).

In order to keep the motivation to participate high, the user is directly presented with five datasets and two tasks, respectively. For both tasks, a surface mesh of an intracranial aneurysm and its parent vessel is shown. The rendering is performed using the WebGL2 API, which is available in most current desktop browsers. To increase shape perception, the mesh is illuminated using the Phong lighting model. The user has control over the camera (rotation, zoom and panning), although the interaction range is limited to ensure that the neck region is always visible and centered in the image.

For the first task, the user is asked to draw a neck curve onto the vessel mesh by selecting surface vertices (Figure 4.49b). These vertices can be selected in an arbitrary order and also be individually removed, as to allow the user to apply corrections and increase usability. To connect the selected vertices to form a neck curve, we are looking for the shortest circular path that connects all points. Therefore, the surface triangle mesh is interpreted as a bidirectional graph and the shortest paths between the selected points are then calculated using the A* algorithm by Hart et al. [220]. However, the A* algorithm is only designed to find the shortest path between two points, not the shortest path connecting a list of unordered points. Therefore, we sort the list of points automatically using a distance matrix storing the length of the shortest path between all points (Algorithm 2). Whenever a new point is added to the matrix, the distances to all other points are determined using A*. To remove a point, the corresponding row and column is deleted from the matrix.

After each change to the matrix, the list of points is sorted. We begin with a list containing only one randomly chosen point. As long as there are points that have not been added to the list, we take the last entry, look up the closest point (based on the distance matrix) that has not been added yet, and add it to the back of the list. After all points have been processed, the resulting list contains the shortest path connecting all neck points selected by the user. The last and first point in the list will only be connected if their distance is shorter than the longest path between any of the other adjacent points in the list (Figure 4.49c). This prevents the neck curve from closing before the user has finished adding all desired points. The sorting and path finding algorithms are implemented in JavaScript and are performed entirely on the client side. Even on less powerful devices, they run interactively without any noticeable delay.

After submitting the neck curve, the server calculates the associated ostium surface mesh. To not overstrain the client hardware, this step is calculated on the server using a Matlab script based on the work of Saalfeld et al. [211]. The Matlab script receives the previously selected surface meshes and downsamples them to a predefined threshold

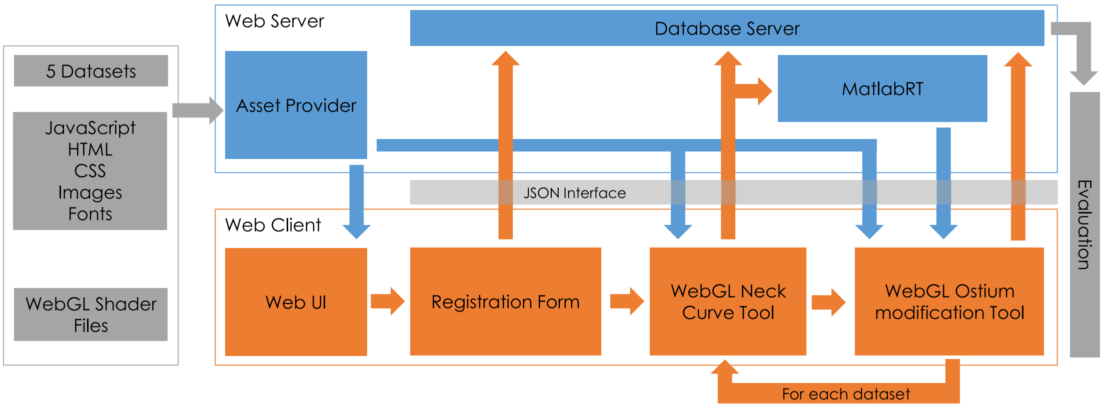


Figure 4.48: *Illustration of the different parts of the VICTORIA web application.*

(Figure 4.50a). For the illustration, the neck curve points are resampled to 32 points. The ostium triangulation is analytically defined and centered around the origin, with all z-coordinates set to 0 (Figure 4.50b). Next, the border points are replaced by the neck curve points and simple Laplacian smoothing (50 iterations, $\sigma = 1.0$) is applied to all vertices except the border points, resulting in the smooth ostium surface (Figure 4.50c).

The second task consists of the adjustment of the previously calculated ostium surface (Figure 4.49d). This is important, as a better approximation of the inflow area allows to better approximate the parent vessel's original geometry in further postprocessing steps. During this task, only the backfaces of the vessel are rendered to reveal the ostium. The user can grab and drag any vertex of the ostium surface that is not part of its border (Figure 4.49e). To keep the interaction as simple and straightforward as possible, the vertices can only be moved in the direction of the average surface normal of the ostium, i.e. *up* and *down*. As moving one vertex at a time would be tedious, dragging one vertex by a distance of v also moves the surrounding vertices by a distance of v_n based on their normalized Euclidean distance d_n to the original vertex (Figure 4.49f). The parameter a controls the size of the affected area. We empirically determined a value of $\frac{1}{3}$ of the average radius of the neck curve to result in an interaction that feels natural. This second task can be considered optional, as it is possible for the user to directly accept and submit the ostium surface generated by the server without performing any modifications.

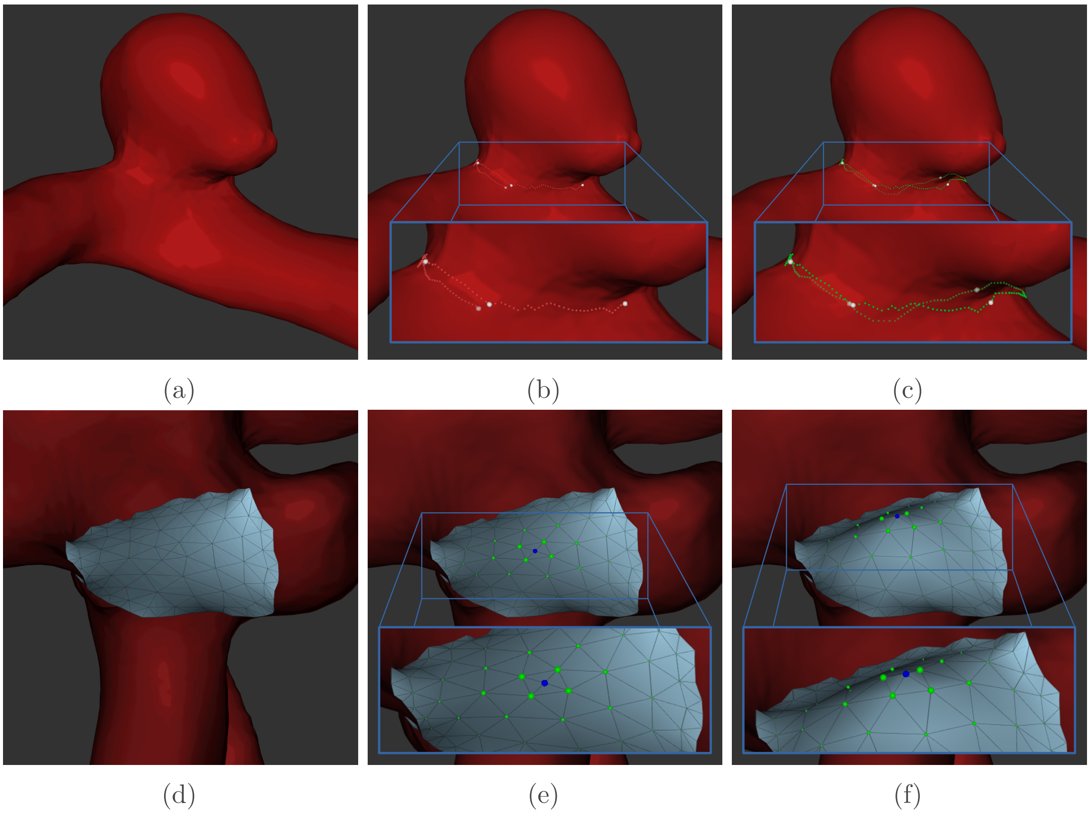


Figure 4.49: *The different steps of the ostium definition; 3D visualization of the surface model with WebGL (a). The user can interactively select points on the aneurysm surface (b), which are automatically connected. If the first and last points are close to each other, the neck curve is automatically closed (c). In the second step, an automatic ostium triangulation is provided (d). The user can hover over the points and the active point is highlighted (e). The point can be moved (including a reduced movement of its neighbors) until the user is satisfied with the ostium shape (f).*

Algorithm 2 Pseudocode to sort the list of points

```
$G \leftarrow mesh$                                ▷ Use mesh as bidirectional graph
$M \leftarrow []$                                  ▷ Distance matrix
$points \leftarrow []$                             ▷ Unordered list of points
$ordered \leftarrow []$                            ▷ Ordered list of points

procedure ADDPOINT($v$)                            ▷ Add a vert. from graph to unordered list
    for all $p$ in $points$ do
        $M_{vp} \leftarrow M_{pv} \leftarrow A^*(G, p, v)$    ▷ Use $A^*$ do calc. dist. betw. $p$ and $v$ in $G$
        $M_{vv} \leftarrow 0$                      ▷ Diagonals of distance matrix are always 0
    end for
    Add $v$ to $points$
    $ordered \leftarrow$ SORTPOINTLIST
end procedure

function SORTPOINTLIST                             ▷ Produce an ordered list as neck curve path
    $u \leftarrow points$                          ▷ Points that have not been visited
    $o \leftarrow []$                              ▷ Empty ordered list
    Add POP($u$) to $o$                            ▷ Move one element to the ordered list
    while $u$ not empty do
        $p_0 =$ TOP( $o$ )                         ▷ Get last point from ordered list
        $p_1 \leftarrow$ CLOSESTPOINT($M, p_0, u$)  ▷ Find closest point $p_1$ to $p_0$ using $M$
        Remove $p_1$ from $u$                      ▷ Move $p_1$ from unordered to ordered list
        Add $p_1$ to $o$
    end while
    return $o$                                     ▷ Return ordered list
end function
```

$$v_n = v \cdot (max(0, d_n - (1 - a)) * a^{-1})^{0.75} \tag{4.9}$$

Finally, we implemented a registration form combined with a questionnaire. Users enter their name, e-mail address, occupation and affiliation, and answer questions about their experience with intracranial aneurysms. With this information, an examination of differences between user groups (e.g., physicians vs. engineers) can be conducted in the future.

Results

The VICTORIA study can be accessed at https://VICTORIA.cs.ovgu.de/. We tested this interactive survey sucessfully with three medical cooperation partners. Initially, they requested more hints and icons regarding the controlling of the application, which we included. The neck curve definition requires approximately two minutes per case and less than ten minutes overall. The study requires the use of current versions of *Mozilla Firefox*, *Google Chrome* or *Microsoft Edge*. Other less commonly used browsers, such as *Apple Safari* or *Microsoft Internet Explorer*, do not support the required WebGL technologies. The evaluation of the submitted ostia segmentations is not part of this work, but ongoing research.

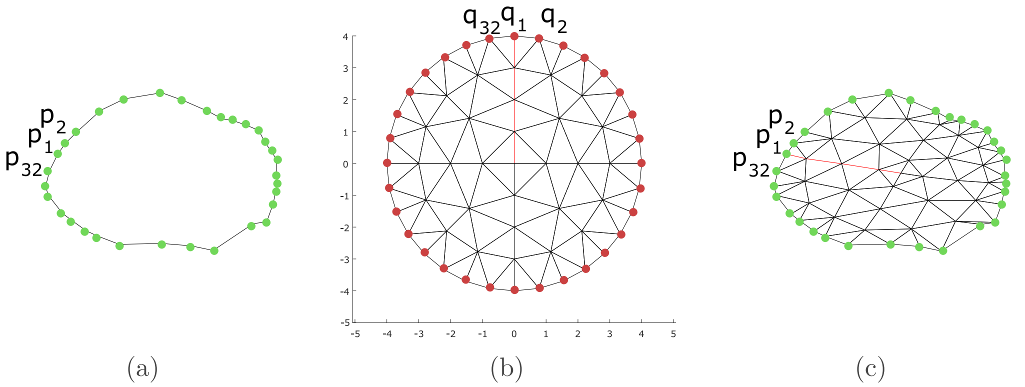


Figure 4.50: *Illustration of the ostium triangulation; The neck points are resampled to 32, p_1 - p_{32} (a). The predefined triangulation is centered around 0 with border points q_1 - q_{32} (b). Next, q_1 - q_{32} are replaced by p_1 - p_{32} and all points undergo Laplacian smoothing yielding the triangulated ostium surface (c).*

Discussion

Currently, no neck curve and ostium ground truth is available, yet it is required for many postprocessing steps of intracranial aneurysm models. We presented an approach that allows for an easy participation of international field experts and physicians by utilizing a combination of JavaScript, PHP and Matlab-programs, including 3D visualizations, editing and modification options.

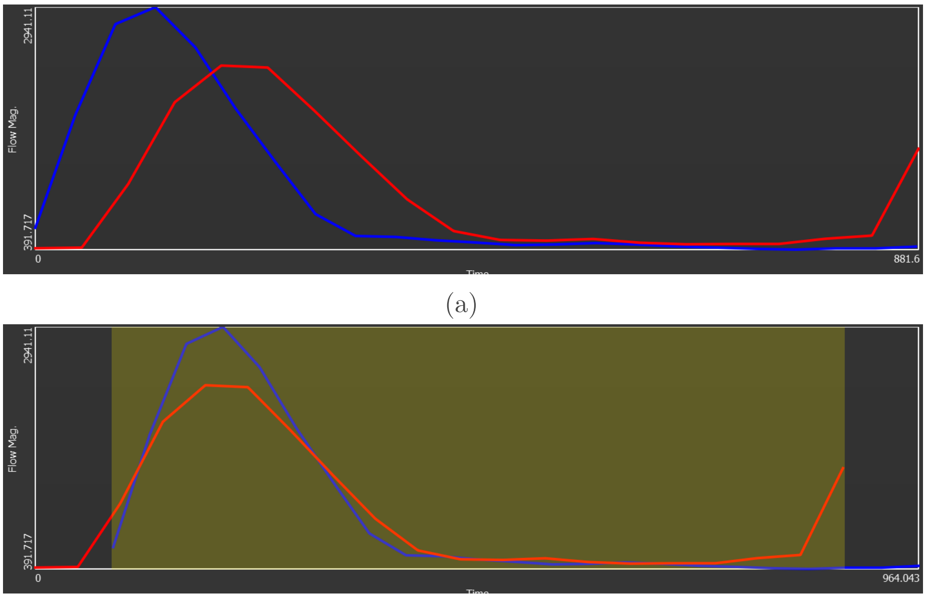
Based on a ground truth ostium definition, a precise morphological evaluation of the 3D aneurysm shape is further promoted, which is highly beneficial for the quantification of hemodynamic flow simulations [221]. Furthermore, a 3D neck curve determination and subsequent parameter evaluation is superior to 2D analysis [217]. Particularly, since relevant blood flow parameters that are associated with rupture (e.g. normalized wall shear stress, shear concentration index, oscillatory shear index [222]) need to be calculated with high accuracy, wrong aneurysm-vessel-separation or high user-dependency can lead to clear variations regarding the analysis.

Furthermore, we expect our survey results to be utilized as ground truth data for deep learning-based ostium extraction approaches, e.g. similar to the MeshCNN network [223]. Finally, our architecture can be easily adapted to other medical image processing questions that require 3D models and user interaction.

4.3.4 A Framework for Visual Comparison of 4D PC-MRI Aortic Blood Flow Data

The content of the following section is based on:
Benjamin Behrendt, Sebastian Ebel, Matthias Gutberlet, and Bernhard Preim. "A Framework for Visual Comparison of 4D PC-MRI Aortic Blood Flow Data". In: *Eurographics Workshop on Visual Computing for Biology and Medicine.* The Eurographics Association, 2018, pp. 117–121

The ability to compare and contrast multiple datasets is key to gain new insights about the inter-relation between changes in blood flow and vessel morphology. For 4D PC-MRI data, such a comparison is often facilitated by extracting and quantitative measures or visualizations from each dataset. A common way to extract measurements at medically interesting positions within the anatomy is the placement of measuring planes, which can be used to obtain measures such as stroke volume and regurgitation fractions [192]. However, this approach not only requires the user to know what specific type of differences they are looking for, but it also bears the risk of overlooking possibly valuable details. Additionally, even small changes in the positioning of the planes can influence the resulting measurements, so the comparability of measures between different datasets with varying anatomical shapes is limited. To solve these issues, we propose an explorative approach to comparing 4D PC-MRI data with a flexible tool for the analysis of multiple 4D PC-MRI datasets. In addition to exploring singular datasets, the analysis of differences and similarities between multiple datasets can lead to further insights, regarding not only the specific characteristics of certain pathologies, but also the influences of data acquisition and reconstruction techniques.



(a)

(b)

Figure 4.51: *Initial (a) and registered (b) flow curves of two datasets with overlapping time steps marked in yellow.*

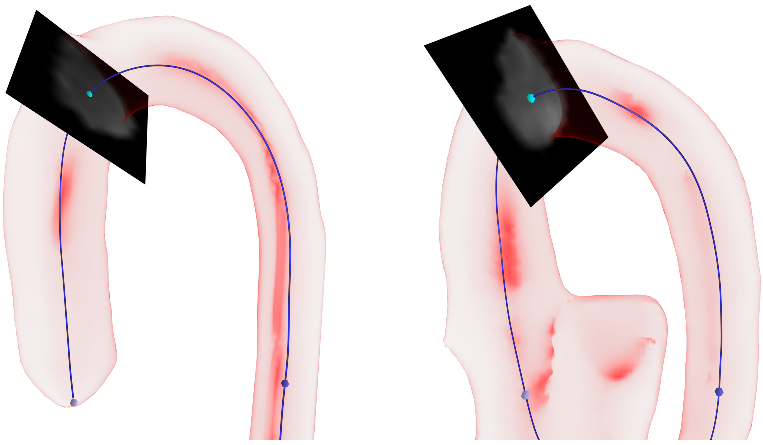


Figure 4.52: *Two datasets with a normalized centerline area between two markers.*

Method

In cooperation with our clinical partners, we developed a workflow concept for the explorative comparison of 4D PC-MRI aortic blood flow data. The tools our partners are currently using are focused primarily on the visualization of single datasets. They are, however, interested in finding systematic differences between multiple datasets that are part of an ongoing study. A core component of our concept is image-based comparison using the juxtaposition of visualizations. Additionally, we provide feature-based comparisons by calculating and visualizing differences in flow velocity, direction and flow jet position of both datasets. This section presents the workflow for our toolset, starting with the registration steps necessary for comparing datasets. Since we display information directly related to one of the datasets as well as to comparative measures between the datasets, we employ color-coding to distinguish these types of visualizations (red for the first dataset, blue for the second). Comparative visualizations are generally marked white.

Registration: For meaningful comparisons, the heart cycles depicted in both datasets need to be matched with each other. While performing the temporal registration, both datasets are represented by their flow curves. They are generated by adding up the flow magnitude from all voxels inside the segmentation for each time step. The resulting curve clearly shows the systolic and diastolic phases of the heart cycle (Figure 4.51a). In this case, the red dataset includes more than a single heart cycle, as a second systolic phase begins at its end. The user can manually adjust the offsets by dragging the curve onto a new position (Figure 4.51b). Additionally, the user has the option to add a scaling parameter to the datasets' time frame in order to account for different heart rates. Due to the low temporal resolution of the datasets, a pixel-perfect registration of both curves is usually not required.

The spatial registration consists of three tasks: *Visual registration*, *centerline normalization* and *rotational alignment*. The *visual registration* can be performed mostly automatically by applying the patient matrix of each dataset to the surface and pathline models. Next, the *centerline normalization* is performed by placing landmarks. Initially, two markers are automatically placed at the beginning and end of the centerline, which can be independently moved along the centerlines of both datasets. They are used to crop the centerline, so that the remaining part covers the same anatomical areas in both datasets. In most cases, this is sufficient to normalize the centerline. However, in case

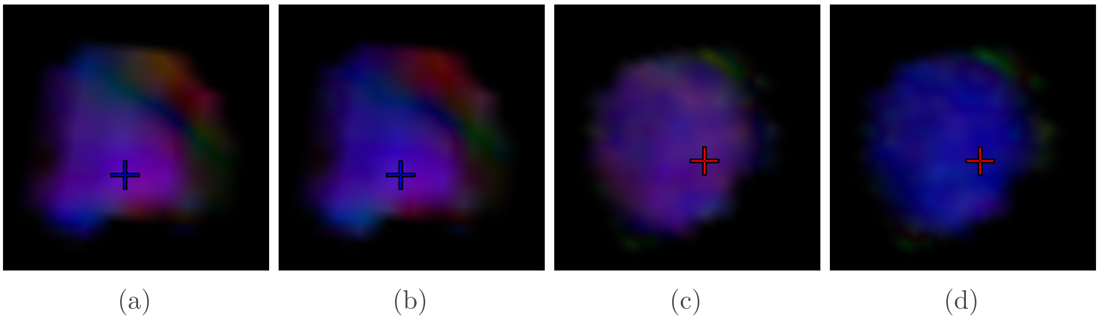


Figure 4.53: *Two planes sampled from two datasets (a, b and c, d) at the same anatomical position colorized by mapping the flow vectors to RGB colors. The first images (a, c) use the data coordinate system, the second images (b, d) use the local plane coordinate system.*

of deformations of the vessel shape or the presence of bypasses, additional markers can be placed to achieve a satisfying normalization. The *rotational alignment* needs to be performed manually by individually rotating each plane by up to 180°. The easiest way to find the correct angle is to move the planes to the supra-aortic vessel branches and rotate one of the planes until they correctly overlap in both planes.

2D Visual Comparison: To extract 2D slices from the MRI data, multiple measuring planes can be created and moved along the vessel centerline (Figure 4.52). The temporal position of a plane can be adjusted using a slider. By default, planes are squares with a side length of 4 cm, which is sufficient in most cases to cover the entire diameter of the aorta. If the aortic diameter strongly deviates from the norm at certain points, for example due to an aneurysm, the size of the individual planes can be manually adjusted.

Each plane is defined by their center position ($\vec{v_c}$) and three directional unit vectors ($\vec{v_x}$, $\vec{v_y}$ and $\vec{v_z}$). These vectors represent the local coordinate system of the plane, with the first two vectors spanning the plane and the last denoting its normal. The work of Köhler et al. gives detailed information about the calculation of these vectors [192]. The sampled velocity vector $\vec{s}$ is visualized on the plane either by normalizing and directly mapping it onto RGB colors or by using a color scale. It is possible to toggle between the data coordinate system (where the X, Y and Z axis corresponds to the respective axes of the image grid) and the local plane coordinate system (where the coordinate system axes are represented by $\vec{v_x}$, $\vec{v_y}$ and $\vec{v_z}$). The rotated velocity vector $\vec{s'}$ is calculated using the dot product of the sampled vector with each of the local coordinate system axes. With the plane coordinate system, it is easier to distinguish laminar flow (which primarily moves in the direction of the Z axis) from non-laminar flow (showing additional motion on the X and Y axis). This is exemplified in Figure 4.53. By switching to the plane coordinate system, it becomes clear that one plane (c and d) shows laminar flow (colored mostly blue due to being aligned with the Z axis), whereas the other one (a and b) does not. The flow jet position is shown in each plane using a cross (Figure 4.53). It is calculated from the average of all pixel positions in the plane, weighted by velocity and normalized using the vessel diameter to reduce the influence of noise [225]. To better convey the shape of the flow, a color overlay highlighting the 50%, 75% and 90% quantile of the flow speed can be enabled (Figure 4.54). The opacity of the color overlay is modulated with the ratio of the highest speed in the plane and the overall highest speed of the dataset to deemphasize slower diastolic flow, which is more susceptible to noise (Figure 4.54c).

In addition to showing data sampled directly from the flow fields, we also display two feature-based 2D comparative visualizations (Figure 4.55). The first one shows differences in flow speed and is generated by subtracting the sampled flow speed of the second dataset from the first dataset (Figure 4.55a). In white areas, the flow speed is equal, whereas colored areas indicate that the respective dataset (red or blue) has faster flow at this position. The second comparative visualization shows differences in flow angles, with a color scale ranging from yellow (0° angle difference) to red (180° angle difference) (Figure 4.55b). The crosses highlighting the flow jet from both datasets are mirrored in the comparative visualizations as well. The degrees of freedom in positioning the measuring planes could easily cause details in the dataset to be overlooked, as the user would have to manually explore the entirety of the spatio-temporal domain of the dataset. Therefore, we added bull's eye plots with a spatio-temporal encoding inspired by Köhler et al. [174] and familiar in cardiology as an overview visualization.

Each point in these plots corresponds to a plane at a specific point in the spatio-temporal domain using polar coordinates. The distance from the center of the plot encodes the position on the centerline, the angle encodes the time-point. Clicking on the plot will move the current measuring plane to the selected spatial and temporal position.

There is a total number of eight bull's eye plots available (Figure 4.56). The first six encode the flow jet displacement and regurgitation fraction (Figure 4.56 a, b, e and c), which is an important clinical indicator for a heart valve disease. The first two plots of each type only show information from one of the datasets, therefore their outline is colored red and blue, respectively (Figure 4.56a and 4.56d). To calculate the flow jet displacement, the distance of the flow jet from the vessel centerline is computed for each plane and normalized with the vessel diameter. The last flow jet plot shows the distance between the flow jets in each dataset (Figure 4.56b). Similarly, the last regurgitation plot show a comparison of the regurgitation fractions of both datasets (Figure 4.56e). Two additional plots show a comparison between flow speed and average flow angle (Figure 4.56c and 4.56f). To support orientation, a circular grid with eight radial lines is overlayed on top of the plot. All bull's eye plots use discrete color scales to highlight regions with high or low values. Different types of information (e.g. flow jet, velocity, angle) are shown with different color scales to prevent confusion.

3D Visual Comparison: The 3D visualization shows the surface model and pathlines of both datasets side by side using a synchronized camera, allowing for an image-based comparison. We consider the vascular surface to be a context object, therefore we use

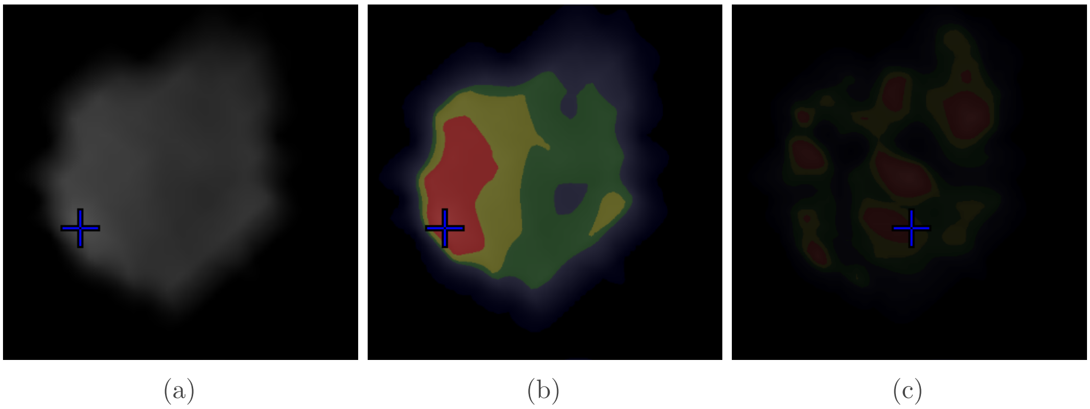


Figure 4.54: *Sampled plane with (b,c) and without (a) color overlay during the systolic (a,b) and diastolic (c) phase.*

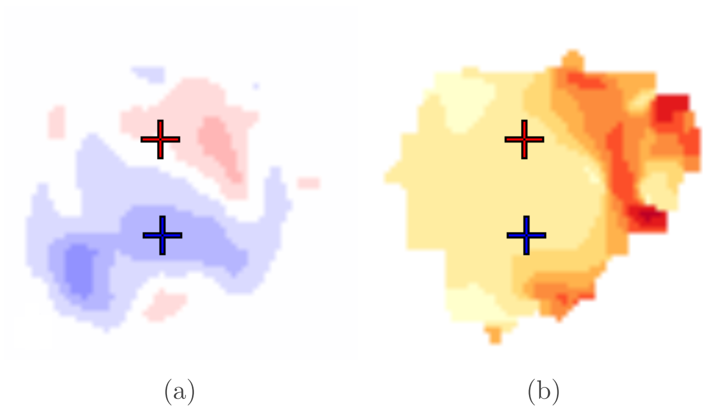


Figure 4.55: *2D comparative visualizations for flow velocity (a) and flow angle difference (b).*

the Fresnel opacity presented by Gasteiger et al. to prevent occlusion with the inlying pathlines [151]. Parameters such as speed or pressure can be mapped onto the pathline color or opacity. To support orientation in the dataset, the selected planes are also visible in the 3D view. They can be individually hidden to prevent possible occlusion.

Implementation

Sampling from the velocity field is performed off-screen using a fragment shader in combination with multiple 2D textures bound to the framebuffer. The velocity field itself is stored as a set of 3D textures. Both the current and next time step are bound as active textures together with an interpolation value. Once for each texel of the 2D output texture, the fragment shader samples values from the velocity field and interpolates them

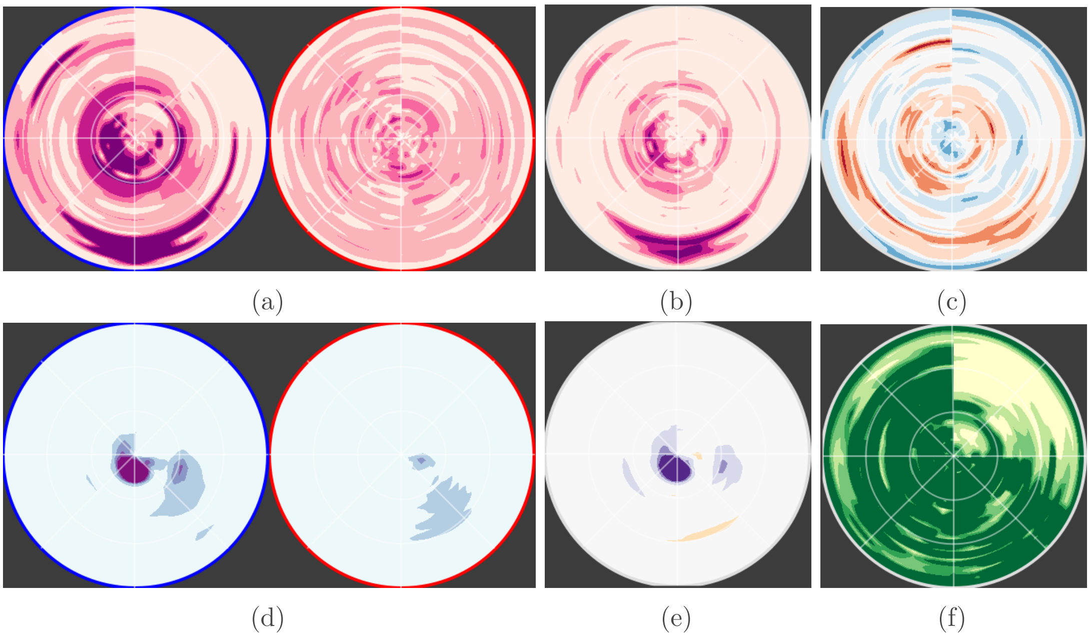


Figure 4.56: *Bull's eye plots, showing individual (a) and comparative (b) flow jet displacement, individual (d) and comparative (e) regurgitation fraction, velocity (c) and angle comparison (f).*

according to the interpolation value. Sampled data is written to three different textures. The first contains the raw flow data sampled from the flow field. The second and third textures are used for rendering and contain the data after the application of the color scale and the flow speed overlay, respectively. In both the 2D and 3D view, the user can switch between binding one of these textures, or both of them at the same time.

Once the user has finished the spatial and temporal registration, the flow jet and bull's eye plots are generated. A number of planes is automatically placed spatially along the registered centerline with a distance of 1cm, which was chosen empirically as a trade-off between plane coverage and required computation time. For each plane, a set of values is calculated and stored in a single pixel of a two-dimensional image. These values include the flow jet position as a 2D vector, the flow jet displacement in relation to the centerline and the average flow velocity in the plane. This results in a square image where each pixel represents a single plane at a single point in time, covering the entire spatio-temporal domain. To display the flow jet or generate the bull's eye plots, this image is then sampled using linear interpolation.

Results

To evaluate our method, we performed an informal interview with three expert radiologists. They were shown a total of four aortic datasets available for a pairwise comparison. Two were acquired from the same volunteer right after one another using different MRI sequences. Additionally, a dataset from a different volunteer as well as a patient dataset were used.

Overall, the radiologists found the ability to directly compare flow datasets useful. The amount and complexity of manual input required for the registration of the data was deemed acceptable. One of the physicians noted that if the DICOM data contained data about the heart rate during acquisition, this information could be used to normalize the flow velocity for both datasets in an optional pre-processing step. A point of critique was the lack of a way to restrict the spatio-temporal domain of the bull's eye plots to certain phases of the heart beat or anatomical regions that the physicians were specifically interested in. Two of the radiologists were especially interested in comparing data from the same patient, either at different time-points or using different sequences, to evaluate different MRI sequences or perform follow-up examinations. The third radiologist expressed interest in using our tool to support the extraction of standard values regarding blood flow in patients and healthy volunteers. The comparison of the two datasets acquired using different MRI sequences allowed the radiologists to detect artifacts in one of the datasets (Figure 4.57). While one sequence shows primarily laminar flow, the same anatomical region contains a vortex when acquired using a different MRI sequence, which is clearly visible in the average flow angle bull's eye plot (Figure 4.57e). During the evaluation, the radiologists noticed a problem that affected the flow jet calculation in some of the datasets. Depending on the size of the measuring planes and the overall vessel geometry, a plane placed in the descending aorta may intersect with the left ventricle. This may impact the flow jet calculation, causing the bull's eye plot to show a strong displacement that does not actually exist in the data.

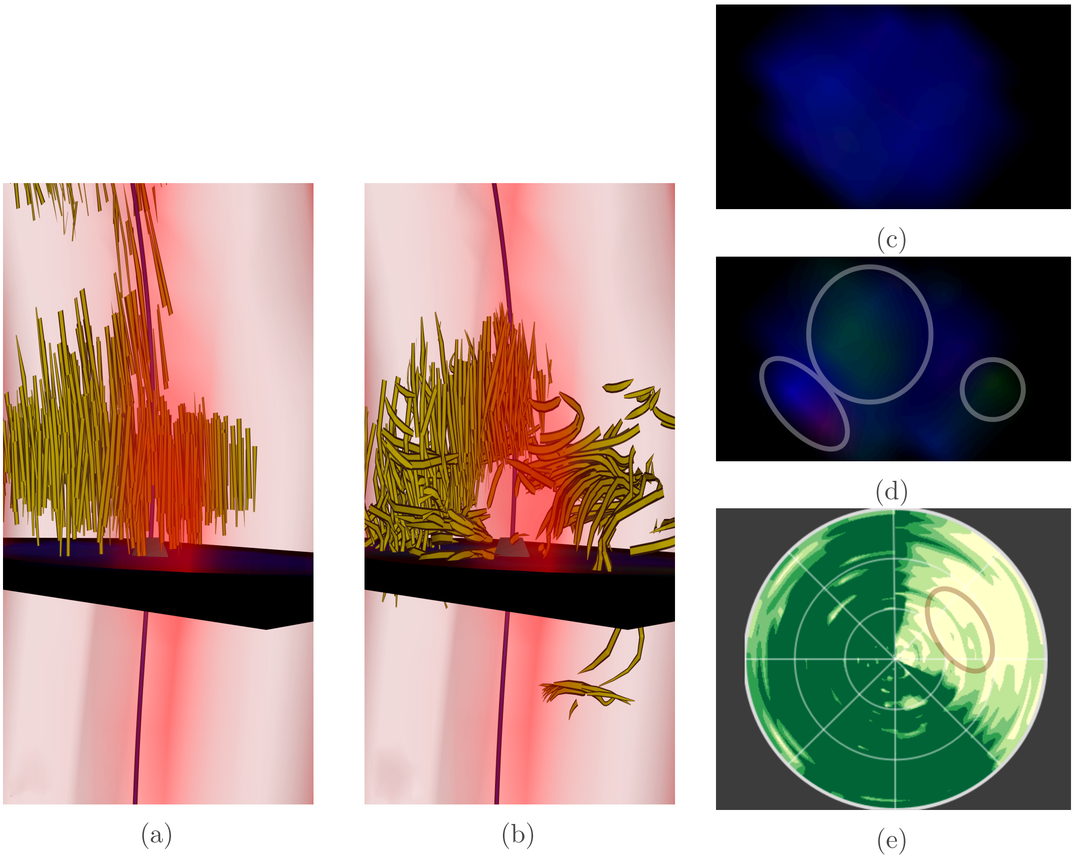


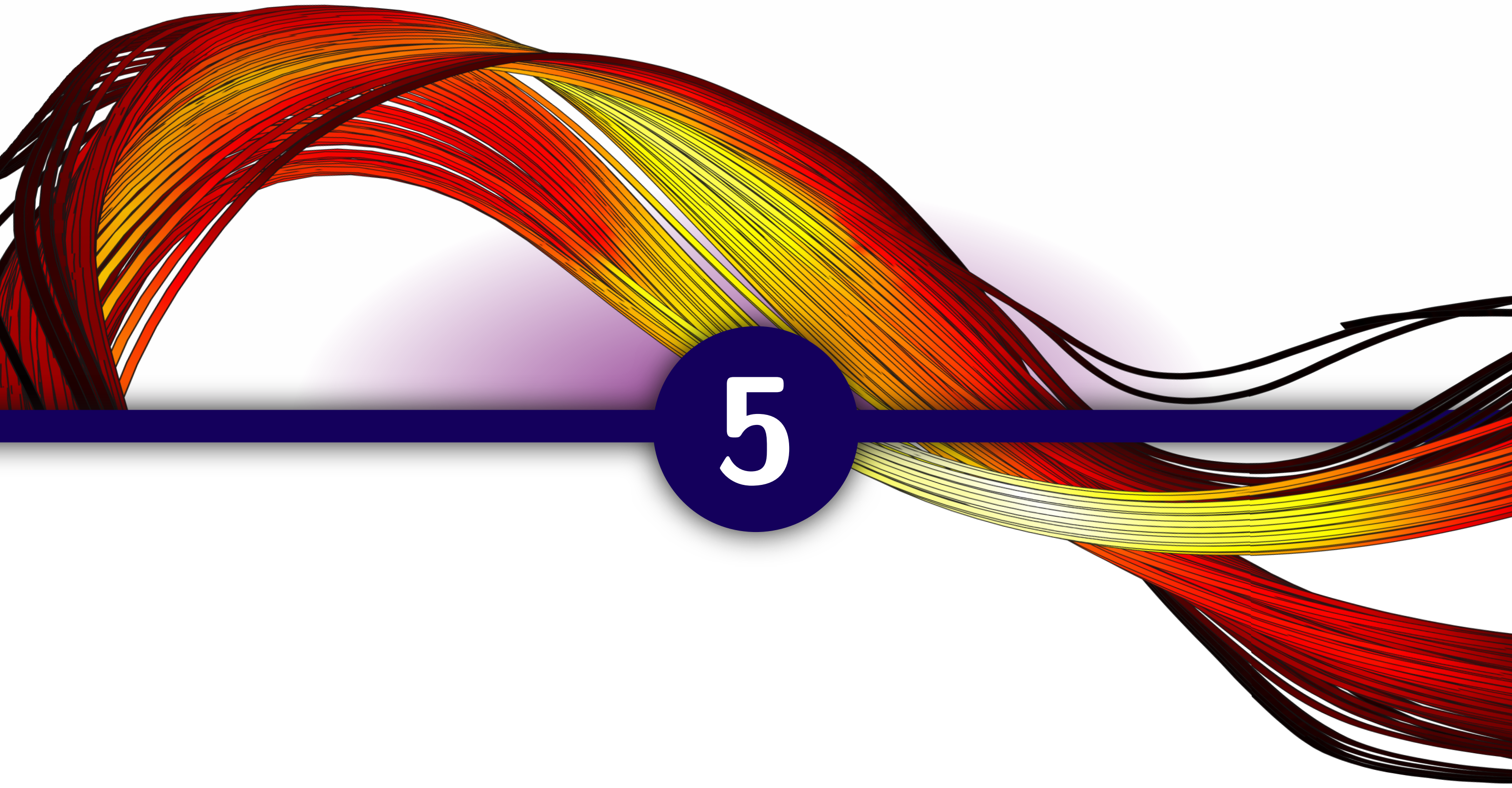
Figure 4.57: Changes in flow representation (a,b) and average flow angle bull's eye plot (e) due to deviations in the flow field (c,d) as a result of different MRI sequences.

Discussion

In this paper, we have presented a set of tools for the comparative exploration of two 4D PC-MRI datasets. An informal interview with three experienced radiologists indicated the usefulness of our approach. Although we focused on the examination of aortic blood flow, our methods should be applicable to other vessels such as the pulmonary artery with only minimal adjustments. However, the exploration of other structures, such as the left or right ventricle, would be more challenging since we rely on the presence of a centerline to place measuring planes. Therefore, a different method of placing planes and performing a spatial MRI registration would need to be implemented. Additionally, a dynamic segmentation would be needed to account for the strong ventricular movement during the cardiac phases.

Currently, our application only allows comparisons between 4D PC-MRI datasets. In the future, we plan to add support for data from computational fluid dynamics (CFD) simulations. This would enable the user to explore the differences between measured and simulated flow in the same vessel. A key problem that would need to be solved is the handling of data with vastly different resolution and storage format. It may also be challenging to sample the high-resolution CFD data in real-time, requiring additional pre-processing steps to allow for an interactive frame rate.

In addition to the visual exploration of the differences between two datasets, a tool to process a larger number of datasets would also be useful. This would support the systematic evaluation of a larger database of flow data. Such an approach requires a higher degree of automation regarding the registration as well as methods to automatically cluster the datasets based on their similarity and extract the systematic differences between the clusters.



5

Conclusion

Summary 5.1

Current medical imaging modalities allow us to obtain patient-specific blood flow information, either by means of direct measurement or hemodynamic simulation. In combination with explorative visualization techniques, this information is an important tool in the quest to gain a deeper understanding of the causes and progression of vascular pathologies and, in turn, developing better individual treatment plans for patients.

During the course of my PhD thesis, we have built a flexible integrated framework that supports both measured 4D PC-MRI data as well as simulated blood flow data. While most of the presented techniques are applicable to both measured and simulated data, some were specifically designed for 4D PC-MRI datasets. To gain a quick overview of 4D PC-MRI datasets, a volume rendering which does not require a segmentation is available. Pathlines can be integrated into this visualization by using a front-face culling algorithm adapted for use in volume rendering. Our framework offers *segmentation functionality* for 4D PC-MRI data based on the commonly used magnitude images as well as enhanced FTLE images, incorporating directional information. This eases the task of segmenting low-contrast images. Additionally, *direct comparisons* between different 4D PC-MRI datasets offer insights into systematic differences between healthy volunteers, patient groups and different image acquisition sequences.

To improve the *spatial perception* of complex vascular structures, a depth-enhancing shading based on the pseudo-chromadepth color scale that does not inhibit the mapping of hemodynamic parameters onto the vessel surface is available as an option for all visualizations based on surface meshes.

A novel *pathline seeding strategy* based on evolutionary algorithms is available to efficiently improve pathline density in medically interesting regions. By adjusting the related fitness function, pathlines with different properties, such as a hight residence time within the aneurysm or the distance to a specific anatomical region, can be targeted. The relation of near-surface flow to the vessel wall can be explored using *surface-based filtering*. By selecting surface regions either freely or based on it's hemodynamic parameters, associated flow structures can be highlighted.

In addition to the aforementioned integrated framework, we also developed a web-based application to collect information from medical professionals on their *definition of the neck curve* that separates an aneurysm from its harboring vessel. This application supports the ongoing research on reaching a consensus in the medical community on the placement of neck curves.

Future Work 5.2

The field of blood flow visualization has by no means been exhaustively researched. There is plethora of future research topics, some of which are detailed in the following.

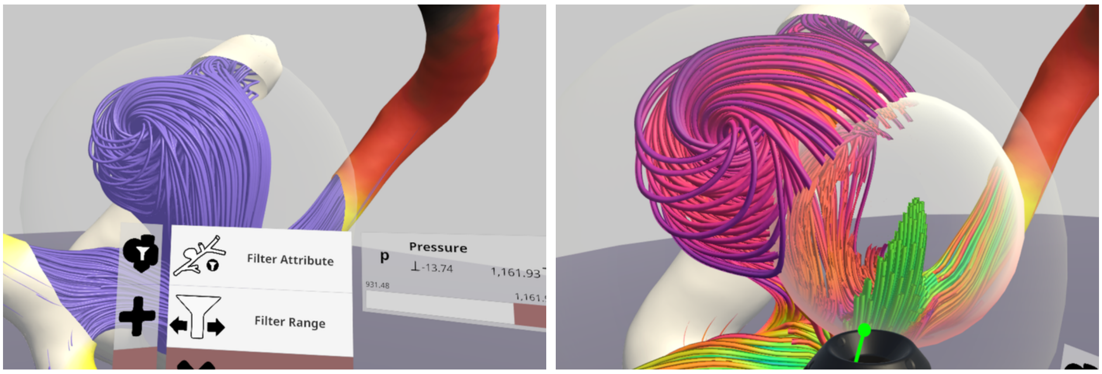
Cohort exploration and visual analytics: Due to the complex and highly variable vessel shapes, a direct visual comparison of multiple flow datasets is challenging, especially when considering patients with different pathologies. However, examining the specific differences and similarities in blood flow between different patients or healthy volunteers can potentially lead to further insights into the effects and consequences of various pathologies. Our previous work in this field only allows for the direct comparison of two datasets and is limited to 4D PC-MRI data.
Analyzing cohorts instead of singular datasets is the next logical step towards a better understanding of cardiovascular pathologies. Due to patient- and pathology-specific anatomical variations, a normalization of both vascular morphology and hemodynamic parameters is an important pre-processing step.
Visual analytics techniques allow medical professionals to gain insights into the similarities and dissimilarities within and between groups of patients and find hidden correlations between hemodynamic parameters and the severity or risk of vascular pathologies. Such techniques have been used in the past to analyze cohorts [226–228]. However, they require *guidance* as medical users are often unfamiliar with the underlying statistical methods and workflows, thus being unable to exploit their full potential [229]. In addition to exploring typical cohort definitions based on age, gender or medical history, visual analytics can also help to uncover new possible cohort definitions by identifying outliers or clusters within existing cohorts. It seems natural to also investigate the possibility of creating a system that can be queried by medical researchers in order find similarities in a database of case files in regards to a specific, new dataset.

Visualization and analysis of wall movement: Within our previous publications and in this PhD thesis, we assumed the vessel wall to be static, which is not medically accurate. Especially the aorta and pulmonary artery exhibit not only general movement over each heart beat, but also changes in diameter. With 4D PC-MRI, the vascular movement can already be captured and visualized [192]. However, the low image contrast during the diastole makes creating a time-dependent segmentation over all time steps challenging. Currently, the available imaging modalities for cerebral vessels do not capture vessel movements. The movement of the vessel wall modeled withing a CFD simulation using fluid-structure simulations (FSI), but this requires accurate information about the local wall thickness [230], which is not acquired in clinical practice. However, recent developments in imaging technologies, namely intracranial optical coherence tomography (OCT), have the potential to reveal not only local wall thickness, but also tissue composition [231]. Additionally, increased field strengths for MRI scanners have the potential to make 4D PC-MRI more feasible for intracranial vessels [232]

Immersive blood flow in VR: In my dissertation, I have focussed on visualizing blood flow data on conventional 2D screens and input methods, such as a mouse and keyboard. However, other output devices such as Virtual Reality (VR) headsets are naturally suited to display complex geometric structures. Unlike a normal monitor, VR technology is capable of exploiting the ability of spatial perception. In addition, the use of specialized input devices allow the usage of natural-feeling interaction gestures in VR that, as opposed to artificial mouse gestures, are not limited to two dimensions. With the ever-increasing computational power of affordable, consumer-grade workstations, virtual reality devices are becoming more widespread and are being used in both medical applications and anatomy education software [233]. Forsberg et al. [234] already presented a VR-based explorative approach for blood flow visualization in the year 2000. They focussed on visualizing local flow phenomena, specifically the hemodynamics in the vicinity of coronary artery grafts, by placing the user within the vessel and providing them with several tools to explore the flow in that area.

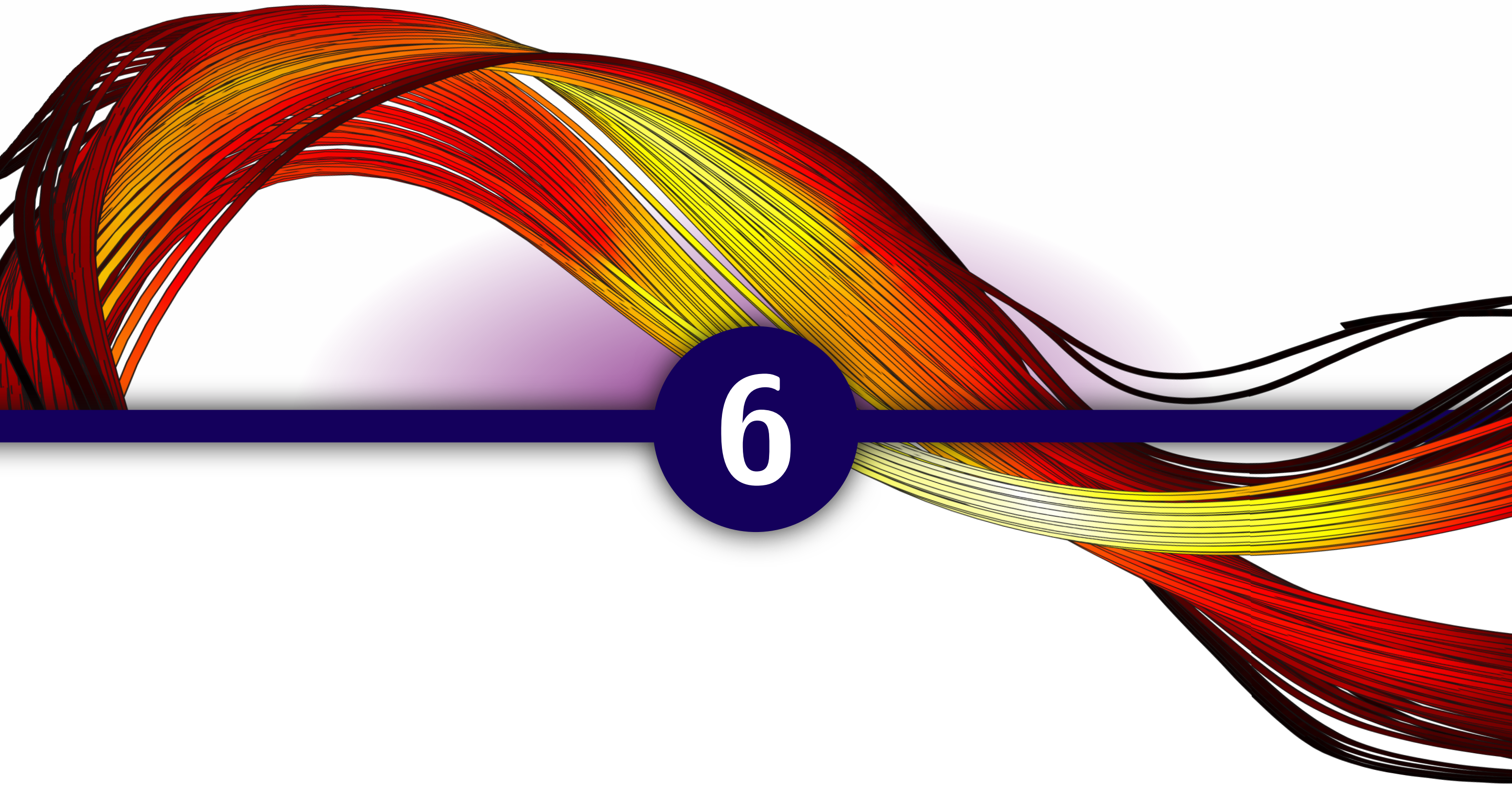
However, VR can also facilitate exploration in entire vessel trees with multiple regions of interest. The immersive visualization and natural interaction have the potential to allow for a intuitive navigation and selection of vessel regions. We have already developed a VR prototype for blood flow exploration by applying local filters [235]. The core principle takes inspiration from the *FlowLens* by Gasteiger et al. [157] and other *Magic lens*-based [236] approaches. Users are placed next to a floating model of the vessel geometry, that can be freely moved, rotated and scaled by grabbing it using pointer gestures. As mouse and keyboard are not available within the virtual environment, the primary input method are motion controllers that work together with the headset. We decided to use a ray cast-based interaction represented with a laser pointer that allows grabbing near and far objects. The user interface is mounted to the virtual representation of the motion controllers and thus can easily be moved in and out of view or brought closer to the camera. In addition to moving around, users can create filter regions ("*lenses*"), which apply customizable predicate-based filters (Figure 5.1a) to vessel regions or pathlines within them. These lenses can be freely scaled, moved and attached to the vessel model to reveal specific flow features (Figure 5.1b). To manipulate filter value ranges, we employ the touchpad of the controller to facilitate an interaction similar to the adjustment of contrast in grey-value medical images ("*windowing*"), which would traditionally be carried out with a mouse.

Our VR prototype is capable of performing the filtering task in real-time. As common use cases would not require many lenses simultaneously, the main bottleneck regarding performance is the complexity of the examined objects. We were able to show that VR Flow Lenses are a viable concept and can support the exploration of blood flow data. However, further improvements would need to be carried out to fully adapt this concept to domains such as medical treatment planning, diagnosis or education.



(a) (b)

Figure 5.1: *Prototype for an immersive VR-based blood flow exploration tool; definition (a) and placement (b) of a filter region within the vessel.*



6

Appendix

Figures

Literature

[1] S. Mendis, P. Puska, B. Norrving, et al. *Global atlas on cardiovascular disease prevention and control.* World Health Organization, 2011.

[2] E. A. Osborn and F. A. Jaffer. “Imaging atherosclerosis and risk of plaque rupture”. In: *Current atherosclerosis reports* 15.10 (2013), p. 359.

[3] M. D. Hope, T. Sedlic, and P. Dyverfeldt. “Cardiothoracic magnetic resonance flow imaging”. In: *Journal of thoracic imaging* 28.4 (2013), pp. 217–230.

[4] E. E. Calkoen, A. A. W. Roest, van der Geest, Rob J, A. de Roos, and J. J. M. Westenberg. “Cardiovascular function and flow by 4-dimensional magnetic resonance imaging techniques: new applications”. In: *Journal of thoracic imaging* 29.3 (2014), pp. 185–196.

[5] R. Mahadevia, A. J. Barker, S. Schnell, P. Entezari, P. Kansal, P. W. M. Fedak, et al. “Bicuspid aortic cusp fusion morphology alters aortic three-dimensional outflow patterns, wall shear stress, and expression of aortopathy”. In: *Circulation* 129.6 (2014), pp. 673–682.

[6] P. Entezari, S. Schnell, R. Mahadevia, C. Malaisrie, P. McCarthy, M. Mendelson, et al. “From unicuspid to quadricuspid: influence of aortic valve morphology on aortic three-dimensional hemodynamics”. In: *Journal of magnetic resonance imaging : JMRI* 40.6 (2014), pp. 1342–1346.

[7] M. D. Hope, T. A. Hope, S. E. S. Crook, K. G. Ordovas, T. H. Urbania, M. T. Alley, et al. “4D flow CMR in assessment of valve-related ascending aortic disease”. In: *JACC. Cardiovascular imaging* 4.7 (2011), pp. 781–787.

[8] L. K. Bickerstaff, P. C. Pairolero, L. H. Hollier, L. J. Melton, H. J. van Peenen, K. J. Cherry, et al. “Thoracic aortic aneurysms: a population-based study”. In: *Surgery* 92.6 (1982), pp. 1103–1108.

[9] R. Young and H. Ostertag. “Häufigkeit, Atiologie und Rupturrisiko des Aortenaneurysms. Eine Autopsiestudie”. In: *Deutsche medizinische Wochenschrift (1946)* 112.33 (1987), pp. 1253–1256.

[10] W. D. Clouse, J. W. Hallett, H. V. Schaff, M. M. Gayari, D. M. Ilstrup, and L. J. Melton. “Improved prognosis of thoracic aortic aneurysms: a population-based study”. In: *JAMA* 280.22 (1998), pp. 1926–1929.

[11] G. Johansson, U. Markström, and J. Swedenborg. “Ruptured thoracic aortic aneurysms: A study of incidence and mortality rates”. In: *Journal of Vascular Surgery* 21.6 (1995), pp. 985–988.

[12] R. R. Davies, L. J. Goldstein, M. A. Coady, S. L. Tittle, J. A. Rizzo, G. S. Kopf, et al. “Yearly rupture or dissection rates for thoracic aortic aneurysms: simple prediction based on size”. In: *The Annals of Thoracic Surgery* 73.1 (2002), pp. 17–28.

[13] N. Etminan, B. A. Buchholz, R. Dreier, P. Bruckner, J. C. Torner, H.-J. Steiger, et al. “Cerebral aneurysms: formation, progression, and developmental chronology”. In: *Translational stroke research* 5.2 (2014), pp. 167–173.

[14] J. P. Greving, M. J. H. Wermer, R. D. Brown, A. Morita, S. Juvela, M. Yonekura, et al. “Development of the PHASES score for prediction of risk of rupture of intracranial aneurysms: a pooled analysis of six prospective cohort studies”. In: *The Lancet Neurology* 13.1 (2014), pp. 59–66.

[15] T. Tsukahara, N. Murakami, Y. Sakurai, M. Yonekura, T. Takahashi, T. Inoue, et al. “Treatment of unruptured cerebral aneurysms; a multi-center study at Japanese national hospitals”. In: *New Trends of Surgery for Stroke and its Perioperative Management*. Vol. 94. Acta Neurochirurgica Supplements. Springer-Verlag, 2005, pp. 77–85.

[16] J. Beck, S. Rohde, M. el Beltagy, M. Zimmermann, J. Berkefeld, V. Seifert, et al. “Difference in configuration of ruptured and unruptured intracranial aneurysms determined by biplanar digital subtraction angiography”. In: *Acta neurochirurgica* 145.10 (2003), 861–5, discussion 865.

[17] F. J. Detmer, B. J. Chung, F. Mut, M. Slawski, F. Hamzei-Sichani, C. Putman, et al. “Development and internal validation of an aneurysm rupture probability model based on patient characteristics and aneurysm location, morphology, and hemodynamics”. In: *International journal of computer assisted radiology and surgery* 13.11 (2018), pp. 1767–1779.

[18] H. Meng, V. M. Tutino, J. Xiang, and A. Siddiqui. “High WSS or low WSS? Complex interactions of hemodynamics with intracranial aneurysm initiation, growth, and rupture: toward a unifying hypothesis”. In: *AJNR. American journal of neuroradiology* 35.7 (2014), pp. 1254–1262.

[19] J. Xiang, R. J. Damiano, N. Lin, K. V. Snyder, A. H. Siddiqui, E. I. Levy, et al. “High-fidelity virtual stenting: modeling of flow diverter deployment for hemodynamic characterization of complex intracranial aneurysms”. In: *Journal of neurosurgery* 123.4 (2015), pp. 832–840.

[20] P. Berg, S. Saalfeld, G. Janiga, O. Brina, N. M. Cancelliere, P. Machi, et al. “Virtual stenting of intracranial aneurysms: A pilot study for the prediction of treatment success based on hemodynamic simulations”. In: *The International journal of artificial organs* 41.11 (2018), pp. 698–705.

[21] S. Oeltze-Jafra, M. Meuschke, M. Neugebauer, S. Saalfeld, K. Lawonn, G. Janiga, et al. “Generation and Visual Exploration of Medical Flow Data: Survey, Research Trends and Future Challenges”. In: *Computer Graphics Forum* 40.4 (2019), p. 860.

[22] J. G. Betts, K. A. Young, J. A. Wise, E. Johnson, B. Poe, D. H. Kruse, et al. *Anatomy and Physiology*. Houston, Texas: OpenStax, 2013.

[23] R. F. Schmidt. *Physiologie des Menschen: Mit Pathophysiologie*. 31., überarb. und aktualisierte Aufl. Springer-Lehrbuch. Heidelberg: Springer, 2010.

[24] Institute for Quality and Efficiency in Health Care. *How does the blood circulatory system work?* 2010. URL: https://www.informedhealth.org/how-does-the-blood-circulatory-system-work.2097.en.html (visited on 02/26/2020).

[25] G. C. Jakanani and W. Adair. “Frequency of variations in aortic arch anatomy depicted on multidetector CT”. In: *Clinical radiology* 65.6 (2010), pp. 481–487.

[26] A. G. Osborn and J. M. Jacobs. *Diagnostic cerebral angiography*. 2. ed. Philadelphia: Lippincott-Raven, 1999.

[27] A. R. Omran. “The epidemiologic transition: a theory of the epidemiology of population change. 1971”. In: *The Milbank quarterly* 83.4 (2005), pp. 731–757.

[28] World Health Organization. *Global Health Estimates 2016: Deaths by Cause, Age, Sex, by Country and by Region, 2000-2016*. 2018. URL: https://www.who.int/healthinfo/global_burden_disease/estimates/en/ (visited on 11/25/2019).

[29] U.S. Department of Health and Human Services, National Institutes of Health, National Heart, Lung, and Blood Institute. *Atherosclerosis*. 2008. URL: https://www.nhlbi.nih.gov/health-topics/atherosclerosis (visited on 12/17/2019).

[30] V. L. Roger, A. S. Go, D. M. Lloyd-Jones, R. J. Adams, J. D. Berry, T. M. Brown, et al. "Heart disease and stroke statistics–2011 update: a report from the American Heart Association". In: *Circulation* 123.4 (2011), e18–e209.

[31] K. Withers, G. Carolan-Rees, and M. Dale. "Pipeline^{TM} embolization device for the treatment of complex intracranial aneurysms: a NICE Medical Technology Guidance". In: *Applied health economics and health policy* 11.1 (2013), pp. 5–13.

[32] M. Syed and M. Lesch. "Coronary artery aneurysm: A review". In: *Progress in Cardiovascular Diseases* 40.1 (1997), pp. 77–84.

[33] E. M. Isselbacher. "Thoracic and abdominal aortic aneurysms". In: *Circulation* 111.6 (2005), pp. 816–828.

[34] J. A. Elefteriades. "Natural history of thoracic aortic aneurysms: indications for surgery, and surgical versus nonsurgical risks". In: *The Annals of Thoracic Surgery* 74.5 (2002), S1877–S1880.

[35] F. Bonneville, N. Sourour, and A. Biondi. "Intracranial aneurysms: An overview". In: *Neuroimaging clinics of North America* 16.3 (2006), pp. 371–82, vii.

[36] A. Morita, T. Kirino, K. Hashi, N. Aoki, S. Fukuhara, N. Hashimoto, et al. "The natural course of unruptured cerebral aneurysms in a Japanese cohort". In: *The New England journal of medicine* 366.26 (2012), pp. 2474–2482.

[37] D. O. Wiebers. "Unruptured intracranial aneurysms: Natural history, clinical outcome, and risks of surgical and endovascular treatment". In: *The Lancet* 362.9378 (2003), pp. 103–110.

[38] M. J. H. Wermer, I. C. van der Schaaf, A. Algra, and G. J. E. Rinkel. "Risk of rupture of unruptured intracranial aneurysms in relation to patient and aneurysm characteristics: an updated meta-analysis". In: *Stroke* 38.4 (2007), pp. 1404–1410.

[39] R. R. Lall, C. S. Eddleman, B. R. Bendok, and H. H. Batjer. "Unruptured intracranial aneurysms and the assessment of rupture risk based on anatomical and morphological factors: sifting through the sands of data". In: *Neurosurgical focus* 26.5 (2009), E2.

[40] U. Niemann, P. Berg, A. Niemann, O. Beuing, B. Preim, M. Spiliopoulou, et al. "Rupture Status Classification of Intracranial Aneurysms Using Morphological Parameters". In: *IEEE International Symposium on Computer-Based Medical Systems (CBMS)*. 2018, pp. 48–53.

[41] B. J. Chung, R. Doddasomayajula, F. Mut, F. Detmer, M. B. Pritz, F. Hamzei-Sichani, et al. "Angioarchitectures and Hemodynamic Characteristics of Posterior Communicating Artery Aneurysms and Their Association with Rupture Status". In: *AJNR. American journal of neuroradiology* 38.11 (2017), pp. 2111–2118.

[42] R. Doddasomayajula, B. Chung, F. Hamzei-Sichani, C. M. Putman, and J. R. Cebral. "Differences in Hemodynamics and Rupture Rate of Aneurysms at the Bifurcation of the Basilar and Internal Carotid Arteries". In: *AJNR. American journal of neuroradiology* 38.3 (2017), pp. 570–576.

[43] F. J. Detmer, B. J. Chung, C. Jimenez, F. Hamzei-Sichani, D. Kallmes, C. Putman, et al. "Associations of hemodynamics, morphology, and patient characteristics with aneurysm rupture stratified by aneurysm location". In: *Neuroradiology* 61.3 (2019), pp. 275–284.

[44] J. Xiang, V. M. Tutino, K. V. Snyder, and H. Meng. "CFD: computational fluid dynamics or confounding factor dissemination? The role of hemodynamics in intracranial aneurysm rupture risk assessment". In: *AJNR. American journal of neuroradiology* 35.10 (2014), pp. 1849–1857.

[45] P. S. Martin, B. Kloesel, R. A. Norris, M. Lindsay, D. Milan, and S. C. Body. “Embryonic Development of the Bicuspid Aortic Valve”. In: *Journal of cardiovascular development and disease* 2.4 (2015), pp. 248–272.

[46] L. J. Olson, R. Subramanian, and W. D. Edwards. “Surgical Pathology of Pure Aortic Insufficiency: A Study of 225 Gases”. In: *Mayo Clinic Proceedings* 59.11-12 (1984), pp. 835–841.

[47] R. O. Bonow, B. Carabello, A. C. de Leon, L. H. Edmunds, B. J. Fedderly, M. D. Freed, et al. “Guidelines for the management of patients with valvular heart disease: executive summary. A report of the American College of Cardiology/American Heart Association Task Force on Practice Guidelines (Committee on Management of Patients with Valvular Heart Disease)”. In: *Circulation* 98.18 (1998), pp. 1949–1984.

[48] J. Soler-Soler and E. Galve. “Worldwide perspective of valve disease”. In: *Heart* 83.6 (2000), pp. 721–725.

[49] V. T. Nkomo, J. M. Gardin, T. N. Skelton, J. S. Gottdiener, C. G. Scott, and M. Enriquez-Sarano. “Burden of valvular heart diseases: a population-based study”. In: *The Lancet* 368.9540 (2006), pp. 1005–1011.

[50] D. Horstkotte and F. Loogen. “The natural history of aortic valve stenosis”. In: *European heart journal* 9 Suppl E (1988), pp. 57–64.

[51] C. Ward. “Clinical significance of the bicuspid aortic valve”. In: *Heart* 83.1 (2000), pp. 81–85.

[52] S. C. Siu and C. K. Silversides. “Bicuspid aortic valve disease”. In: *Journal of the American College of Cardiology* 55.25 (2010), pp. 2789–2800.

[53] H.-H. Sievers and C. Schmidtke. “A classification system for the bicuspid aortic valve from 304 surgical specimens”. In: *The Journal of thoracic and cardiovascular surgery* 133.5 (2007), pp. 1226–1233.

[54] T. M. Tadros, M. D. Klein, and O. M. Shapira. “Ascending aortic dilatation associated with bicuspid aortic valve: pathophysiology, molecular biology, and clinical implications”. In: *Circulation* 119.6 (2009), pp. 880–890.

[55] F. Robicsek, M. J. Thubrikar, J. W. Cook, and B. Fowler. “The congenitally bicuspid aortic valve: how does it function? Why does it fail?” In: *The Annals of Thoracic Surgery* 77.1 (2004), pp. 177–185.

[56] P. A. McKee, W. P. Castelli, P. M. McNamara, and W. B. Kannel. “The natural history of congestive heart failure: the Framingham study”. In: *The New England journal of medicine* 285.26 (1971), pp. 1441–1446.

[57] D. D. Schocken, M. I. Arrieta, P. E. Leaverton, and E. A. Ross. “Prevalence and mortality rate of congestive heart failure in the United States”. In: *Journal of the American College of Cardiology* 20.2 (1992), pp. 301–306.

[58] F. Bailliard and R. H. Anderson. “Tetralogy of Fallot”. In: *Orphanet journal of rare diseases* 4 (2009), p. 2.

[59] P. Dyverfeldt, J. E. Kvitting, A. Sigfridsson, J. Engvall, A. F. Bolger, and T. Ebbers. “Assessment of fluctuating velocities in disturbed cardiovascular blood flow: in vivo feasibility of generalized phase-contrast MRI”. In: *Journal of magnetic resonance imaging : JMRI* 28.3 (2008), pp. 655–663.

[60] A. J. Barker, M. Markl, J. Bürk, R. Lorenz, J. Bock, S. Bauer, et al. “Bicuspid aortic valve is associated with altered wall shear stress in the ascending aorta”. In: *Circulation. Cardiovascular imaging* 5.4 (2012), pp. 457–466.

[61] R. L. F. van der Palen, A. J. Barker, E. Bollache, J. Garcia, M. J. Rose, P. van Ooij, et al. “Altered aortic 3D hemodynamics and geometry in pediatric Marfan syndrome patients”. In: *Journal of Cardiovascular Magnetic Resonance* 19.1 (2017), A164.

[62] M. T. Vlaardingerbroek and J. A. den Boer. *Magnetresonanzbildgebung*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.

[63] P. R. Moran. “A flow velocity zeugmatographic interlace for NMR imaging in humans”. In: *Magnetic resonance imaging* 1.4 (1982), pp. 197–203.

[64] H. Y. Carr and E. M. Purcell. “Effects of Diffusion on Free Precession in Nuclear Magnetic Resonance Experiments”. In: *Physical Review* 94.3 (1954), pp. 630–638.

[65] N. J. Pelc, R. J. Herfkens, A. Shimakawa, D. R. Enzmann, et al. “Phase contrast cine magnetic resonance imaging”. In: *Magnetic resonance quarterly* 7.4 (1991), pp. 229–254.

[66] Z. Stankovic, B. D. Allen, J. Garcia, K. B. Jarvis, and M. Markl. “4D flow imaging with MRI”. In: *Cardiovascular diagnosis and therapy* 4.2 (2014), pp. 173–192.

[67] P. Chai and R. Mohiaddin. “How We Perform Cardiovascular Magnetic Resonance Flow Assessment Using Phase-Contrast Velocity Mapping”. In: *Journal of Cardiovascular Magnetic Resonance* 7.4 (2005), pp. 705–716.

[68] M. Markl, F. P. Chan, M. T. Alley, K. L. Wedding, M. T. Draney, C. J. Elkins, et al. “Time-resolved three-dimensional phase-contrast MRI”. In: *Journal of magnetic resonance imaging : JMRI* 17.4 (2003), pp. 499–506.

[69] B. Köhler, R. Gasteiger, U. Preim, H. Theisel, M. Gutberlet, and B. Preim. “Semi-automatic vortex extraction in 4D PC-MRI cardiac blood flow data using line predicates”. In: *IEEE transactions on visualization and computer graphics* 19.12 (2013), pp. 2773–2782.

[70] E. J. Nett, K. M. Johnson, A. Frydrychowicz, A. M. Del Rio, E. Schrauben, C. J. Francois, et al. “Four-dimensional phase contrast MRI with accelerated dual velocity encoding”. In: *Journal of magnetic resonance imaging: JMRI* 35.6 (2012), pp. 1462–1471.

[71] S. Schnell, S. A. Ansari, C. Wu, J. Garcia, I. G. Murphy, O. A. Rahman, et al. “Accelerated dual-venc 4D flow MRI for neurovascular applications”. In: *Journal of magnetic resonance imaging : JMRI* 46.1 (2017), pp. 102–114.

[72] B. Köhler, S. Born, Pelt, Roy F. P. van, U. Preim, and B. Preim. “A Survey of Cardiac 4D PC-MRI Data Processing”. In: *Eurographics Workshop on Visual Computing for Biology and Medicine*. The Eurographics Association, 2015.

[73] C. Karmonik, J. Bismuth, M. G. Davies, and A. B. Lumsden. “Computational fluid dynamics as a tool for visualizing hemodynamic flow patterns”. In: *Methodist Debakey Cardiovasc J* 5.3 (2009), pp. 26–33.

[74] H. E. Westerlaan, J. M. C. van Dijk, M. J. van Dijk, M. C. Jansen-van der Weide, J. C. de Groot, R. J. M. Groen, et al. “Intracranial aneurysms in patients with subarachnoid hemorrhage: CT angiography as a primary examination tool for diagnosis–systematic review and meta-analysis”. In: *Radiology* 258.1 (2011), pp. 134–145.

[75] L. Hacein-Bey and J. M. Provenzale. “Current imaging assessment and treatment of intracranial aneurysms”. In: *AJR. American journal of roentgenology* 196.1 (2011), pp. 32–44.

[76] S. Glaßer, P. Berg, M. Neugebauer, and B. Preim. “Reconstruction of 3D Surface Meshes for Blood Flow Simulations of Intracranial Aneurysms”. In: *Proceedings of Computer- and Robot-Assisted Surgery (CURAC)*. 2015, pp. 163–168.

[77] J. L. Brisman, J. K. Song, and D. W. Newell. “Cerebral aneurysms”. In: *The New England journal of medicine* 355.9 (2006), pp. 928–939.

[78] N. E. Green, S.-Y. J. Chen, J. C. Messenger, B. M. Groves, and J. D. Carroll. “Three-dimensional vascular angiography”. In: *Current problems in cardiology* 29.3 (2004), pp. 104–142.

[79] W. E. Lorensen and H. E. Cline. “Marching cubes: A high resolution 3D surface construction algorithm”. In: *ACM SIGGRAPH Computer Graphics* 21.4 (1987), pp. 163–169.

[80] L. Antiga, M. Piccinelli, L. Botti, B. Ene-Iordache, A. Remuzzi, and D. A. Steinman. “An image-based modeling framework for patient-specific computational hemodynamics”. In: *Medical & biological engineering & computing* 46.11 (2008), pp. 1097–1112.

[81] P. Berg, D. Stucht, G. Janiga, O. Beuing, O. Speck, and D. Thévenin. “Cerebral blood flow in a healthy Circle of Willis and two intracranial aneurysms: Computational fluid dynamics versus four-dimensional phase-contrast magnetic resonance imaging”. In: *Journal of biomechanical engineering* 136.4 (2014).

[82] J. R. Cebral, M. A. Castro, S. Appanaboyina, C. M. Putman, D. Millan, and A. F. Frangi. “Efficient pipeline for image-based patient-specific analysis of cerebral aneurysm hemodynamics: technique and sensitivity”. In: *IEEE transactions on medical imaging* 24.4 (2005), pp. 457–467.

[83] H. Hauser, R. S. Laramee, and H. Doleisch. “State-of-the-art report 2002 in flow visualization”. In: *VRVis Research Center, Vienna, Tech. Rep. TR-VRVis-2002-003* (2002).

[84] D. Weiskopf and G. Erlebacher. “Overview of Flow Visualization”. In: *Visualization Handbook*. Elsevier, 2005, pp. 261–278.

[85] B. Cabral and L. C. Leedom. “Imaging vector fields using line integral convolution”. In: *Proc. of ACM SIGGRAPH*. ACM Press, 1993, pp. 263–270.

[86] P. P. Sengupta, G. Pedrizzetti, P. J. Kilner, A. Kheradvar, T. Ebbers, G. Tonti, et al. “Emerging trends in CV flow visualization”. In: *JACC. Cardiovascular imaging* 5.3 (2012), pp. 305–316.

[87] T. McLoughlin, R. S. Laramee, R. Peikert, F. H. Post, and M. Chen. “Over Two Decades of Integration-Based, Geometric Flow Visualization”. In: *Computer Graphics Forum* 29.6 (2010), pp. 1807–1829.

[88] M. Edmunds, R. S. Laramee, G. Chen, N. Max, E. Zhang, and C. Ware. “Surface-based flow visualization”. In: *Computers & Graphics* 36.8 (2012), pp. 974–990.

[89] M. Hummel, C. Garth, B. Hamann, H. Hagen, and K. I. Joy. “IRIS: illustrative rendering for integral surfaces”. In: *IEEE transactions on visualization and computer graphics* 16.6 (2010), pp. 1319–1328.

[90] J. L. Helman and L. Hesselink. “Surface representations of two- and three-dimensional fluid flow topology”. In: *Proceedings of the First IEEE Conference on Visualization: Visualization ‘90*. IEEE Comput. Soc. Press, 1990, pp. 6–13.

[91] H. Theisel, T. Weinkauf, H.-C. Hege, and H.-P. Seidel. “Saddle connectors - an approach to visualizing the topological skeleton of complex 3D vector fields”. In: *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*. IEEE, 2003, pp. 225–232.

[92] J. Jeong and F. Hussain. “On the identification of a vortex”. In: *Journal of Fluid Mechanics* 285.-1 (1995), p. 69.

[93] H. Theisel, J. Sahner, T. Weinkauf, H.-C. Hege, and H.-P. Seidel. “Extraction of Parallel Vector Surfaces in 3D Time-Dependent Fields and Application to Vortex Core Line Tracking”. In: *VIS 05. IEEE Visualization, 2005*. IEEE, 2005, pp. 631–638.

[94] A. van Gelder. “Vortex core detection: back to basics”. In: *Visualization and Data Analysis 2012*. Ed. by P. C. Wong, D. L. Kao, M. C. Hao, C. Chen, R. Kosara, M. A. Livingston, et al. SPIE Proceedings. SPIE, 2012, p. 829413.

[95] V. Verma, D. Kao, and A. Pang. “A flow-guided streamline seeding strategy”. In: *Proceedings Visualization 2000. VIS 2000 (Cat. No.00CH37145)*. IEEE, 2000, pp. 163–170.

[96] B. Köhler, U. Preim, M. Gutberlet, K. Fischbach, and B. Preim. “Robust Cardiac Function Assessment in 4D PC-MRI Data”. In: *VCBM*. 2014, pp. 1–9.

[97] J. M. Tyszka, D. H. Laidlaw, J. W. Asa, and J. M. Silverman. “Three-dimensional, time-resolved (4D) relative pressure mapping using magnetic resonance imaging”. In: *Journal of magnetic resonance imaging : JMRI* 12.2 (2000), pp. 321–329.

[98] J. R. Cebral, F. Mut, J. Weir, and C. M. Putman. “Association of hemodynamic characteristics and cerebral aneurysm rupture”. In: *AJNR. American journal of neuroradiology* 32.2 (2011), pp. 264–270.

[99] B. Köhler, M. Grothoff, M. Gutberlet, and B. Preim. “Bloodline: A system for the guided analysis of cardiac 4D PC-MRI data”. In: *Computers & Graphics* 82 (2019), pp. 32–43.

[100] M. H. Buonocore and H. G. Bogren. “Analysis of flow patterns using MRI”. In: *International journal of cardiac imaging* 15.2 (1999), pp. 99–103.

[101] S. Meckel, A. F. Stalder, F. Santini, E.-W. Radü, D. A. Rüfenacht, M. Markl, et al. “In vivo visualization and analysis of 3-D hemodynamics in cerebral aneurysms with flow-sensitized 4-D MR imaging at 3 T”. In: *Neuroradiology* 50.6 (2008), pp. 473–484.

[102] T. A. Hope, M. D. Hope, D. D. Purcell, C. von Morze, D. B. Vigneron, M. T. Alley, et al. “Evaluation of intracranial stenoses and aneurysms with accelerated 4D flow”. In: *Magnetic resonance imaging* 28.1 (2010), pp. 41–46.

[103] A. Frydrychowicz, M. Markl, D. Hirtler, A. Harloff, C. Schlensak, J. Geiger, et al. “Aortic hemodynamics in patients with and without repair of aortic coarctation: in vivo analysis by 4D flow-sensitive magnetic resonance imaging”. In: *Investigative radiology* 46.5 (2011), pp. 317–325.

[104] J. F. Rodríguez-Palomares, L. Dux-Santoy, A. Guala, R. Kale, G. Maldonado, G. Teixidó-Turà, et al. “Aortic flow patterns and wall shear stress maps by 4D-flow cardiovascular magnetic resonance in the assessment of aortic dilatation in bicuspid aortic valve disease”. In: *Journal of Cardiovascular Magnetic Resonance* 20.1 (2018), p. 28.

[105] T. Nakayama, S.-i. Sugiyama, and M. Ohta. “Classification of Blood Flow in Cerebral Aneurysm Considering the Parent Artery Curves”. In: *Volume 3A: Biomedical and Biotechnology Engineering*. American Society of Mechanical Engineers, 2013.

[106] J. R. Cebral, M. A. Castro, J. E. Burgess, R. S. Pergolizzi, M. J. Sheridan, and C. M. Putman. “Characterization of Cerebral Aneurysms for Assessing Risk of Rupture By Using Patient-Specific Computational Hemodynamics Models”. In: *American Journal of Neuroradiology* 26.10 (2005), pp. 2550–2559.

[107] R. Gasteiger, D. J. Lehmann, R. van Pelt, G. Janiga, O. Beuing, A. Vilanova, et al. “Automatic Detection and Visualization of Qualitative Hemodynamic Characteristics in Cerebral Aneurysms”. In: *IEEE transactions on visualization and computer graphics* 18.12 (2012), pp. 2178–2187.

[108] A. Elnakib, G. Gimel’farb, J. S. Suri, and A. El-Baz. “Medical Image Segmentation: A Brief Survey”. In: *Multi Modality State-of-the-Art Medical Image Segmentation and Registration Methodologies.* Vol. 2. Springer New York, 2011, pp. 1–39.

[109] D. Lesage, E. D. Angelini, I. Bloch, and G. Funka-Lea. “A review of 3D vessel lumen segmentation techniques: models, features and extraction schemes”. In: *Medical image analysis* 13.6 (2009), pp. 819–845.

[110] R. van Pelt, J. O. Bescós, M. Breeuwer, R. E. Clough, M. E. Gröller, B. ter Haar Romenij, et al. “Exploration of 4D MRI Blood Flow using Stylistic Visualization”. In: *IEEE transactions on visualization and computer graphics* 16.6 (2010), pp. 1339–1347.

[111] A. C. S. Chung, J. A. Noble, and P. Summers. “Vascular segmentation of phase contrast magnetic resonance angiograms based on statistical mixture modeling and local phase coherence”. In: *IEEE transactions on medical imaging* 23.12 (2004), pp. 1490–1507.

[112] A. Hennemuth, O. Friman, C. Schumann, J. Bock, J. Drexl, M. Huellebrand, et al. “Fast interactive exploration of 4D MRI flow data”. In: *SPIE Medical Imaging.* SPIE Proceedings. SPIE, 2011, 79640E.

[113] J. E. Solem, M. Persson, and A. Heyden. “Velocity Based Segmentation in Phase Contrast MRI Images”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI.* Vol. 3216. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2004, pp. 459–466.

[114] G. Haller. “Lagrangian coherent structures from approximate velocity data”. In: *Physics of Fluids* 14.6 (2002), p. 1851.

[115] S. C. Shadden, F. Lekien, and J. E. Marsden. “Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows”. In: *Physica D: Nonlinear Phenomena* 212.3-4 (2005), pp. 271–304.

[116] S. C. Shadden, J. O. Dabiri, and J. E. Marsden. “Lagrangian analysis of fluid transport in empirical vortex ring flows”. In: *Physics of Fluids* 18.4 (2006), p. 047105.

[117] H. Krishnan, C. Garth, J. Guhring, M. A. Gulsun, A. Greiser, and K. I. Joy. “Analysis of time-dependent flow-sensitive PC-MRI data”. In: *IEEE transactions on visualization and computer graphics* 18.6 (2012), pp. 966–977.

[118] R. van Pelt, J. O. Bescós, M. Breeuwer, R. E. Clough, M. E. Gröller, B. t. H. Romenij, et al. “Interactive virtual probing of 4D MRI blood-flow”. In: *IEEE transactions on visualization and computer graphics* 17.12 (2011), pp. 2153–2162.

[119] M. A. Gülsün and H. Tek. “Robust vessel tree modeling”. In: *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention* 11.Pt 1 (2008), pp. 602–611.

[120] M. A. Gülsün and H. Tek. “Segmentation of carotid arteries by graph-cuts using centerline models”. In: *Medical Imaging 2010: Visualization, Image-Guided Procedures, and Modeling*. SPIE Proceedings. SPIE, 2010, p. 762530.

[121] R. van Pelt, H. Nguyen, B. ter Haar Romeny, and A. Vilanova. “Automated segmentation of blood-flow regions in large thoracic arteries using 3D-cine PC-MRI measurements”. In: *International journal of computer assisted radiology and surgery* 7.2 (2012), pp. 217–224.

[122] P. Volonghi, D. Tresoldi, M. Cadioli, A. M. Usuelli, R. Ponzini, U. Morbiducci, et al. “Automatic extraction of three-dimensional thoracic aorta geometric model from phase contrast MRI for morphometric and hemodynamic characterization”. In: *Magnetic resonance in medicine* 75.2 (2016), pp. 873–882.

[123] M. Bustamante, S. Petersson, J. Eriksson, U. Alehagen, P. Dyverfeldt, C.-J. Carlhäll, et al. “Atlas-based analysis of 4D flow CMR: automated vessel segmentation and flow quantification”. In: *Journal of Cardiovascular Magnetic Resonance* 17 (2015), p. 87.

[124] R. V. Bergen, H.-Y. Lin, M. E. Alexander, and C. P. Bidinosti. “4D MR phase and magnitude segmentations with GPU parallel computing”. In: *Magnetic resonance imaging* 33.1 (2015), pp. 134–145.

[125] K. Lawonn, M. Luz, B. Preim, and C. Hansen. “Illustrative Visualization of Vascular Models for Static 2D Representations”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI*. 2015, pp. 399–406.

[126] K. Lawonn, M. Luz, and C. Hansen. “Improving spatial perception of vascular models using supporting anchors and illustrative visualization”. In: *Computers & Graphics* 63 (2017), pp. 37–49.

[127] N. Lichtenberg, C. Hansen, and K. Lawonn. “Concentric Circle Glyphs for Enhanced Depth-Judgment in Vascular Models”. In: *Eurographics Workshop on Visual Computing for Biology and Medicine*. The Eurographics Association, 2017.

[128] D. Bartz, D. W. Cunningham, J. Fischer, and C. Wallraven. “The Role of Perception for Computer Graphics”. In: *Eurographics (STARs)*. 2008, pp. 59–80.

[129] B. Preim, A. Baer, D. Cunningham, T. Isenberg, and T. Ropinski. “A survey of perceptually motivated 3d visualization of medical image data”. In: *Computer Graphics Forum*. Vol. 35. 2016, pp. 501–525.

[130] F. Ritter, C. Hansen, V. Dicken, O. Konrad, B. Preim, and H.-O. Peitgen. “Real-time illustration of vascular structures”. In: *IEEE transactions on visualization and computer graphics* 12.5 (2006), pp. 877–884.

[131] C. Hansen, J. Wieferich, F. Ritter, C. Rieder, and H.-O. Peitgen. “Illustrative visualization of 3D planning models for augmented reality in liver surgery”. In: *International journal of computer assisted radiology and surgery* 5.2 (2010), pp. 133–141.

[132] P. Rheingans and D. Ebert. “Volume illustration: Nonphotorealistic rendering of volume models”. In: *IEEE transactions on visualization and computer graphics* 7.3 (2001), pp. 253–264.

[133] A. Joshi, X. Qian, D. P. Dione, K. R. Bulsara, C. K. Breuer, A. J. Sinusas, et al. “Effective visualization of complex vascular structures using a non-parametric vessel detection method”. In: *IEEE transactions on visualization and computer graphics* 14.6 (2008), pp. 1603–1610.

[134] T. Ropinski, F. Steinicke, and K. Hinrichs. “Visually Supporting Depth Perception in Angiography Imaging”. In: *Smart Graphics*. Vol. 4073. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, pp. 93–104.

[135] A. Grosset, M. Schott, G.-P. Bonneau, and C. D. Hansen. “Evaluation of Depth of Field for depth perception in DVR”. In: *IEEE Pacific Visualization Symposium (PacificVis)*. 2013, pp. 81–88.

[136] R. A. Steenblik. “The Chromostereoscopic Process: A Novel Single Image Stereoscopic Process”. In: SPIE Proceedings. 1987, pp. 27–34.

[137] M. Bailey and D. Clark. “Using ChromaDepth to Obtain Inexpensive Single-image Stereovision for Scientific Visualization”. In: *Journal of Graphics Tools* 3.3 (1998), pp. 1–9.

[138] M. Kersten-Oertel, S. J.-S. Chen, and D. L. Collins. “An evaluation of depth enhancing perceptual cues for vascular volume visualization in neurosurgery”. In: *IEEE transactions on visualization and computer graphics* 20.3 (2014), pp. 391–403.

[139] J. Kreiser, P. Hermosilla, and T. Ropinski. “Void Space Surfaces to Convey Depth in Vessel Visualizations”. In: *IEEE transactions on visualization and computer graphics* PP (2020).

[140] T.-Y. Lee, O. Mishchenko, H.-W. Shen, and R. Crawfis. “View point evaluation and streamline filtering for flow visualization”. In: *IEEE Pacific Visualization Symposium*. 2011, pp. 83–90.

[141] T. Salzbrunn and G. Scheuermann. “Streamline predicates”. In: *IEEE transactions on visualization and computer graphics* 12.6 (2006), pp. 1601–1612.

[142] S. Born, M. Pfeifle, M. Markl, M. Gutberlet, and G. Scheuermann. “Visual analysis of cardiac 4D MRI blood flow using line predicates”. In: *IEEE transactions on visualization and computer graphics* 19.6 (2013), pp. 900–912.

[143] W. Engelke and I. Hotz. “Evolutionary Lines for Flow Visualization”. In: *EuroVis - Short Papers*. 2018, pp. 7–11.

[144] T. Günther, C. Rössl, and H. Theisel. “Opacity optimization for 3D line fields”. In: *ACM Transactions on Graphics* 32.4 (2013), p. 1.

[145] T. Günther, C. Rössl, and H. Theisel. “Hierarchical opacity optimization for sets of 3D line fields”. In: *Computer Graphics Forum* 33.2 (2014), pp. 507–516.

[146] N. de Hoon, K. Lawonn, A. Jalba, E. Eisemann, and A. Vilanova. “InkVis: A High-Particle-Count Approach for Visualization of Phase-Contrast Magnetic Resonance Imaging Data”. In: *Eurographics Workshop on Visual Computing for Biology and Medicine*. 2019, pp. 177–188.

[147] A. J. M. Broos, Hoon, Niels H. L. C. de, Koning, Patrick J. H. de, Geest, Rob J. van der, A. Vilanova, and A. C. Jalba. “A Framework for Fast Initial Exploration of PC-MRI Cardiac Flow”. In: *Proceedings of the Eurographics Workshop on Visual Computing for Biology and Medicine* (2016), pp. 69–78.

[148] J. M. Esturo, M. Schulze, C. Rössl, and H. Theisel. “Global Selection of Stream Surfaces”. In: *Computer Graphics Forum* 32.2pt1 (2013), pp. 113–122.

[149] S. S. Barakat and X. Tricoche. “Adaptive refinement of the flow map using sparse samples”. In: *IEEE transactions on visualization and computer graphics* 19.12 (2013), pp. 2753–2762.

[150] I. Viola, A. Kanitsar, and M. E. Groller. “Importance-driven volume rendering”. In: *IEEE Visualization*. 2004, pp. 139–145.

[151] R. Gasteiger, M. Neugebauer, C. Kubisch, and B. Preim. “Adapted Surface Visualization of Cerebral Aneurysms with Embedded Blood Flow Information”. In: *VCBM*. 2010, pp. 25–32.

[152] A. Baer, R. Gasteiger, D. Cunningham, and B. Preim. “Perceptual evaluation of ghosted view techniques for the exploration of vascular structures and embedded flow”. In: *Computer Graphics Forum*. Vol. 30. 2011, pp. 811–820.

[153] S. Glaßer, P. Saalfeld, P. Berg, N. Merten, and B. Preim. “How to Evaluate Medical Visualizations on the Example of 3D Aneurysm Surfaces”. In: *Eurographics Workshop on Visual Computing for Biology and Medicine*. The Eurographics Association, 2016.

[154] K. Lawonn, R. Gasteiger, and B. Preim. “Adaptive Surface Visualization of Vessels with Embedded Blood Flow Based on the Suggestive Contour Measure”. In: *VMV - Vision, Modeling, Visualization*. 2013, pp. 113–120.

[155] K. Lawonn, R. Gasteiger, and B. Preim. “Adaptive Surface Visualization of Vessels with Animated Blood Flow”. In: *Computer Graphics Forum* 33.8 (2014), pp. 16–27.

[156] D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella. “Suggestive contours for conveying shape”. In: *Proc. of ACM SIGGRAPH*. New York, New York, USA: ACM Press, 2003, p. 848.

[157] R. Gasteiger, M. Neugebauer, O. Beuing, and B. Preim. “The FLOWLENS: A focus-and-context visualization approach for exploration of blood flow in cerebral aneurysms”. In: *IEEE transactions on visualization and computer graphics* 17.12 (2011), pp. 2183–2192.

[158] K. Lawonn, S. Glasser, A. Vilanova, B. Preim, and T. Isenberg. “Occlusion-free Blood Flow Animation with Wall Thickness Visualization”. In: *IEEE transactions on visualization and computer graphics* 22.1 (2016), pp. 728–737.

[159] M. Meuschke, S. Voss, O. Beuing, B. Preim, and K. Lawonn. “Combined Visualization of Vessel Deformation and Hemodynamics in Cerebral Aneurysms”. In: *IEEE transactions on visualization and computer graphics* 23.1 (2017), pp. 761–770.

[160] M. Neugebauer, K. Lawonn, O. Beuing, P. Berg, G. Janiga, and B. Preim. “AmniVis - A System for Qualitative Exploration of Near-Wall Hemodynamics in Cerebral Aneurysms”. In: *Computer Graphics Forum* 32.3pt3 (2013), pp. 251–260.

[161] M. Meuschke, S. Oeltze-Jafra, O. Beuing, B. Preim, and K. Lawonn. “Classification of Blood Flow Patterns in Cerebral Aneurysms”. In: *IEEE transactions on visualization and computer graphics* 25.7 (2019), pp. 2404–2418.

[162] S. Oeltze, D. J. Lehmann, A. Kuhn, G. Janiga, H. Theisel, and B. Preim. “Blood Flow Clustering and Applications in Virtual Stenting of Intracranial Aneurysms”. In: *IEEE transactions on visualization and computer graphics* 20.5 (2014), pp. 686–701.

[163] S. Oeltze-Jafra, J. R. Cebral, G. Janiga, and B. Preim. “Cluster Analysis of Vortical Flow in Simulations of Cerebral Aneurysm Hemodynamics”. In: *IEEE transactions on visualization and computer graphics* 22.1 (2016), pp. 757–766.

[164] R. Englund, T. Ropinski, and I. Hotz. “Coherence Maps for Blood Flow Exploration”. In: *Proc. of the EG workshop on Visual Computing for Biology and Medicine (VCBM)* (2016), pp. 79–88.

[165] M. Neugebauer, G. Janiga, O. Beuing, M. Skalej, and B. Preim. “Anatomy-Guided Multi-Level Exploration of Blood Flow in Cerebral Aneurysms”. In: *Computer Graphics Forum* 30.3 (2011), pp. 1041–1050.

[166] R. van Pelt, R. Gasteiger, K. Lawonn, M. Meuschke, and B. Preim. “Comparative Blood Flow Visualization for Cerebral Aneurysm Treatment Assessment”. In: *Computer Graphics Forum* 33.3 (2014), pp. 131–140.

[167] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts. “Visual comparison for information visualization”. In: *Information Visualization* 10.4 (2011), pp. 289–309.

[168] V. Verma and A. Pang. “Comparative flow visualization”. In: *IEEE transactions on visualization and computer graphics* 10.6 (2004), pp. 609–624.

[169] H.-G. Pagendarm and F. H. Post. “Studies in Comparative Visualization of Flow Features”. In: *Scientific Visualization.* 1994.

[170] D. Schneider, A. Wiebel, H. Carr, M. Hlawitschka, and G. Scheuermann. “Interactive Comparison of Scalar Fields Based on Largest Contours with Applications to Flow Visualization”. In: *IEEE transactions on visualization and computer graphics* 14.6 (2008), pp. 1475–1482.

[171] S. Oeltze-Jafra, M. Meuschke, M. Neugebauer, S. Saalfeld, K. Lawonn, G. Janiga, et al. “Generation and Visual Exploration of Medical Flow Data: Survey, Research Trends and Future Challenges”. In: *Computer Graphics Forum* 40.4 (2018), p. 860.

[172] S. Born, M. Markl, M. Gutberlet, and G. Scheuermann. “Illustrative visualization of cardiac and aortic blood flow from 4D MRI data”. In: *2013 IEEE Pacific Visualization Symposium (PacificVis).* IEEE, 2013, pp. 129–136.

[173] P. Angelelli and H. Hauser. “Straightening Tubular Flow for Side-by-Side Visualization”. In: *IEEE transactions on visualization and computer graphics* 17.12 (2011), pp. 2063–2070.

[174] B. Köhler, M. Meuschke, U. Preim, K. Fischbach, M. Gutberlet, and B. Preim. “2D Plot Visualization of Aortic Vortex Flow in Cardiac 4D PC-MRI Data”. In: *Bildverarbeitung für die Medizin 2015.* Informatik aktuell. Springer Berlin Heidelberg, 2015, pp. 257–262.

[175] M. Meuschke, B. Köhler, U. Preim, B. Preim, and K. Lawonn. “Semi-automatic Vortex Flow Classification in 4D PC-MRI Data of the Aorta”. In: *Computer Graphics Forum* 35.3 (2016), pp. 351–360.

[176] S. Glaßer, S. Oeltze, U. Preim, A. Bjørnerud, H. Hauser, and B. Preim. “Visual analysis of longitudinal brain tumor perfusion”. In: SPIE Proceedings. SPIE, 2013, 86700Z.

[177] M. Neugebauer, R. Gasteiger, O. Beuing, V. Diehl, M. Skalej, and B. Preim. “Map Displays for the Analysis of Scalar Data on Cerebral Aneurysm Surfaces”. In: *Computer Graphics Forum* 28.3 (2009), pp. 895–902.

[178] M. Neugebauer, R. Gasteiger, V. Diehl, O. Beuing, and B. Preim. “Automatic generation of context visualizations for cerebral aneurysms from MRA datasets”. In: *International Journal of Computer Assisted Radiology and Surgery (CARS)* 4 (Supplement 1) (2009), pp. 112–113.

[179] B. Köhler, M. Grothoff, M. Gutberlet, and B. Preim. “Pressure-based vortex extraction in cardiac 4D PC-MRI blood flow data”. In: *EuroVis 2018: Eurographics / IEEE VGTC Conference on Visualization 2018.* 2018, to appear.

[180] B. Köhler, U. Preim, M. Grothoff, M. Gutberlet, and B. Preim. “Adaptive Animations of Vortex Flow Extracted from Cardiac 4D PC-MRI Data”. In: *Bildverarbeitung für die Medizin (BVM).* 2016, pp. 194–199.

[181] B. Köhler, M. Grothoff, M. Gutberlet, and B. Preim. “Visualization of Cardiac Blood Flow Using Anisotropic Ambient Occlusion for Lines”. In: *Vision, Modelling und Visualization (VMV).* 2017.

[182] B. Köhler, M. Meuschke, U. Preim, K. Fischbach, M. Gutberlet, and B. Preim. “Two-Dimensional Plot Visualization of Aortic Vortex Flow in Cardiac 4D PC-MRI Data”. In: *Proc. of Bildverarbeitung für die Medizin.* 2015, pp. 257–261.

[183] B. Köhler, U. Preim, M. Grothoff, M. Gutberlet, K. Fischbach, and B. Preim. “Robust Cardiac Function Assessment in 4D PC-MRI Data of the Aorta and Pulmonary Artery”. In: *Computer Graphics Forum* 35.1 (2016), pp. 32–43.

[184] B. Köhler, M. Grothoff, M. Gutberlet, and B. Preim. “Visual and quantitative analysis of great arteries’ blood flow jets in cardiac 4D PC-MRI data”. In: *Computer Graphics Forum* 37.3 (2018), pp. 195–204.

[185] M. Meuschke, K. Lawonn, B. Köhler, U. Preim, and B. Preim. “Clustering of Aortic Vortex Flow in Cardiac 4D PC-MRI Data”. In: *Bildverarbeitung für die Medizin (BVM).* 2016, pp. 182–187.

[186] M. Meuschke, S. Voß, B. Preim, and K. Lawonn. “Exploration of blood flow patterns in cerebral aneurysms during the cardiac cycle”. In: *Computers & Graphics* 72 (2018), pp. 12–25.

[187] M. Meuschke, S. Voß, O. Beuing, B. Preim, and K. Lawonn. “Glyph–Based Comparative Stress Tensor Visualization in Cerebral Aneurysms”. In: *Computer Graphics Forum* 36.3 (2017), pp. 99–108.

[188] M. Meuschke, T. Gunther, P. Berg, R. Wickenhofer, B. Preim, and K. Lawonn. “Visual Analysis of Aneurysm Data using Statistical Graphics”. In: *IEEE transactions on visualization and computer graphics* (2018).

[189] **Benjamin Behrendt**, B. Köhler, D. Gräfe, M. Grothoff, M. Gutberlet, and B. Preim. “Semi-Automatic Vessel Boundary Detection in Cardiac 4D PC-MRI Data Using FTLE fields”. In: *Eurographics Workshop on Visual Computing for Biology and Medicine.* The Eurographics Association, 2016.

[190] D. Kainmuller, R. Unterhinninghofen, S. Ley, and R. Dillmann. “Level set segmentation of the heart from 4D phase contrast MRI”. In: *Medical Imaging 2008: Image Processing.* SPIE Proceedings. SPIE, 2008, pp. 370–377.

[191] Y. Y. Boykov and M.-P. Jolly. “Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images”. In: *Eighth IEEE International Conference on Computer Vision.* 2001, pp. 105–112.

[192] B. Köhler, U. Preim, M. Grothoff, M. Gutberlet, K. Fischbach, and B. Preim. “Motion-aware stroke volume quantification in 4D PC-MRI data of the human aorta”. In: *International journal of computer assisted radiology and surgery* 11.2 (2016), pp. 169–179.

[193] C. van Leeuwen. “Spatial-temporal pathline clustering based on FTLE fields”. Master’s Thesis. Delft: Delft University of Technology, 2014.

[194] P. G. Walker, G. B. Cranney, M. B. Scheidegger, G. Waseleski, G. M. Pohost, and A. P. Yoganathan. “Semiautomated method for noise reduction and background phase error correction in MR phase velocity data”. In: *Journal of Magnetic Resonance Imaging* 3.3 (1993), pp. 521–530.

[195] H. Hoppe. “New quadric metric for simplifying meshes with appearance attributes”. In: *Proceedings Visualization ’99 (Cat. No.99CB37067).* IEEE, 1999, pp. 59–510.

[196] G. Taubin, T. Zhang, and G. Golub. “Optimal surface smoothing as filter design”. In: *Computer Vision — ECCV ’96*. Vol. 1064. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1996, pp. 283–292.

[197] **Benjamin Behrendt**, B. Köhler, U. Preim, and B. Preim. “Enhancing visibility of blood flow in volume rendered cardiac 4D PC-MRI data”. In: *Bildverarbeitung für die Medizin*. Springer, 2016, pp. 188–193.

[198] **Benjamin Behrendt**. “Vollautomatische Visualisierung von kardialen 4D PC-MRI Blutflussdaten mit intelligenter Fokussierung auf Features”. Master’s Thesis. Otto-von-Guericke Universität Magdeburg, 2015.

[199] K. J. Zuiderveld, A. Koning, and M. A. Viergever. “Techniques for speeding up high-quality perspective maximum intensity projection”. In: *Pattern Recognition Letters* 15.5 (1994), pp. 507–517.

[200] A. F. Stalder, M. A. Gulsun, A. Greiser, and M. Jolly. “Fully automatic visualization of 4D Flow data”. In: *Proc. Intl. Soc. Mag. Reson. Med* 21 (2013), p. 1434.

[201] **Benjamin Behrendt**, P. Berg, B. Preim, and S. Saalfeld. “Combining Pseudo Chroma Depth Enhancement and Parameter Mapping for Vascular Surface Models”. In: *Eurographics Workshop on Visual Computing for Biology and Medicine*. The Eurographics Association, 2017.

[202] **Benjamin Behrendt**, P. Berg, O. Beuing, B. Preim, and S. Saalfeld. “Explorative Blood Flow Visualization using Dynamic Line Filtering based on Surface Features”. In: *Computer Graphics Forum* 37.3 (2018), pp. 183–194.

[203] N. Thibieroz. “Order-independent transparency using per-pixel linked lists”. In: *GPU Pro* 2 (2011), pp. 409–431.

[204] S. Zachow, P. Muigg, T. Hildebrandt, H. Doleisch, and H.-C. Hege. “Visual exploration of nasal airflow”. In: *IEEE transactions on visualization and computer graphics* 15.6 (2009), pp. 1407–1414.

[205] K. Mühler, C. Tietjen, F. Ritter, and B. Preim. “The medical exploration toolkit: An efficient support for visual computing in surgical planning and training”. In: *IEEE transactions on visualization and computer graphics* 16.1 (2010), pp. 133–146.

[206] **Benjamin Behrendt**, W. Engelke, P. Berg, O. Beuing, B. Preim, I. Hotz, et al. “Evolutionary Pathlines for Blood Flow Exploration in Cerebral Aneurysms”. In: *Eurographics Workshop on Visual Computing for Biology and Medicine*. The Eurographics Association, 2019.

[207] C. Darwin. *The Origin of Species by means of Natural Selection, or, the Preservation of Favoured Races in the struggle for life*. John Murray, 1859.

[208] T. Bäck. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.

[209] R. Kruse, C. Borgelt, C. Braune, S. Mostaghim, and M. Steinbrecher. *Computational Intelligence: A Methodological Introduction, 2nd Edition*. Springer, New York, 2016.

[210] F. Ritter, T. Boskamp, A. Homeyer, H. Laue, M. Schwier, F. Link, et al. “Medical image analysis”. In: *IEEE pulse* 2.6 (2011), pp. 60–70.

[211] S. Saalfeld, P. Berg, A. Niemann, M. Luz, B. Preim, and O. Beuing. "Semiautomatic neck curve reconstruction for intracranial aneurysm rupture risk assessment based on morphological parameters". In: *International journal of computer assisted radiology and surgery* 13.11 (2018), pp. 1781–1793.

[212] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. "A density-based algorithm for discovering clusters in large spatial databases with noise". In: *Kdd*. Vol. 96. 1996, pp. 226–231.

[213] T. Lawton. *DSCImageCalc v1.2a*. 2017.

[214] B. Preim, T. Ropinski, and P. Isenberg. "A Critical Analysis of the Evaluation Practice in Medical Visualization". In: *Eurographics Workshop on Visual Computing for Biology and Medicine*. The Eurographics Association, 2018, pp. 45–56.

[215] **Benjamin Behrendt**, S. Voss, O. Beuing, B. Preim, P. Berg, and S. Saalfeld. "VICTORIA - An interactive online tool for the VIrtual neck Curve and True Ostium Reconstruction of Intracranial Aneurysms". In: *Proc. of Bildverarbeitung für die Medizin (BVM)*. 2020, pp. 209–214.

[216] N. Paliwal, V. M. Tutino, H. Shallwani, J. S. Beecher, R. J. Damiano, H. J. Shakir, et al. "Ostium Ratio and Neck Ratio Could Predict the Outcome of Sidewall Intracranial Aneurysms Treated with Flow Diverters". In: *AJNR. American journal of neuroradiology* 40.2 (2019), pp. 288–294.

[217] S. C. Wong, O. Nawawi, N. Ramli, and K. A. Abd Kadir. "Benefits of 3D rotational DSA compared with 2D DSA in the evaluation of intracranial aneurysm". In: *Academic radiology* 19.6 (2012), pp. 701–707.

[218] J. Xiang, S. K. Natarajan, M. Tremmel, D. Ma, J. Mocco, L. N. Hopkins, et al. "Hemodynamic-morphologic discriminants for intracranial aneurysm rupture". In: *Stroke* 42.1 (2011), pp. 144–152.

[219] A. Lauric, M. I. Baharoglu, and A. M. Malek. "Ruptured status discrimination performance of aspect ratio, height/width, and bottleneck factor is highly dependent on aneurysm sizing methodology". In: *Neurosurgery* 71.1 (2012), pp. 38–45.

[220] P. Hart, N. Nilsson, and B. Raphael. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". In: *IEEE Transactions on Systems Science and Cybernetics* 4.2 (1968), pp. 100–107.

[221] P. Berg and O. Beuing. "Multiple intracranial aneurysms: A direct hemodynamic comparison between ruptured and unruptured vessel malformations". In: *International journal of computer assisted radiology and surgery* 13 (2017), pp. 83–93.

[222] J. R. Cebral, F. Mut, J. Weir, and C. Putman. "Quantitative characterization of the hemodynamic environment in ruptured and unruptured brain aneurysms". In: *AJNR. American journal of neuroradiology* 32.1 (2011), pp. 145–151.

[223] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or. "MeshCNN". In: *ACM Transactions on Graphics* 38.4 (2019), pp. 1–12.

[224] **Benjamin Behrendt**, S. Ebel, M. Gutberlet, and B. Preim. "A Framework for Visual Comparison of 4D PC-MRI Aortic Blood Flow Data". In: *Eurographics Workshop on Visual Computing for Biology and Medicine*. The Eurographics Association, 2018, pp. 117–121.

[225] M. Sigovan, P. Dyverfeldt, J. Wrenn, E. E. Tseng, D. Saloner, and M. D. Hope. "Extended 3D approach for quantification of abnormal ascending aortic flow". In: *Magnetic resonance imaging* 33.5 (2015), pp. 695–700.

[226] M. D. Steenwijk, J. Milles, M. Buchem, J. Reiber, and C. P. Botha. “Integrated visual analysis for heterogeneous datasets in cohort studies”. In: *IEEE VisWeek Workshop on Visual Analytics in Health Care.* Vol. 3. 2010, p. 3.

[227] P. Klemm, S. Oeltze-Jafra, K. Lawonn, K. Hegenscheid, H. Völzke, and B. Preim. “Interactive Visual Analysis of Image-Centric Cohort Study Data”. In: *IEEE transactions on visualization and computer graphics* 20.12 (2014), pp. 1673–1682.

[228] J. Bernard, D. Sessler, T. May, T. Schlomm, D. Pehrke, and J. Kohlhammer. “A visual-interactive system for prostate cancer cohort analysis”. In: *IEEE Computer Graphics and Applications* 35.3 (2015), pp. 44–55.

[229] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, H.-J. Schulz, M. Streit, et al. “Characterizing Guidance in Visual Analytics”. In: *IEEE transactions on visualization and computer graphics* 23.1 (2017), pp. 111–120.

[230] S. Voß, S. Glaßer, T. Hoffmann, O. Beuing, S. Weigand, K. Jachau, et al. “Fluid-Structure Simulations of a Ruptured Intracranial Aneurysm: Constant versus Patient-Specific Wall Thickness”. In: *Computational and mathematical methods in medicine* 2016 (2016), p. 9854539.

[231] T. Roleder, J. Jąkała, G. L. Kałuża, Ł. Partyka, K. Proniewska, E. Pociask, et al. “The basics of intravascular optical coherence tomography”. In: *Postepy w kardiologii interwencyjnej = Advances in interventional cardiology* 11.2 (2015), pp. 74–83.

[232] P. van Ooij, J. J. M. Zwanenburg, F. Visser, C. B. Majoie, E. vanBavel, J. Hendrikse, et al. “Quantification and visualization of flow in the Circle of Willis: time-resolved three-dimensional phase contrast MRI at 7 T compared with 3 T”. In: *Magnetic resonance in medicine* 69.3 (2013), pp. 868–876.

[233] B. Preim and P. Saalfeld. “A survey of virtual human anatomy education systems”. In: *Computers & Graphics* 71 (2018), pp. 132–153.

[234] A. S. Forsberg, D. H. Laidlaw, A. van Dam, R. M. Kirby, G. E. Kafniadakis, and J. L. Elion. “Immersive virtual reality for visualizing flow through an artery”. In: *IEEE Proceedings Visualization (VIS).* 2000, pp. 457–460.

[235] **Benjamin Behrendt**, L. Piotrowski, S. Saalfeld, B. Preim, and P. Saalfeld. “The Virtual Reality Flow Lens for Blood Flow Exploration”. In: *Eurographics Workshop on Visual Computing for Biology and Medicine.* The Eurographics Association, 2020, pp. 37–41.

[236] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. “Toolglass and magic lenses”. In: *Proc. of ACM SIGGRAPH.* ACM Press, 1993, pp. 73–80.

[237] P. Berg, S. Saalfeld, **Benjamin Behrendt**, S. Voß, G. Hille, and N. Larsen. “Local flow analysis in unruptured middle cerebral artery aneurysms with vessel wall enhancement”. In: *Proc. of 6th International Conference on Computational and Mathematical Biomedical Engineering - CMBE 2019 // CMBE19.* International Conference on Computational & Mathematical Biomedical Engineering, 2227-3085. Zeta Computational Resources Ltd, 2019, pp. 558–561.

[238] S. Ebel, J. Dufke, B. Köhler, B. Preim, **Benjamin Behrendt**, B. Riekena, et al. “Automated Quantitative Extraction and Analysis of 4D flow Patterns in the Ascending Aorta: An intraindividual comparison at 1.5 T and 3 T”. In: *Scientific reports* 10.1 (2020), p. 2949.

[239] P. Berg, **Benjamin Behrendt**, S. Saalfeld, G. Janiga, and O. Beuing. “Advanced blood flow filtering for the rupture risk assessment of intracranial aneurysms”. In: *8th World Congress of Biomechanics (WCB).* Dublin, Ireland, 2018.

[240] S. Ebel, B. Köhler, B. Preim, **Benjamin Behrendt**, B. Jung, C. Lücke, et al. "Vergleich der Drehrichtung aortalen Blutflusses in gesunden Probanden und Patienten mit bikuspider Aortenklappe". In: *101. Deutscher Röntgenkongress und 9. Gemeinsamer Kongress der DRG und ÖRG*. RöFo - Fortschritte auf dem Gebiet der Röntgenstrahlen und der bildgebenden Verfahren. Georg Thieme Verlag KG, 2020.

Publication List & Division of Work

The following section contains a list of publications that I was involved in during my doctoral studies, as well as a short description of the division of work for each paper. Unless explicitly stated otherwise, I implemented the respective methods and wrote the paper for all publications where I am the first author.

Journal Articles

Benjamin Behrendt, Philipp Berg, Oliver Beuing, Bernhard Preim, and Sylvia Saalfeld. "Explorative Blood Flow Visualization using Dynamic Line Filtering based on Surface Features". In: *Computer Graphics Forum* 37.3 (2018), pp. 183–194

O. Beuing supplied medical datasets, provided feedback and took part in the evaluation. P. Berg conducted the simulation and provided parts of Section 4.1 ("Extraction of Simulated Blood Flow"). B. Preim ans S. Saalfeld provided additional methodological feedback.

Conference Proceedings

Benjamin Behrendt, Benjamin Köhler, Uta Preim, and Bernhard Preim. "Enhancing visibility of blood flow in volume rendered cardiac 4D PC-MRI data". In: *Bildverarbeitung für die Medizin.* Springer, 2016, pp. 188–193

This work is an extension of my Master's Thesis [198], which was supervised by B. Köhler and B. Preim. For this publication, I extended the approach and conducted the expert interview with U. Preim.

Benjamin Behrendt, Benjamin Köhler, Daniel Gräfe, Matthias Grothoff, Matthias Gutberlet, and Bernhard Preim. "Semi-Automatic Vessel Boundary Detection in Cardiac 4D PC-MRI Data Using FTLE fields". In: *Eurographics Workshop on Visual Computing for Biology and Medicine.* The Eurographics Association, 2016

I implemented the EFTLE image generator. The images were loaded into a segmentation tool developed by B. Köhler, which was used for the evaluation with D. Gräfe. B. Köhler, B. Preim, M. Grothoff and M. Gutberlet provided methodological feedback. Additionally, M. Grothoff and M. Gutberlet also supplied us with datasets to use.

Benjamin Behrendt, Philipp Berg, Bernhard Preim, and Sylvia Saalfeld. "Combining Pseudo Chroma Depth Enhancement and Parameter Mapping for Vascular Surface Models". In: *Eurographics Workshop on Visual Computing for Biology and Medicine.* The Eurographics Association, 2017

P. Berg provided the simulated datasets used in the study, which were previously segmented by S. Saalfeld. Additionally, S. Saalfeld offered methodological feedback. B. Preim helped with the design of the user study.

Benjamin Behrendt, Sebastian Ebel, Matthias Gutberlet, and Bernhard Preim. "A Framework for Visual Comparison of 4D PC-MRI Aortic Blood Flow Data". In: *Eurographics Workshop on Visual Computing for Biology and Medicine.* The Eurographics Association, 2018, pp. 117–121

B. Preim provided methodological advice, S. Ebel and M. Gutberlet provided medical feedback and took part in the evaluation.

Philipp Berg, Sylvia Saalfeld, **Benjamin Behrendt**, Samuel Voß, Georg Hille, and Naomi Larsen. "Local flow analysis in unruptured middle cerebral artery aneurysms with vessel wall enhancement". In: *Proc. of 6th International Conference on Computational and Mathematical Biomedical Engineering - CMBE 2019 // CMBE19.* International Conference on Computational & Mathematical Biomedical Engineering, 2227-3085. Zeta Computational Resources Ltd, 2019, pp. 558–561

I contributed to this publication by adapting my research prototype to the specific research question, performing the local flow evaluation, providing screenshots for the figures and writing the respective section within the paper.

Benjamin Behrendt, Wito Engelke, Philipp Berg, Oliver Beuing, Bernhard Preim, Ingrid Hotz, and Sylvia Saalfeld. "Evolutionary Pathlines for Blood Flow Exploration in Cerebral Aneurysms". In: *Eurographics Workshop on Visual Computing for Biology and Medicine.* The Eurographics Association, 2019

S. Saalfeld and P. Berg wrote part of the Medical Background section and provided the simulated datasets. O. Beuing provided medical feedback and helped to select interesting cases for the evaluation. W. Engelke helped to design the evolutionary algorithm, wrote Section 2.2 ("Evolutionary Algorithms") and was heavily involved in the design of the technical evaluation. He also created the graphs (Figure 4 and 6) shown in the evaluation section. B. Preim, I. Hotz and S. Saalfeld provided methodological feedback.

Sebastian Ebel, Josefin Dufke, Benjamin Köhler, Bernhard Preim, **Benjamin Behrendt**, Boris Riekena, Bernd Jung, Christian Stehning, Siegfried Kropf, Matthias Grothoff, and Matthias Gutberlet. “Automated Quantitative Extraction and Analysis of 4D flow Patterns in the Ascending Aorta: An intraindividual comparison at 1.5 T and 3 T”. in: *Scientific reports* 10.1 (2020), p. 2949

For this publication, I provided technical feedback and wrote part of the section “Measurements and Flow Classification”.

Benjamin Behrendt, Samuel Voss, Oliver Beuing, Bernhard Preim, Philipp Berg, and Sylvia Saalfeld. “VICTORIA - An interactive online tool for the VIrtual neck Curve and True Ostium Reconstruction of Intracranial Aneurysms”. In: *Proc. of Bildverarbeitung für die Medizin (BVM).* 2020, pp. 209–214

O. Beuing provided a set of medical datasets, which were segmented and simulated by S. Saalfeld, S. Voss and P. Perg. From this set, S. Voss and P. Perg selected the datasets. S. Saalfeld provided additional methodological feedback.

Others

Philipp Berg, **Benjamin Behrendt**, Sylvia Saalfeld, Gabor Janiga, and Oliver Beuing. “Advanced blood flow filtering for the rupture risk assessment of intracranial aneurysms”. In: *8th World Congress of Biomechanics (WCB).* Dublin, Ireland, 2018

I adapted my research prototype to the research question, performed the local flow analysis, provided the screenshots and wrote a short section on the flow analysis.

Sebastian Ebel, Benjamin Köhler, Bernhard Preim, **Benjamin Behrendt**, Bernd Jung, Christian Lücke, Christian Krieghoff, Robin Gohmann, Boris Riekena, Michael Borger, Phillip Lurz, Matthias Grothoff, and Matthias Gutberlet. “Vergleich der Drehrichtung aortalen Blutflusses in gesunden Probanden und Patienten mit bikuspider Aortenklappe”. In: *101. Deutscher Röntgenkongress und 9. Gemeinsamer Kongress der DRG und ÖRG.* RöFo - Fortschritte auf dem Gebiet der Röntgenstrahlen und der bildgebenden Verfahren. Georg Thieme Verlag KG, 2020

For this publication, I provided technical and methodological feedback.