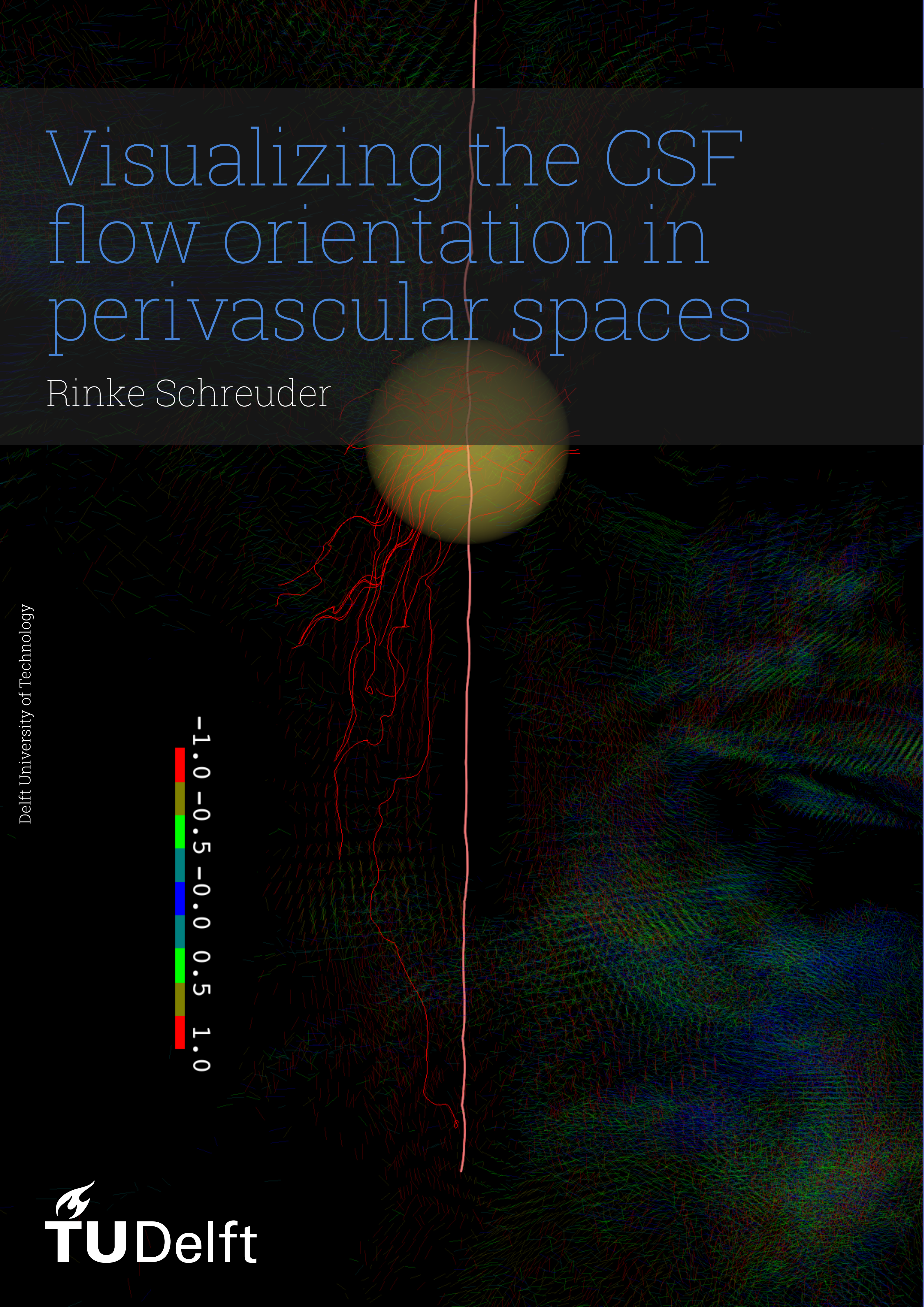
Visualizing the CSF flow orientation in perivascular spaces

Rinke Schreuder



Delft University of Technology

TU Delft

Visualizing the CSF flow orientation in perivascular spaces

by

Rinke Schreuder

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday July 11, 2025 at 13:00 AM.

Faculty:	Electrical Engineering, Mathematics and Computer Science	
Programme:	Master Computer Science	
Track:	Software Technology	
Research Group:	Computer Graphics and Visualization	
Student number:	4687787	
Project duration:	March 1, 2024 – June 12, 2025	
Thesis committee:	T. Höllt PhD,	TU Delft, Supervisor
	Dr. J.S. de Pinho Gonçalves,	TU Delft

Contents

Abstract

The brain and central nervous system handle waste transport differently from the rest of the body. The pathways through which waste in the brain is transported is still a debated topic. The glymphatic system describes a pathway from the perivascular spaces around veins and arteries to the subarachnoid space. This pathway is filled with Cerebrospinal Fluid (CSF) and exploring this fluid and the orientation of the flow relative to the blood vessels is essential to get new insight into the waste transport and related neurological issues. We analyze this system using a non-invasive diffusion weighted MRI sequencing technique called CSF-STREAM, which generates a DTI-like tensor field called CSF-Mobility, allowing us to track the CSF. The tensor data can be used to analyze the flow of the CSF by applying eigen-decomposition on the tensor matrix. We derive a vector field from the tensors' principal eigenvectors that represents the general flow of CSF. We introduce a workflow aimed at analysis, comparison and interpretation of the CSF data. It begins by selecting a region that includes a vessel and the surrounding CSF. The vessel is segmented from the region of interest (ROI) and its centerline is extracted to represent the vessel's orientation. The vector field then visualized using a hedgehog plot with a color mapping that indicates the relative orientation, as well as with a streamlines visualization. The streamlines are restricted to the vicinity of the vessel by generating seed points within a spherical radius surrounding the centerline. To make the visualizations easier to interpret, we apply two different transformations on the vector field. A straightening transformation is applied to the vector field by using straightened Curved Planar Reformation (CPR) on the vessel centerline. An additional unfolding transformation is introduced in which the vectors are rotated around the straightened centerline, creating a cross-sectional view of the vector field. We conduct a user study to evaluate the workflow and compare the visualization methods and to to see which methods work best for interpreting the relation between the flow orientation and the vessels. Results show that the workflow and visualizations are indeed suitable techniques and that the straightening of the vector field along the vessel makes it easier to interpret the data, while the unfolding transformation makes the context too complex to understand with limited time and explanation. Overall, the workflow and tool set have the capability to give researchers more insight into the waste transport in the brain.

1

Introduction

The lymphatic system takes care of waste products throughout the body via the lymph network carrying lymph fluid. This system is crucial in the proper functioning of the body. However, not much is known about how the waste transport works in the brain and the central nervous system (CNS) in general. While the brain consumes a lot of energy and generates plenty of waste, it does not have a lymphatic system to transport waste fluid or an equivalent system. Instead, it relies on Cerebrospinal Fluid (CSF), which is a clear, watery fluid similar to blood plasma that circulates throughout the CNS. It is generally thought that CSF has a function in waste clearance [21, 12, 40], however it is still being debated how exactly this works.

CSF has been found to play a role in the energy consumption of the brain and has been linked to functions of sleep [40] and to neurodegenerative diseases such as Alzheimer's disease [21, 33]. Finding out more on what drives the fluid and which pathways are used for the waste clearance may provide insight in these links and there are ongoing discussions on the possible pathways of the CSF. One such possibility is that waste is transported by the CSF flow in the perivascular spaces, the space surrounding blood vessels, to the subarachnoid spaces, and is known as the glymphatic system [12]. Figure 1.1 shows a simplified overview of the spaces containing CSF in the brain. Another proposed hypothesis is called the Intramural Peri-Arterial Drainage (IPAD) [3], which suggests that fluid moves along the smooth muscle cell membranes, however in this report the main focus is on the glymphatic system.

Because the blood vessels in the brain can be very small, it is not possible to directly observe or measure their activity. There are several methods to study the system and its driving forces, such as using microscopy [21, 25] or looking at the CSF diffusion using MRI, either by using a contrast agent or by applying diffusion tensor imaging (DTI, see Chapter 2). However these methods are not always viable. A microscopy approach would require a cranial window or use a tracer injection and is not viable in live humans. Using a contrast agent requires a lumbar injection [29], which is a very invasive measure with a lot of risk for the patient.

As each of these methods has its limitations, Harrison et al. [12] introduced a non-invasive approach to image the CSF. In MRI, blood and CSF are generally difficult to distinguish. Harrison et al.'s technique relies on the fact that CSF is mostly water, which has a longer echo time than the blood, meaning that the CSF signal can be isolated from the blood by simply waiting. By taking multiple scans with different velocity gradients, a tensor-like data field is constructed from which we can derive properties such as orientation and the velocity of the CSF flow. This tensor data allows Diffusion Tensor Imaging (DTI) to be used to visualize the diffusion and to derive the structures of the brain and vector field visualization can be used to represent the flow of CSF.

There does not exist much research on the directionality of CSF relative to the blood vessel. In previous work, Jos Huigen addressed this problem and a visualization was made to show the directionality [19]. Although this serves as a good first step and starting point for this thesis, there is still a lot of room for improvement, both in readability and accuracy, as well as in the variety visualizations that can be applied. The goal of this paper is to focus on comparing existing vector field visualizations in a

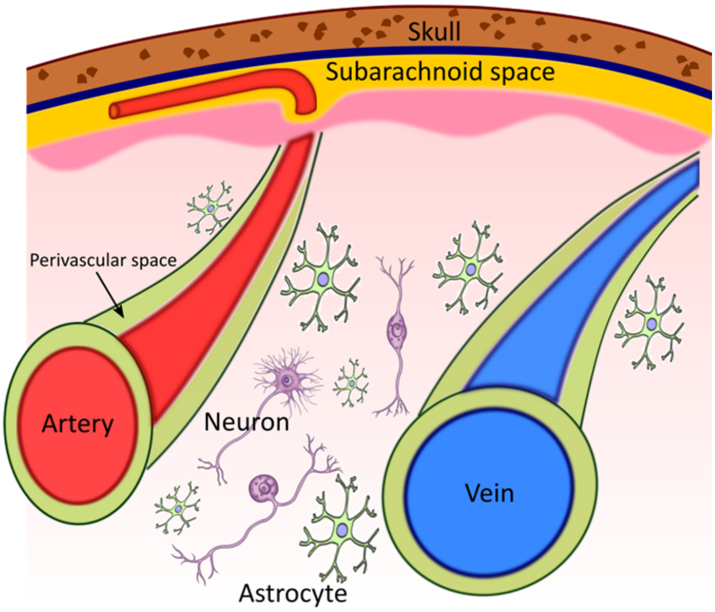


Figure 1.1: A representation of the artery and vein in the brain. The perivascular space is the space wrapped around the blood vessels and leads into the subarachnoid space. Image from "Visualizing the glymphatic system in humans" by L. Hirschler.

qualitative analysis of the CSF flow in the glymphatic system relative to blood vessels. Additionally, two transformations on the data are explored to see if they can simplify the vector field visualizations. The contribution is to provide a clear, interactive and responsive view and workflow of the CSF using multiple vector field visualizations. To achieve this, the medical visualization application '3D Slicer'[1] is used along with several tools, as will be described in Chapter 5.

2

Background

2.1. Diffusion Tensor Imaging

Magnetic Resonance Imaging (MRI) is used in medical imaging to produce in vivo images and has been used for decades. In MRI, a magnetic field of a certain strength interacts with the nuclei with spin in the body, either absorbing or emitting photons [17, 36]. A variant of MRI called diffusion weighted MRI (dMRI) is also used to track the water diffusion in human tissue for example in the brain. The water molecules can only move along specific directions, as they are restricted by the brain matter fibers. This is used to derive the orientation of the fibers. The acquired dMRI data contains the CSF signal. The diffusion tensors represent the flow orientation of the fluid using a three by three matrix $\mathcal{D}$.

$$\mathcal{D} = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{bmatrix}$$

The dMRI tensors are often analyzed by means of Eigen-decomposition, where the matrix $\mathcal{D}$ is represented by its eigenvectors and eigenvalues. The eigenvectors e_1, e_2, e_3 indicate the direction of the diffusion, which can be a linear, planar or spherical anisotropy depending on the constraints of the surrounding tissue. The corresponding eigenvalues of the diagonalized matrix, $\lambda_1, \lambda_2, \lambda_3$, are ordered such that λ_1 is the largest of the three. This indicates how much each eigenvector influences the diffusion direction, with the first eigenvalue being the main orientation. The eigenvectors and eigenvalues are commonly used in visualizing the diffusion tensors by deriving scalar invariants, such as mean diffusivity $\bar{D}$:

$$\bar{D} = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3}$$

and fractional anisotropy (FA):

$$FA = \sqrt{\frac{3}{2}}\frac{||D - \bar{D}I||}{||D||}$$

The scalar values can be converted to an iso-surface model or visualized by using volume-rendering by means of a transfer-function that maps a scalar value or property to some color or opacity. Although this scalar invariant approach is simple to implement, it also loses a lot of the depth from the the multi-dimensional tensor data as it is scaled down to a one-dimensional attribute.

The complete tensor information can be visualized using tensor glyphs. Glyphs are used because can represent all the necessary degrees of freedom in the data, using a separate attribute per dimension, such as a shape, color, or size. The local differences between each glyph give an indication of the flow. Even though this method is capable of representing each dimension of the tensor data, it is often limited to showing local information and can become cluttered quite easily, especially for a three dimensional space.

Another approach to simplify the visualization is to use the eigenvectors directly as a vector field. Vector fields are data fields containing a vector at each point, most often in two- or three-dimensional space. The vectors represent the orientation, direction or velocity of a flow at each position, which makes vector field visualizations prominent in research fields that involve fluid dynamics. In our case we use the e_1 eigenvectors, each represented by a line or arrow, showing their position and orientation. Color maps and shapes can help here to show additional information. This type of visualization creates a global overview of the 3D data.

2.2. The glymphatic system

As mentioned in Chapter 1, we follow the glymphatic system model, through which CSF is transported along a pathway of the perivascular spaces around blood vessels and the subarachnoid space. The flow of CSF has been linked to certain neurological issues related to sleep and diseases such as Alzheimers[40, 21, 33] and the way that the CSF may be linked is that it takes care of waste transport [28].

To understand this better we analyze this system by tracking the CSF flow using non-invasive imaging techniques, such as diffusion weighted MRI (dMRI). The problem is that the CSF signal must be separated from the blood, which Harrison et al. [12] tried to solve by using the different magnetic properties between blood and CSF. They developed a dMRI sequence and applied an ultra-long echo time and the signal is designed to attenuate the blood and parenchyma that surround the PVS, leaving only a trace of the CSF. This technique showed that when blood vessels nearby expand and contract there is approximately a 300% increase in the movement of CSF in the perivascular space.

Wen et al. attempt a similar method called dynamic Diffusion Weighted Imaging (dDWI) to find out more about the driving forces and CSF waveforms [39]. They demonstrate that the waveform of the fluid is dependent on the cardiac-cycle, that the CSF dynamics is closely related to the pressure of artery wall and that aging participants experience a wider systolic peak compared to younger participants.

Hirschler et al. [14] present a high resolution technique that measures CSF-mobility, inspired by the long echo time DTI approach by Harrison et al. and is called CSF-Selective T2-prepared REadout with Acceleration and Mobility-encoding (CSF-STREAM). CSF-mobility describes the velocity of the flow of CSF and is derived from the eigenvalues of the diffusion tensors in the CSF-STREAM data. This approach allows us to measure the CSF-mobility at a high resolution, including in the perivascular spaces. To find out more on the driving forces behind the CSF-flow, several cycles are compared with the CSF-mobility, such as the respiratory cycle, cardiac pulsation and the vasomotion, which is a spontaneous, rhythmic contraction of the blood vessels. These comparisons help identify the impact of each cycle [13, 16].

The CSF is measured using a non-motion-sensitized reference scan, along with motion-sensitizing gradients that are used encode the mobility in six orthogonal directions, from which a DTI-like tensor is derived [15]. From this tensor we can derive the necessary data fields discussed in Section 2.1. Figure 2.1 shows a saggital slice of a T2 prepared image of the mean cardiac phase.

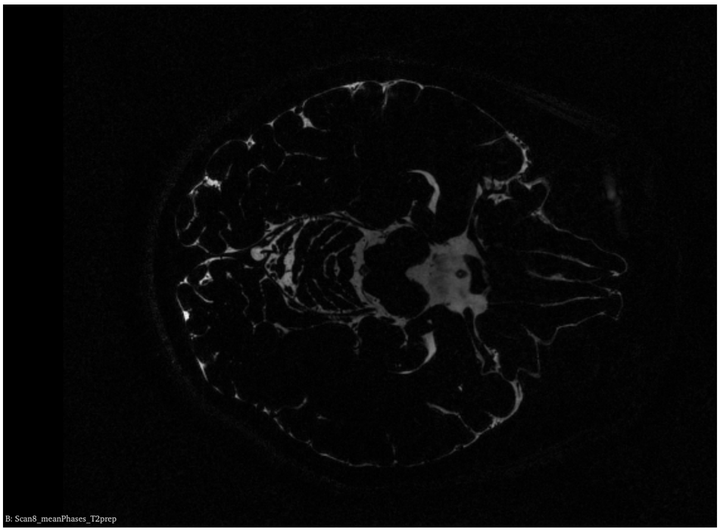


Figure 2.1: A saggital slice of a T2 mean phase prepared image.

2.3. Visualization

As mentioned earlier, tensor data is visualized with multiple techniques. Each method has its reasons to be used or to be avoided, depending on what needs to be conveyed to the user.

Glyphs represent multiple dimensions of tensor data, where six degrees of freedom are often enough for tensors in three-dimensional space. There are many different types of glyphs, although most commonly a cuboid or ellipsoid glyph is used. Figure 2.2 shows how shapes can be adjusted by using superquadric surfaces [23] to represent the type of anisotropy of the tensor, where a longer shape may represent a linear anisotropy and a flatter shape represents a planar anisotropy while a uniform shape represents an isotropic tensor. Superquadric tensor glyphs were introduced to address issues with asymmetry and ambiguity that cuboid and ellipsoid glyphs have.
The tensor glyphs display information at a local level and require little computation to generate, however they are more difficult to use with dense data fields, although it depends on the sample size. Increasing the data size or scaling up a dimension may cause the visualization to be cluttered and points to be obscured. Selecting a region of interest within the data can help with this, however limiting the scope means that it becomes more difficult to view the global spacial relations.

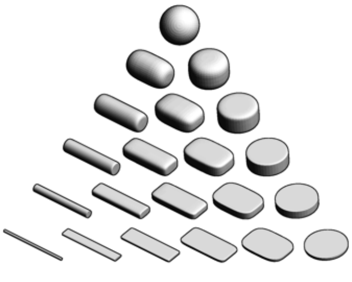


Figure 2.2: Superquadric tensor glyphs. [23]

The 3D vector field data can be visualized by simple glyph shapes such as arrows or lines, resulting in a hedgehog plot, instead of a more complex glyph required for a full 6D tensor. Each line or arrow in the hedgehog plot represents the position, orientation and magnitude of a vector, where arrows also indicate a specific direction. This type of direct mapping has the benefit that it is simple to implement, requires little extra calculation and is easy to interpret. On the other hand, this approach suffers from obscuring and scalability, especially in a three-dimensional space. Furthermore, if extra information from the data needs to be conveyed, one is limited to using color maps or changing the shape from a line or arrow to something more complex, which might make it more difficult to interpret the visualization.

Field lines are widely used to visualize fluid dynamics and work well with dense vector field data, and show a more continuous view of the flow. A field line is an integral path that a massless particle follows through a flow in the vector field, which can be time-dependent. The Euler method is a simple and widely used numerical integration technique to create field lines, tracing the flow path of a weightless particle through the vector field. The Euler method is the simplest variation of the Runge-Kutta (RK) methods as it is a first order equation, creating a rough approximation of the path. The accuracy can be improved with a smaller step size, however this also requires more steps. The higher order RK methods take intermediary estimates along the trajectory, giving the path a higher precision for the same step size, at the cost of extra computation time. The fourth order variation RK4 is the most common higher order RK method. The eigenvectors derived from the tensor data can also used for tractography, which is type of field line tracing, often tracking the white matter fibers in the brain. The starting points of each line are obtained from some seeding defined by the user. The disadvantage of using field lines is that the displayed information is three-dimensional, leaving out some degrees of freedom and the process is subject to clutter and error accumulation at each calculation step.

3

Domain description

This section covers the domain of this thesis. We outline the problem description, along with a brief abstraction of the data, followed by an abstraction of the tasks and requirements.

3.1. Problem domain

It is unclear which pathways and driving forces are behind the waste transport in the brain. This thesis follows the glymphatic system hypothesis in which the CSF crosses from the perivascular space around the blood vessels into the subarachnoid space. In order to analyze the diffusion of CSF in this system, the CSF-STREAM method is used to generate the CSF-Mobility tensors. The idea is to make a qualitative assessment of this data using interactive visualization tools. While it is beneficial to assess the data in a qualitative manner, there currently does not seem to exist such an interactive tool to specifically compare the orientation of the CSF to the vessels, aside from the visualizations created by Jos Huigen, which was not the main focus of his thesis. The goal for this then becomes to provide users with a tool set using vector field visualization methods and a workflow to analyze and compare them. The visualization methods are expanded upon by applying transformations on the data that are intended to make it easier to interpret the data.

Current visualizations of the CSF-mobility are difficult to interpret due to the fuzziness and ambiguity of the data, and the occlusion in 3D space. These visualizations do not make it clear how the CSF flow is oriented relative to the vessel. Viewing the flow from one angle may make it seem to align one way to the vessel, whereas another angle may give a different conclusion. It is unclear how exactly the CSF flow is expected to align to the vessel, so we take into account several characterizations that the flow could potentially have: the flow could be be aligned parallel or perpendicular to the vessel, or it could be a combination where the flow spirals around the vessel. It could also be none of these options with a turbulent flow.

3.2. Data abstraction

In Section 2.2 we discussed the CSF-STREAM method, where six gradients are applied on T2-prepared scans, which encode the velocity (or mobility) in six directions, along with an image without a gradient, which his used to improve the resolution. The motion encoding allows the calculation of a DTI-like tensor called CSF-Mobility from which the dataset is obtained. The required data consists of multiple voxel volumes derived from the CSF-Mobility tensors through the eigen-decomposition as described in Section 2.1. The first voxel volume is the dense data field of the principal eigenvectors as a 3D vector field, the second is a 3D scalar field of the T2 Mean Phase scan as a scalar field and the third is a scalar field called the ADC (Apparent Diffusion Coefficient). The original T2-prepared data is included in the set for convenience. The value range of the CSF-mobility is 0.04-0.06mm^2s^{-1} for larger spaces such as the MCA, and 0.01-0.02mm^2s^{-1} for smaller spaces. Any value higher than 0.1mm^2s^{-1} is considered to be noise. The volumes have a resolution of 450x556x422 voxels and the voxels have an isotropic resolution of 0.45mm^3.

3.3. Task abstraction

Given what has been described above, the workflow can be abstracted into three main tasks that the tool needs to support. Each of these tasks are important for the implementation, however the focus of this thesis lies on the third task, in which the data is presented using different methods.

- **T1** *Identify* an ROI of a blood vessel and the surrounding CSF based on anatomy.
- **T2** *Derive* a segmentation of the vessel and its orientation.
- **T3** *Compare* the motion of the CSF-flow with respect to the vessel orientation.
  - **T3a** *Present* the CSF-flow data using CSF-mobility.
  - **T3b** *Identify* shape features and similarities in the CSF motion relative to the vessel. We also want to identify coherency within a structure independent of vessel.

3.4. Requirements

For the implementation of the tool, we derive the additional requirements below. These help to direct the implementation and workflow and highlight which elements are more important to consider than others.

- **R1 (T1)** The user must be able to identify and segment an ROI using the datasets, shown in 3D or 2D slices. To identify ROIs the user requires knowledge of the anatomy of the brain.
- **R2 (T1)** The blood vessel should be aligned with the view of the user.
- **R3 (T2)** To extract the centerline we need to create a segmentation model of the vessel within the ROI.
- **R4 (T3a)** Using all voxels for the visualization of the field data can lead to clutter and occlusion. Individual elements must be distinguishable on the screen.
- **R5 (T3a)** It is useful to visualize the flow in the right context, so the system should be able to display the surrounding structures of the brain using volume rendering of the T2-prepared data, letting the user toggle it on or off.
- **R6 (T3b)** The orientation of each vector and the magnitude must be presented (e.g. by using a color map).
- **R7 (T3b)** To be able to identify the motion features of the flow, shape features (e.g., curvature, torsion) that might be expected to find must be clearly defined.
- **R11** The user must be able to interactively set the selection of the ROI/segmentation around the blood vessel and centerline extraction nodes.
- **R12** The program must be able to handle large datasets, remaining as responsive as possible and without significant performance issues such as crashes.

4

Related Works

We are interested in the flow of the CSF and we use vector fields derived from tensor data, therefore it is important to understand how and where vector field visualizations can be used. This means we need to explore relevant applications of vector field visualization techniques and how they are used differently.

4.1. Vector field visualization

The hedgehog plots and field line visualizations that we use to visualize the vector field data have many other applications in the medical field. The visualization of CSF using DTI-MRI is similar to the visualization of blood flow, which is used to identify patterns in the cardiac system [31] and irregularities related to cardiovascular diseases [24, 26]. There are several blood-flow visualization techniques related to MRI, however this is not limited to the 3D spatial and velocity relations. Markl et al. [24] review 4D flow MRI, which uses a time component as well. Such visualizations typically consist of a vector visualization like a hedgehog plot or a field lines method, where a slice-by-slice view displays a sequence of 3D blood-flow data. This 4D approach is applicable throughout the entire body and is great for quantitative evaluation of blood flow in a 3D volume. A downside to having this additional time component is that it also increases the data acquisition time significantly, although for the visualizing of the the data itself this is not necessarily an issue. Another small difference is that the standard 4D flow MRI approach used three-dimensional velocity encoding [24, 34], whereas the CSF-STREAM method uses six dimensions, although we reduce that to three dimensions.

Van Pelt et al. [27] introduced a framework with interactive visualization techniques, including exploded planar reformations, flow direction highlights and arrow trails. This framework automates the tracking of the temporal component instead of tediously keeping track manually and makes 4D MRI blood flow visualizations more readable. They cover more stylistic and illustrative methods compared to our work, although they also focus on improving perception in a local scope by reducing clutter. Since their data contains a time element, they make use of pathlines rather than streamlines, with a radial seeding strategy. To calculate the pathlines, they also make use of the fourth order Runge-Kutta method. Furthermore, van Pelt et al. use the planar reformation to provide a cross-section along the volume and is based on multi-planar reformation, whereas we use curved planar reformation (CPR) to straighten the vessel.

In an approach to simplify the cardiovascular system, several map-like representations have been created. Eulzer et al. [9] review these mappings and the domain requirements in a survey in which they classify literature based on the mapping type and provide recommendations depending on user tasks and data sources. The vessel maps are different abstractions of the vascular structures and focus on showing a global anatomical overview. However, not all of these methods are intended for visualizing flow characteristics in dense 3D vector fields, which is what our work focuses on. Although, some of these maps are still able to serve a role in simplifying the the flow structures of CSF and the vessels while preserving the relation of the flow to the vessel, such as a straightening map.

A major application of DTI and vector field visualization lies in the analysis of white matter in diffusion weighted MRI [6, 36]. Tractography is a streamline approach to represent the diffusion direction of fluids in the brain and its fiber bundles. The streamline visualizations can be enhanced by using shading or by using a 3D variation such as tubes instead of lines or stream surfaces [7, 20], which are bundles of the curved lines, approximated by a triangulated shape. Further abstractions attempt to improve interpretability, Chen et al. provide a pipeline to create a streamtape representation, which is an abstract version of the stream surface, and they use an entropy-based seeding [8]. Recently, Han et al. [11] proposed a neural network solution using k-means clustering methods and created the Deep Learning framework 'FlowNet', which can be applied on streamlines and flow surfaces. These method improve clarity, however they may not be applicable for our work as they are too abstract and lose detail, and are not always interactive as they require processing, for example using a neural network requires training a model.

Another medical application of vector field visualization is in respiratory research, such as nasal air flow visualization. Zachow et al. [41] demonstrate a technique to visually explore the complex airflow throughout the nose cavities using simulations. Similar to blood flow, this uses spatial and temporal data, however unlike the other applications, the data acquisition is not based on diffusion tensor MRI, instead it uses a high resolution computed tomography scan and simulates the airflow with computational fluid dynamics (CFD). They model the flow by generating a geometric model, formulating a physical model that describes the flow and solving a system of partial differential equations. The simulation results are then visualized with techniques such as field lines and line integral convolution (LIC) (along with techniques such as contour plots and iso-surfaces or volume rendering).

4.2. Curved Planar Reformatting

The workflow that will be explained later in Chapter 5 includes some techniques that reformat the vector data. As we discussed in the previous section, certain maps such as straightening can be applied to simplify a complex (flow) structure. Kanitsar et al. [22] present a technique to resample structures along a central axis called Curved Planar Reformation (CPR). This is used to visualize tubular structures such as blood vessels within one image. CPR and similar techniques such as Multi-Planar Reformation (MPR) are not limited to vascular structures and are widely used with MRI and computed tomography, as these can also be applied on spinal imaging [38, 18] and dental structures such as a jaw [5]. In many cases, CPR is used to obtain a straightened volume of the structure in question, providing clarity. In our case we can use CPR to straighten a vessel along its centerline and apply the straightening transformation to the surrounding vector field.

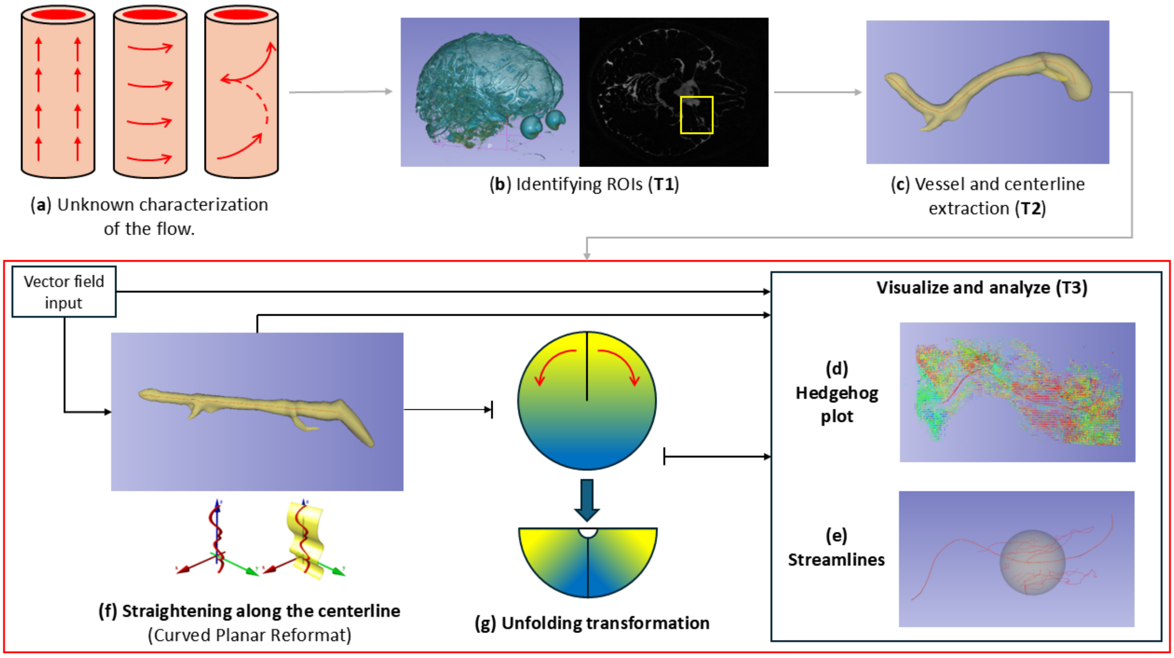
5

Method

We implement a tool along with a workflow to assist users in performing the tasks laid out in Section 3.3. This system adheres closely as possible to the requirements specified in Section 3.4. In this section we describe the steps used to complete the visualization tasks with the help of a pipeline. Figure 5.1 illustrates each stage in the workflow for visualizing the CSF data following a pipeline, from having an unknown flow (Figure 5.1a) to the visualizations and analysis (Figures 5.1d and 5.1e).

5.1. Pipeline overview

The first step of the pipeline is to find and select a suitable ROI that contains a vessel and its surrounding CSF (**T1**). This can be achieved by analysing a data volume such as a Fractional Anisotropy volume or a T2 mean phase dataset . Figure 5.1b shows the T2 reference scan of the brain in a 3D view Direct Volume Rendering, along with 2D views that show the axial, sagittal and coronal slices of the brain. The user can interact with the scene by moving the camera around, rotating the 3D volume or scrolling through the slices and after identifying the relevant anatomical structures, the user can set up a bounding box over the vessel with the surrounding area, indicating the ROI.


(**a**) Unknown characterization of the flow.

(**b**) Identifying ROIs (**T1**)

(**c**) Vessel and centerline extraction (**T2**)

(f) Straightening along the centerline
(Curved Planar Reformat)

(g) Unfolding transformation

Figure 5.1: The pipeline workflow consists of the three main tasks: Identifying an ROI, extracting a vessel and its centerline and presenting the vector field data.

The second step is the extraction of the vessel and its centerline (**T2**). The user has to extract the voxels that lie within in the selected vessel, either manually using selection tools, a threshold or other algorithms that may apply. The selected voxels can be used to create a polygonal model of the vessel. The user can then extract the centerline between two selected endpoints points on the model, which can also be selected automatically. The polygonal vessel and the end points are then used as input to a centerline extraction algorithm [4], for which we use the VMTK centerline extraction algorithm [35]. Figure 5.2 shows a polygonal mesh of the middle cerebral artery (MCA), with an extracted centerline in the middle of it. The algorithm used to extract the centerline uses a Vonoroi diagram approach, which can be seen as a series of maximal inscribed spheres generated along the vessel. Following the center points of these spheres creates a path that should be aligned with the vessel centerline. The centerline can thus be defined as a curve of control points located at the centers of the spheres. The sample size along the curve is important to consider, as the more control points the curve have, the more accurate the local vessel orientation we can derive. For thin vessels it may be required to apply some growing algorithm, as the surface model would otherwise be too small to use the Vonoroi algorithm.

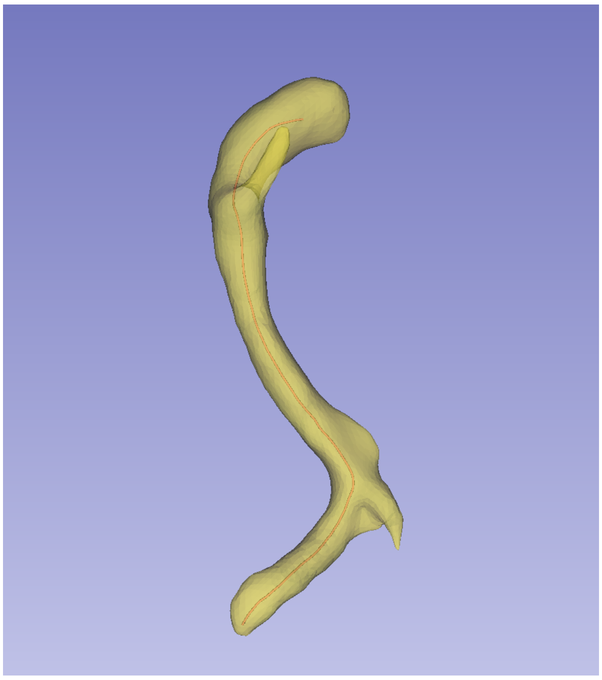


Figure 5.2: A segmentation of the MCA blood vessel and the extracted centerline.

The last step and main focus of this thesis is to visualize the vector field data of the perivascular spaces in relation to the vessels, with a selection of techniques (**T3**), which include a hedgehog plot for which a few different color maps are used, field lines that are focused using a 'lens', the straightening of the vessel center line and the unfolding the vessel. The hedgehog plot and the field lines are the visualization methods that will display the vector field data in a 3D view. The user compares the different presentations and should be able to extract information about the flow's motion by comparing the vector field's orientation to the orientation of the vessel's center line, as well as the coherency of the flow independent from the vessel.

5.2. Visualization workflow

For the presentation of the dense vector field data and to compare the motion of the CSF with respect to the centerline, two main vector field visualization techniques are applied, which are hedgehog plots and field lines. Both of these techniques have their uses and can be expanded upon. For the The T2 prepared scan and the FA datasets are presented using Direct Volume Rendering. The user can overlay the T2 or FA datasets. Before generating a visualization, the noisy data is filtered from the vector field. This is mostly done to remove some unnecessary vectors, which reduces clutter and calculation times.

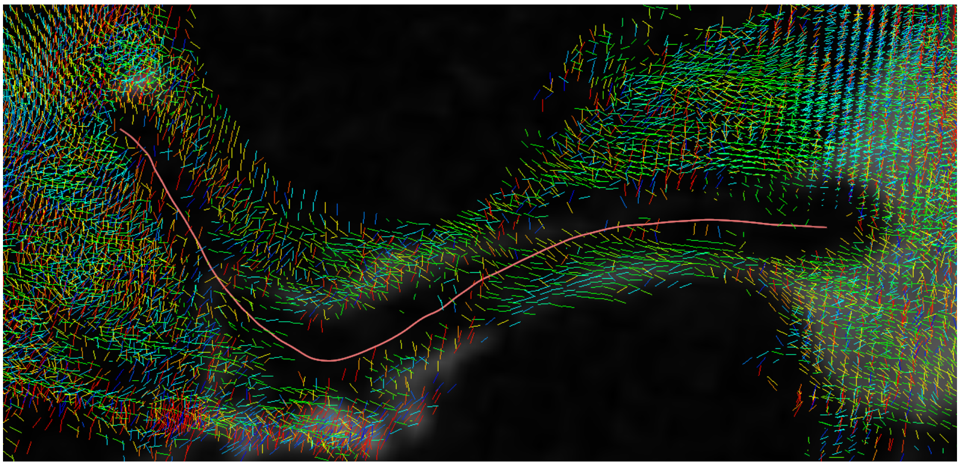
5.2.1. Hedgehog plot

We want to have a plot of the data that is easy to implement and serves as a reference to compare the readability of the other visualization methods. Figure 5.1d shows the first visualizations step, the hedgehog plot. These plots are intuitive and direct representations, where each vector is rendered as a line at a certain point with an orientation and magnitude. Lines are used as opposed to arrows, as the vector data only indicates orientation and not direction. The individual lines are difficult to distinguish for three-dimensional datasets, therefore color mappings are used to differentiate between the vectors.

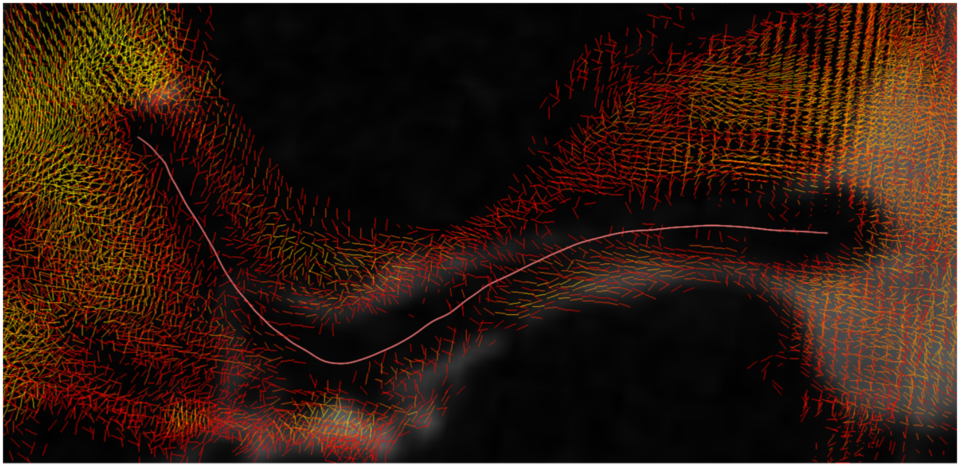
The first color mapping that is used here is a direct translation from XYZ-coordinates to RGB-color ($(r, g, b) = (x, y, z)$). In other words the vectors are colored component-wise, giving an indication of which vectors are oriented similarly. This is a simple way to differentiate between the vectors and does not require extra calculation.

The second color mapping takes a scalar volume (of the same dimensions as the vector field) as the color mapping. The color gradient used to color the vectors is based on the scalar values. This mapping does not require any extra calculation. This is useful in case the user wants to view the relation of the vector field to a specific scalar attribute such as the T2 data, FA or the ADC/CSF-Mobility values. However this does not relate back to the centerline.

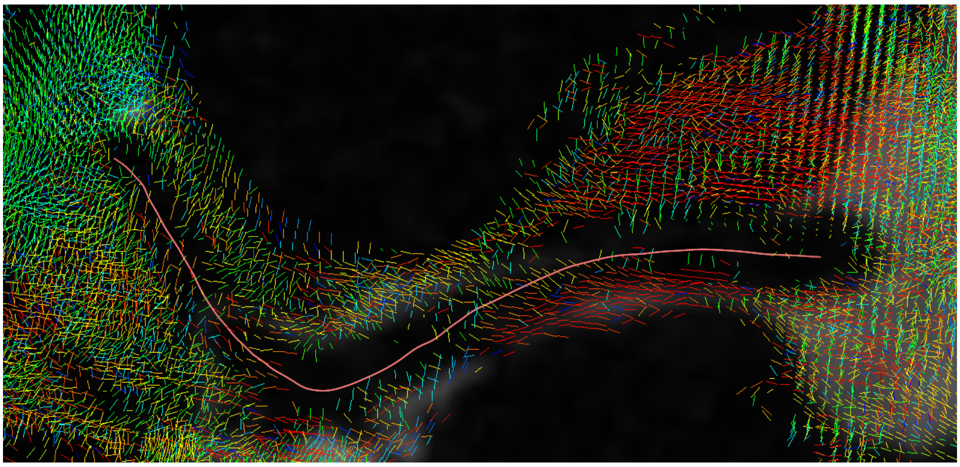
Figure 5.3 displays the hedgehog plots around the MCA with the RGB color mapping (5.3a), the scalar mapping 5.3b and the relative orientation (5.3c). Both the RGB and scalar volume color mappings show some flow structure, however they do not indicate their relation to the vessel orientation. To get such a relation, a third color mapping is generated based on the relative orientation of the vectors. We find the closest point on the centerline for each sample point and compare the orientation between the tangent vector at that closest point with the vector at the sampled point.
Since the extracted vessel centerline is a curve that consists of control points, it is possible to use a KD-tree to calculate the closest point n on the centerline for each point in the vector field p. In Figure 5.3c we show the relative orientation, where red vectors are aligned parallel to the centerline and blue vectors are oriented orthogonally. The sampling of the centerline determines the accuracy of the orientation, because if there are not enough control points, point p falls between two of the control points, such that the tangent vector no longer aligns correctly in curved areas. To mitigate this problem somewhat, the orientation at point n is calculated as the normalized difference between its two neighbouring points; $o = (n + 1) - (n - 1)$. This is not the case for the end points, since they only have one neighbouring point, so here it becomes $o = n - (n - 1)$ or $o = n + 1 - n$, depending on which end point direction you consider. This smoothes the orientation along the curve. The dot product is then calculated between the vector v_p at point p and o, which is used to create a color gradient such that the vectors are colored on how aligned they are to the centerline. A dot product equal to one means that the vector is aligned in parallel to the vessel, while a dot product equal to zero is oriented perpendicularly.



(a) Component-wise RGB



(b) Scalar volume



(c) Relative Orientation

Figure 5.3: The hedgehog plot of the CSF around the MCA using the three color mappings.

5.2.2. Field lines

To get a more global and continuous overview of the data, field lines are generated. We add a pipeline step in Figure 5.1e to apply these field lines. We use streamlines since our data does not have a time component. In Figure 5.4 we generate a series of streamline from different seeding positions. We have the option to choose between the Euler method or a more complex, high-order Runge-Kutta method (see section 2.3). Since we have to visualize the CSF flow around small, thin vessels with dense vector field data and since streamlines are sensitive to numerical errors and changing flows, it makes sense to opt for an algorithm with higher accuracy. Therefore we choose to generate streamlines using the RK4 algorithm, with seeding points being randomly generated.

The trade off is that RK4 requires more computation time, making the system less responsive. The trade-off will likely affect the efficiency of the workflow, however it can be mitigated slightly by adjusting parameters such as step-size, although this also affects the accuracy of the streamlines. Once the streamlines are generated they can be viewed responsively and saved to be compared, even though the visuals are no longer updated in real-time. It is important to note that the vector field only specifies the orientation of the vectors and not the direction, therefore the streamlines are calculated in both directions.

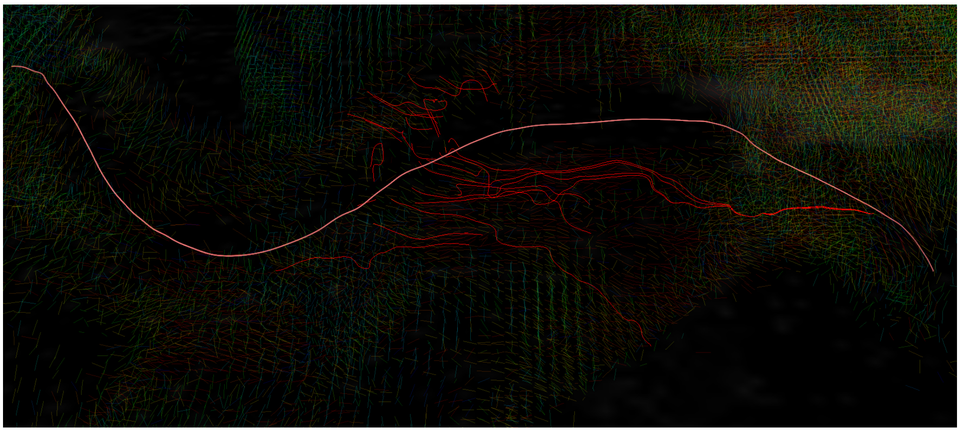


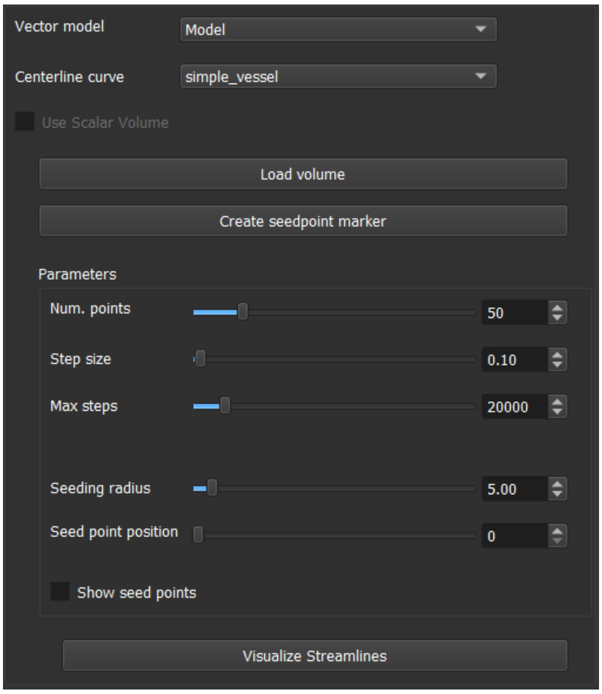
Figure 5.4: Streamline along the MCA centerline generated using the default settings, displayed over the hedgehog plot.

Having many streamlines overlapping makes it difficult to read, therefore it can help to limit the streamlines, to reduce the area in which streamlines are generated and let the user focus (interactively) on specific regions of the data. Therefore some simple stopping criteria are used, as well as a so called 'Magic Lens' to define a seeding area. The stopping criteria include default criteria, namely going out of bounds, encountering a zero vector or any vector that is too small.

The user can adjust some additional parameters, such as the step size and the maximum number of steps used in the algorithm and the number of seed points being generated. A smaller step size leads to a more accurate path, but it will take more steps to get the same path length. In theory, when using an infinitesimal step size and an unlimited amount of steps, the tool should generate a precise, continuous line until it encounters one of the default criteria and would not require a higher order RK4 algorithm. However, in practice this is not feasible and therefore we have to find settings that offer a high accuracy with as few steps as possible. For example, we set the default value for the step size to 0.1 voxels per step and the maximum number of steps is set to 20000, although most streamlines terminate well before the maximum steps are reached. Given that the step size may not be an integer and since vectors move in three dimensions simultaneously, the vectors are unlikely to fall directly onto the voxel centers. Therefore the vector field needs to be interpolated, for which we use trilinear interpolation between the vectors.

Generating a streamline from every point of the volume would take a very long time, reducing the responsiveness, as well as making the streamlines cluttered and obscured, making the streamlines infeasible to use. We can focus the streamlines by randomly generating seed points around the centerline. We add a magic lens, which defines a spherical area around the centerline in which the seeding points are generated. In our implementation, the points are randomly generated using a uniform distribution, although it would also be possible to use an exponential distribution to generate more points towards

the center of the magic lens. The density is determined by size of the sphere and the number of seed points that are generated within it, both parameters can be adjusted by the user. It is also possible to visualize the origins of the seed point as an indication of where the streamline starts.
This magic lens is placed at some point around the centerline to indicate the region where the user wants to generate streamlines. The centerline curve is defined by a number of control points, which we can use to place the spherical area, with the index chosen by the user by dragging a slider. The curve can be resampled to generate enough control points such that the appears to smoothly move along the centerline. The user can also select the number of seeding points to generate and change the radius of the magic lens. The complete parameter view is given in Figure 5.5.



Vector model: Model
Centerline curve: simple_vessel
☐ Use Scalar Volume
Num. points: 50
Step size: 0.10
Max steps: 20000
Seeding radius: 5.00
Seed point position: 0
☐ Show seed points

Figure 5.5: Parameter settings for the streamlines in the VectorStreamLines module.

5.2.3. Straightening the vessel center line

The curvature of the vessel makes it difficult to interpret the orientation of the flow with respect to the center line. By straightening the vessel the relation between the vector field and the centerline will be simplified and should be simpler to interpret. The straightening of the vessel is based on straightened Curved Planar Reformation (CPR) on the centerline curve [22]. This method is generally used to create a a cross-sectional image representation of a vessel, making it easy to see the varying diameter. Figure 5.6 gives some examples of the different CPR methods parallel to different axes. We will apply the straightened CPR method to transform the vector field.

An existing module in the SlicerSandbox extension [1] is wrapped in the tool. This module generates a transformation node that can be used to transform the points and vectors. This transformation is only applied on the domain itself and not the co-domain. For a scalar field this means only the positions are transformed while other attributes remain the same. However, in the case of a vector field we also need to apply the transformation to the co-domain, as we also want to transform the movement or orientation attribute. Therefore, we take the domain transformation from the Slicer module and apply the transformation on the co-domain ourselves. We do this by transforming the vectors by half-stepping in each direction, transforming those points and use them to create a new vector.

---

[1] https://github.com/PerkLab/SlicerSandbox/tree/master/CurvedPlanarReformat

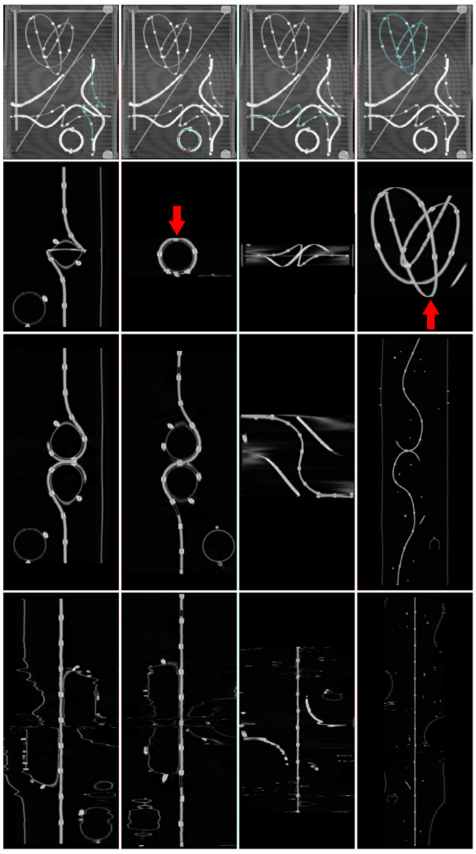


Figure 5.6: Top to bottom: MIP, projected CPR, stretched CPR, straightened CPR. left to right: major central-axis direction parallel to z-axis, parallel to y-axis, parallel to x-axis, in arbitrary direction. [22]

Next, we describe the process of the CPR and the vector field transformation in more detail. The straightened CPR algorithm calculates the tangent vector t_i at each point on the centerline c_i, creating a cross-sectional plane defined by the point and the vector. On this plane, a local coordinate system is defined by generating two vectors of the plane $\{u_i, v_i; u_i \perp v_i\}$. The line-of-interest l_i is defined by an angle-of-interest φ and the vectors of the plane, as shown in Equation 5.2.3, and the image offset for point p_{i+1} equals the distance between c_i and p_{i+1}.

$$l_i = \cos(\varphi)u_i + sin(\varphi)v_i$$

This process generates a global transformation that can be used to also transform the vector field. In general, the vector field must be transformed with the same image offset, only it is required to transform both the vector position and orientation. Since the vectors still only indicate orientation, each vector V_i is transformed by stepping halfway in both directions from the position P_i, then calculating the transformed points of both of these steps and calculate the new orientation based on the difference between these two points. Equations 5.2.3 and 5.2.3 show how each point P_i' and vector V_i' in the vector field are calculated given some transformation function $\mathbf{F}(p)$. Once the vector field has been transformed, the user can generate a new hedgehog plot or streamlines to visualize and compare the changes.

$$P_i' = \mathbf{F}(P_i)$$

$$V_i' = \mathbf{F}(P_i + 0.5V_i) - \mathbf{F}(P_i - 0.5V_i)$$

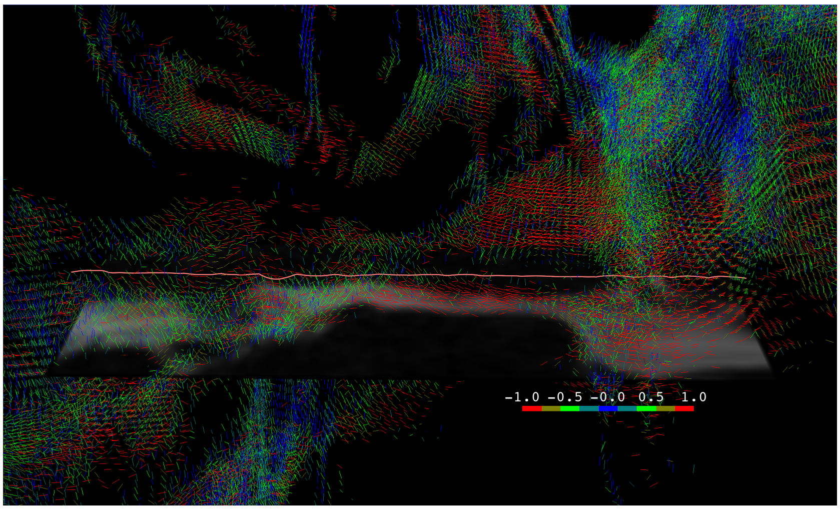


Figure 5.7: The vector field straightened along the MCA displayed with a hedgehog plot, using the relative orientation color mapping.

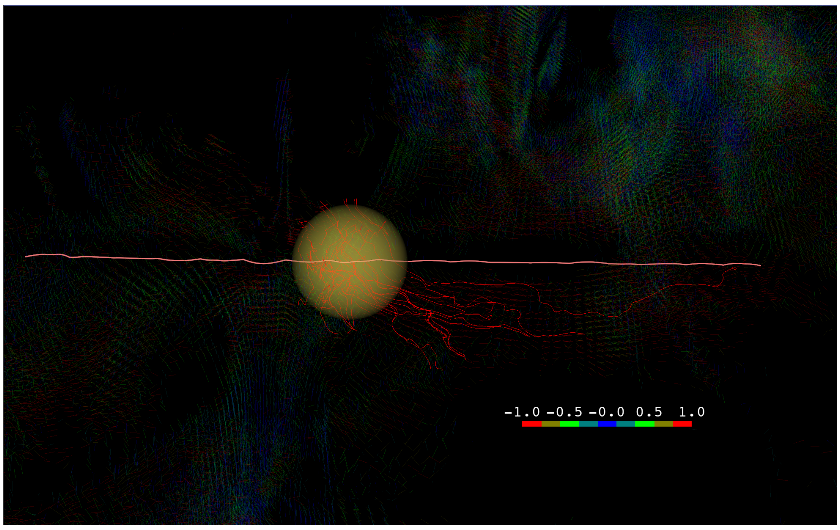


Figure 5.8: Streamlines along the straightened vessel, generated using the default parameter settings.

5.2.4. Unfolding the vessel

After the vessel and vector field have been straightened, the vector field can be unfolded along the vessel to get a cross-sectional view. This idea is based on the Nonlinear Virtual Colon Unfolding by Vilanova et al. [37], where the centerline of the vessel is used as the guiding path. However, instead of unfolding the surface to a sheet, points are transformed by rotating them around the centerline at the nearest point. The process of the unfolding transformation is calculated as follows.

The nearest point on the centerline C is again calculated for each point P of the vector field. However, since the centerline is now a straight line, this does not require a KD-tree, instead we can interpolate where along the line the nearest point is. Furthermore, the orientation of the centerline is now the same for every point along the line and can be defined by the end points. This orientation vector O is used as the axis around which to rotate each point. The curve's normal vector N is used to create a split where the new angle ρ for P is determined by the current angle α between P and N from point C, where V will rotate around O and away from N, and uses the mapping function given in Equation 5.2.4. Figure 5.9 illustrates the rotation of a vector around the centerline using this process.

$$\rho = 0.5\alpha + 90$$

Similar to the straightening of the vessel, each vector is transformed again by half-stepping in both directions and rotating the points there, see Equations 5.2.3 and 5.2.3. There are certain cases where these points fall on opposite sides of the unfolding split, transforming these points will rotate them in opposite directions, resulting in an incorrectly transformed vector that lies across the cross-section. To keep the process simple, these incorrect vectors will be dismissed. To simplify the unfolding further, a cylindrical volume is defined by the end points of the centerline and a user defined radius, such that the cylinder aligns with the straightened vessel. The unfolding transformation is only applied to vectors that lie within this volume. Once the vector field has been unfolded it can again be visualized and compared using a hedgehog plot or streamlines.

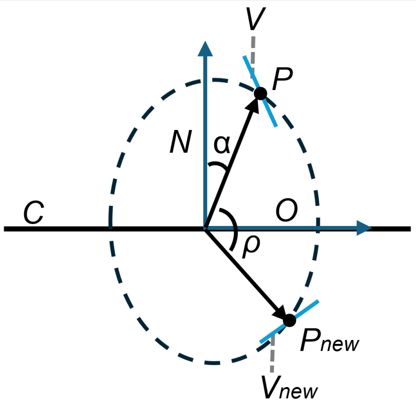


Figure 5.9: Sketch of the unfolding transformation. The vector V will rotate ρ degrees around axis O, which depends on the angle α between the vector's position P and the normal vector N.

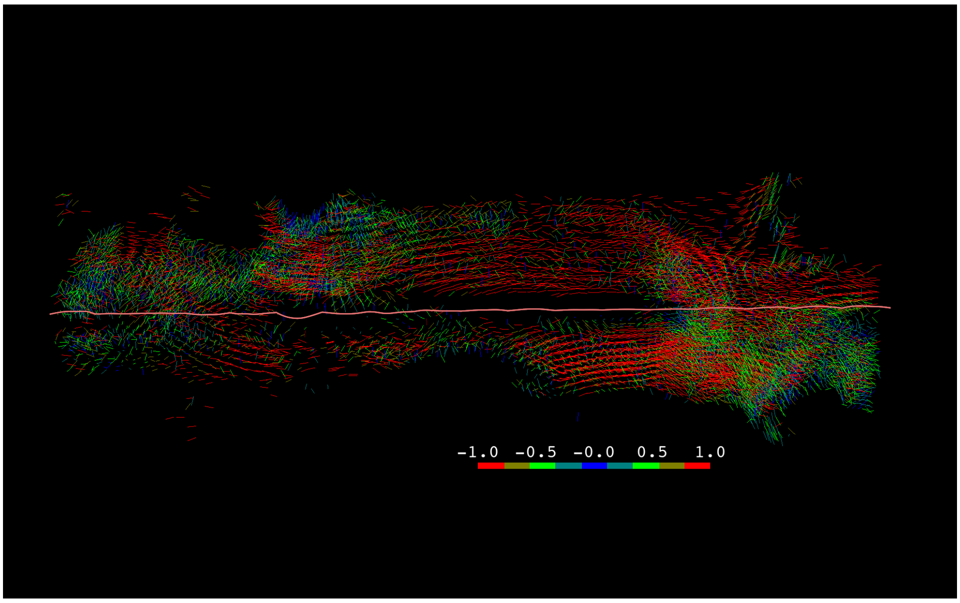


Figure 5.10: The unfolded vector field cross-section from above, displayed with a hedgehog plot using a relative orientation color map.

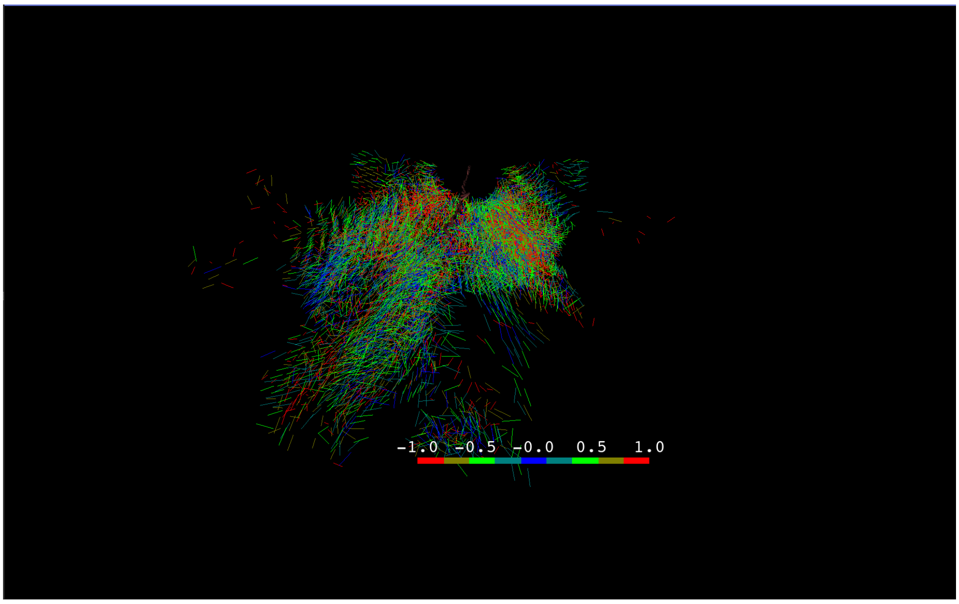


Figure 5.11: The unfolded vector field from the side.

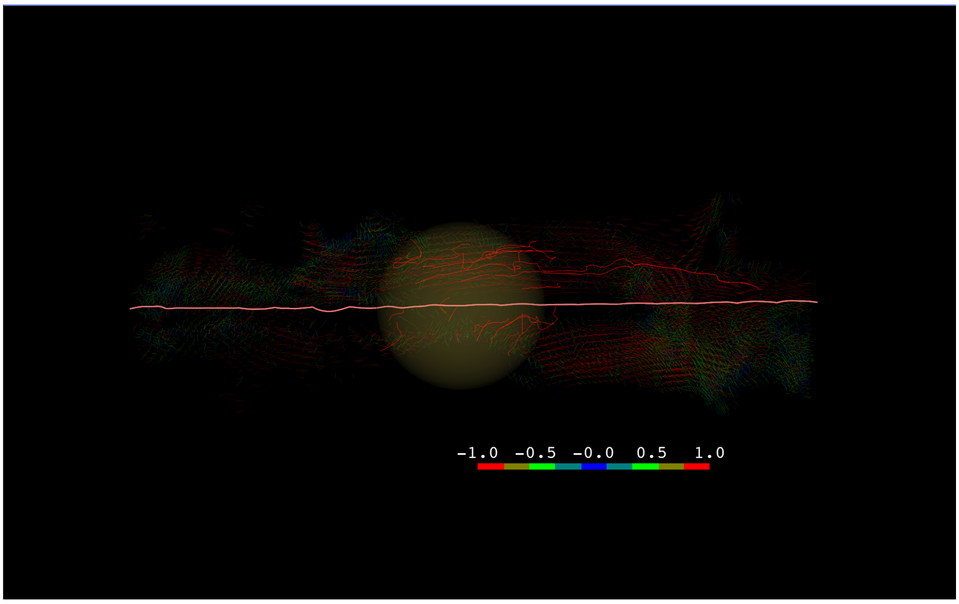


Figure 5.12: Streamlines along the unfolded vector field.

5.3. Software implementation

Given all the techniques in the visualization workflow above, we have to create a tool that allows all of these steps. The tool that we implement is created as as an extension for 3D Slicer [1, 10]. The extension consists of two modules, one as a python script and one as a loadable C++ module. The python script handles the data conversion, implements the centerline extraction, hedgehog plot visualization and unfolding transformation and wraps the curved planar reformation module for the straightening implementation. The C++ module adds the custom streamlines implementation, which includes the 'magic lens'. The entire extension also makes use of The Visualization Toolkit (VTK) [30].

5.4. Small vessel results

The steps discussed in this chapter were accompanied by the visualizations around the MCA, here we discuss some additional visuals of the CSF flow orientation around a thin vessel located towards the top of the brain. Figure 5.13 shows a slice of the T2 image, where the selected thin vessel is marked by the extracted centerline curve. To extract this curve, the vessel was segmented manually and a growing algorithm was applied to make sure the vessel surface is large enough for the centerline extraction algorithm. Figure 5.14 shows the hedgehog plots along the centerline with the relative orientation color map, including the un-transformed data (5.14a), the straightened centerline and vector field (5.14b) and the unfolded vector field within a 5 voxel radius (5.14c). Additionally, Figure 5.15 contains three sections of streamlines around the centerline. The streamlines show areas with a relatively perpendicular flow (5.15a), an area where a parallel flow and a perpendicular meet (5.15b) and an area with a bit more turbulent flow (5.15c).

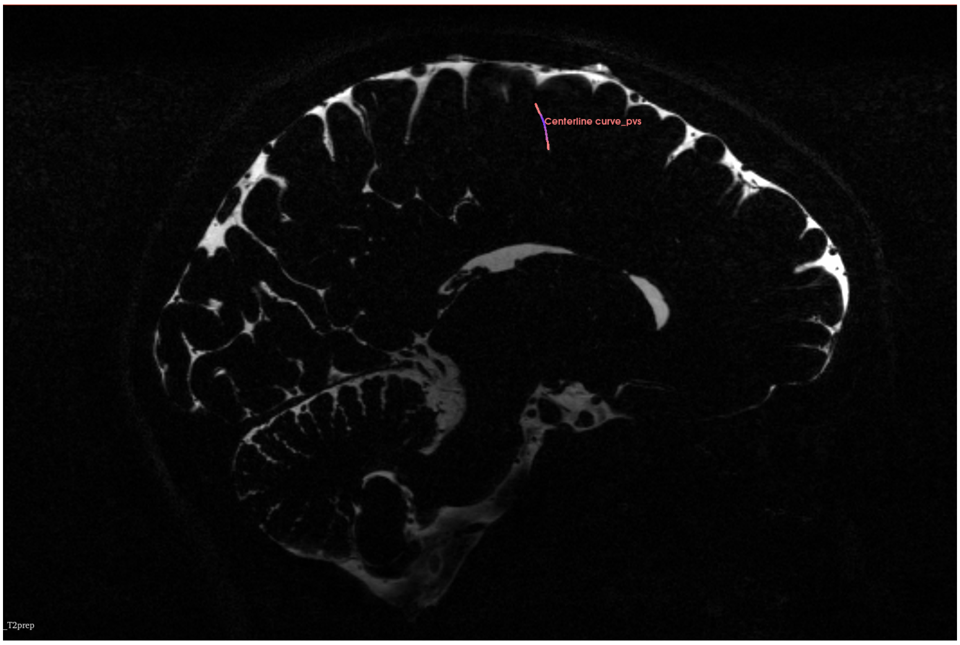
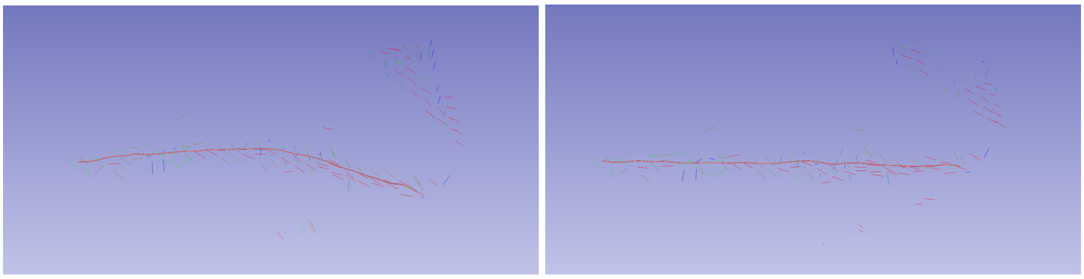
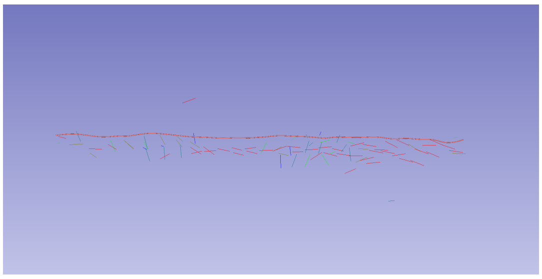


Figure 5.13: A slice of the T2 prepared image. The selected vessel is indicated with the centerline curve.

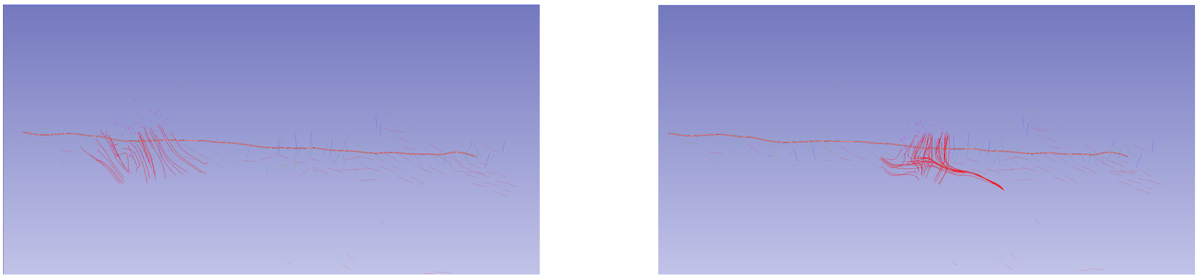


(a) Un-transformed **(b)** Straightened

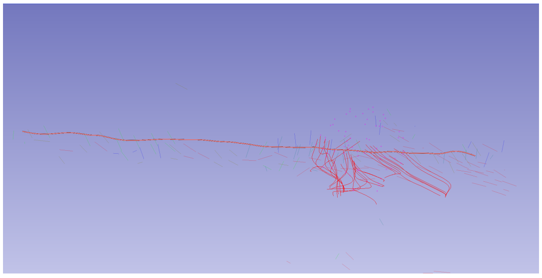


(c) Unfolded

Figure 5.14: The visualized steps applied on the vector field around a thin vessel, each using the relative orientation color mapping.



(a) Relative orientation hedgehog plot. **(b)** Scalar volume



(c) Relative Orientation

Figure 5.15: The streamlines along a thin vessel. These streamlines were generated using 30-40 random seed points, a step-size of 0.1 and a maximum step count of 5000 steps.

6

User study

The visualization methods and transformations discussed in Chapter 5 are aimed to help with interpreting the data and identifying the characteristics of the CSF flow. To evaluate if the discussed methods work as intended and if the described workflow is usable and efficient, we set up a user study with clinical researchers, which can be found in Appendix A along with the associated survey. We asked five participants to use the created tool and to fill in the survey. The participants have expertise in radiology and have some experience with medical imaging and visualization, however they do not necessarily have experience with vector field visualization or diffusion tensor imaging. The participants are given a brief introduction and are explained what the user study is about. Each participant is asked to go through the process described in the setup individually for around 30 minutes.

For the study the participants are asked to use the created tool set with some tasks. The created tool consists of two 3D Slicer modules, named 'VectorVis' and 'VectorStreamlines'. The former consists of input conversion, glyph generation, wrappers for centerline extraction and curved planar reformatting and lastly the unfolding transformation.

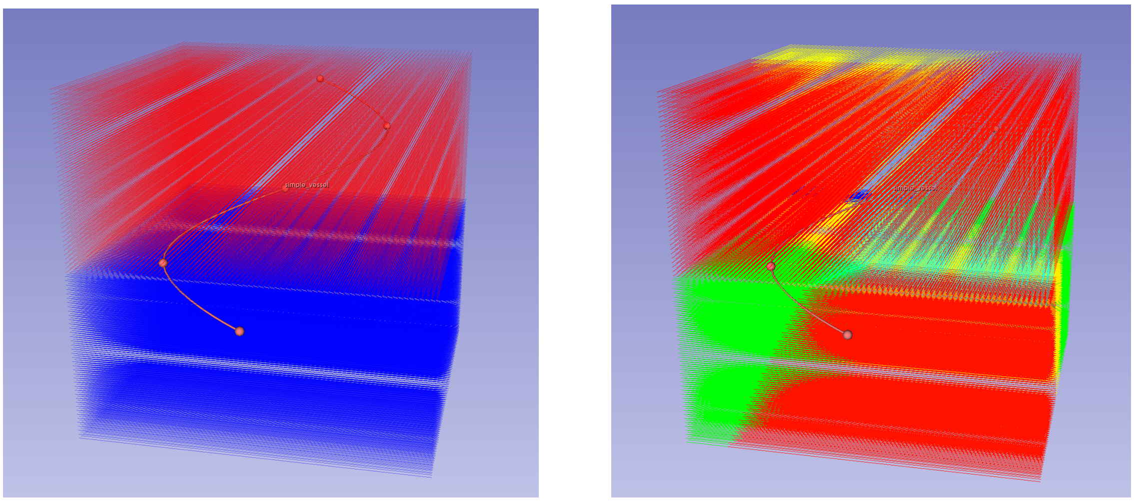
The user study consists of two parts, with the first part serving as an introduction and tutorial to using the tool and the second part serves to let the participants explore the tool and the data more freely. A separate dataset is used for each part, the first being a custom generated synthetic dataset and the second being a (pseudo) anonymous medical dataset provided by LUMC. Before each participant starts, the extension is already initialized in 3D Slicer and loaded with the relevant dataset. The participants were introduced to the to the setup and were explained that for each session the screen and audio would be recorded, asking them to share their thoughts out loud throughout the process.

The software was run on a home computer using an Intel i7-12700k CPU, GTX 1080Ti GPU and 32GB RAM, and a laptop with a remote connection was used for the participants to interact. For each session, the program would already be open and the testing setup for both the first and the second part were pre-loaded with the associated data. There were two or three program crashes throughout the entire study, which were quickly resolved and did not impact the process significantly. After the crashes of the program, the correct dataset was reloaded into the 3D Slicer program by the researcher. After the participants finished using the tool and having explored the data they were asked to fill in the survey.

6.1. Part 1: Tutorial

In the first part, the participants are explained how the tool works as they go through each step of the pipeline using the synthetic dataset. Figure 6.1a shows the visualized dataset, which consists of a cubic $80 \times 80 \times 80$ vector field where the top half of the vectors are oriented parallel to the x-axis direction and the bottom half follow the z-axis. A simple curve is used as representation of a vessel centerline that lies across both halves of the vector field. This dataset is not meant to be interpreted the same way that an actual medical dataset would be, rather it is meant to show how the visualizations and transformations work and to show that they work as expected.

The first step is to convert the synthetic vector volume to a poly-data model. Since the dataset is small enough, it does not need to be filtered. The next step is to generate the hedgehog plot in Figure 6.1b using the relative orientation color mapping.



(a) Red vectors are oriented along the x-axis and blue vectors are oriented along the z-axis.

(b) Vectors mapped using the relative orientation to the curve.

Figure 6.1: The synthetic dataset for the tutorial. The curve represents a vessel centerline.

The participant now switches to the streamlines module where the model is loaded and a seedpoint marker is generated that can be moved along the curve. After adjusting the parameters, the user can generate streamlines and optionally show the generated seed points. Figure 6.2 shows streamlines generated from the middle of the vector field, moving along the horizontal directions.

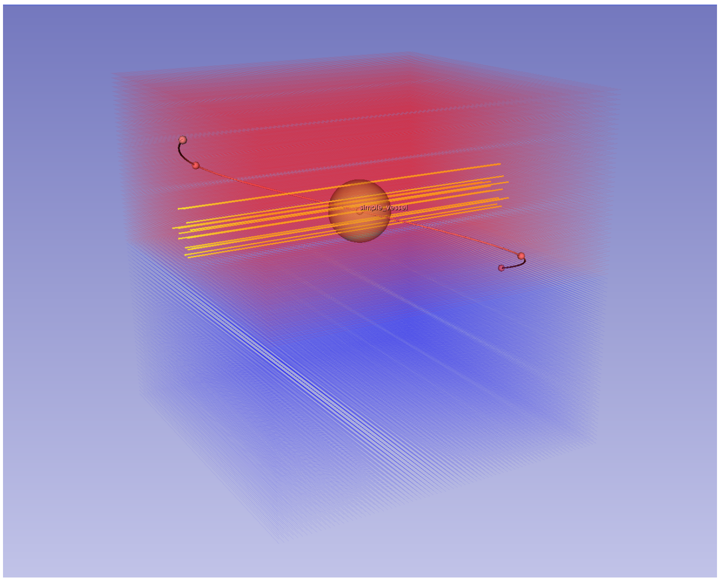


Figure 6.2: Yellow streamlines follow a path through the vector field along the x-axis

Lastly, the participant applies the transformations to the vector field and visualizing them again using the hedgehog plot and streamlines. The CPR wrapper and the unfolding algorithm both return new transformed vector field models that can be used as new input nodes in the currently selected nodes section.

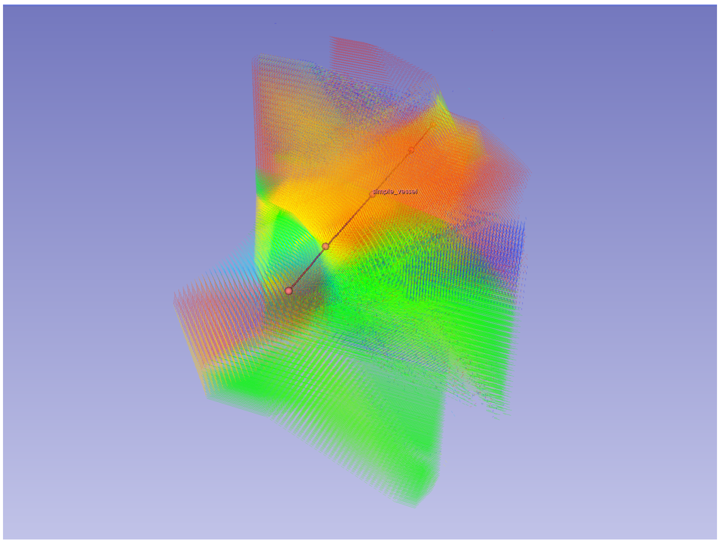
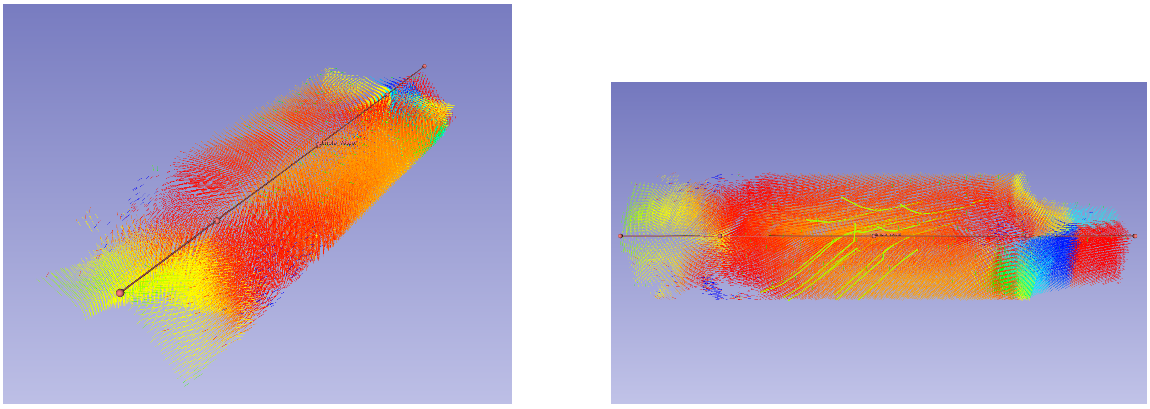


Figure 6.3: The synthetic vector volume following the straightening along the curve. The plot uses the relative orientation color mapping.



(a) The unfolded hedgehog plot using the relative orientation color mapping.

(b) Streamlines positioned in the unfolded vessel.

Figure 6.4: The unfolded vector volume.

6.2. Part 2: Exploration

The second part uses anonymized medical data, provided by LUMC for testing purposes. The ROI for this testing case will be the CSF flow around the Middle Cerebral Artery (MCA), as the vessel has already been segmented and an extracted centerline is provided as well. The T2 scan and ADC image volumes are provided to use as reference data. The user can scroll through the 2D slices of the T2 volume to make sure the data is aligned with the vectors. A bounding box ROI is also provided so that the users will not have to search for the selected vessel, since the focus of this part lies on trying the different techniques to visualize the vector field.

We ask the participants to explore the dataset, applying the tasks that they followed in the tutorial. However, since this dataset is too large the participant must select the vector volume with a mask volume to filter out vectors. It is recommended to use the T2 reference scan image as the filter mask, as it removes vectors coinciding with noisy data. The user is free to apply any of the visualization steps and parameters, for example in the case of the hedgehog plot, the user can now choose between the three color mappings, like in Figure 5.3. After generating a hedgehog plot, the user may need to apply and adjust an ROI bounding box to the generated plots to get a better view of the area around the centerline. A streamline can also be created at this stage, like the ones in Figure 5.4, where the user can try out the different parameters of Figure 5.5. Examples of the generated visualizations can be found throughout Chapter 5. With these results, the participant can make an attempt at analyzing and interpreting the data, as well as comparing the different visualization techniques to get an understanding of the CSF flow. After exploring the medical dataset, they are asked to fill in a survey on their experience of using the tool and their findings.

6.3. Survey questionnaire

The survey from part two of the user study consists of mostly open-ended questions about their experience with using the tool and the interpretation and preferences of the visualizations. The survey is not specifically meant to find conclusive properties of the flow, but rather to see if the visualizations are meaningful and and if the workflow is helpful in interpreting the data. The (groups of) questions, as they are given in the survey in Appendix A, are explained here.

The participants are first asked if they have used any other (vector field) visualization tools. This gives an idea of their experience. Next are three multiple choice questions, ranging one to five, on how they experience using the created tool, specifically if they found it difficult to use the tool, how much time they think it took to complete and if they feel successful using the tool. These questions are meant to show the effectiveness of the tool and the workflow.

- "How difficult do you feel it was to use the tool", 1 (very easy) to 5 (very difficult).
- "How much time do you feel it took you to use the tool?", 1 (very little time) - 5 (a lot of time).
- "How successful do you feel in using the tool", 1 (not successful) - 5 (very successful).

The participants will then be asked to analyze visualizations and interpret the CSF-flow to the best of their ability. This group of questions highlight the results generated by the tool and test if the visualization methods and transformations convey useful information and make it easier to interpret the vector orientation relative to the centerline. This begins with the multiple choice question: "How would you describe the flow of the CSF relative to the blood vessel/centerline", with the following options:

- It flows parallel to the vessel
- It flows perpendicular to the vessel
- The flow spirals around the vessel
- The flow is chaotic or random
- Other / Not sure

The participants are also asked a couple of open ended questions, which lets the participants explain their experience with using the tool and interpreting the CSF orientation:

- "Did you find any other trends or characteristics of the CSF flow?"
- "Which visualization method made it easiest to interpret the flow?"
- "Were there any visualization methods that gave confusing results?

The survey concludes by asking the participants if they encountered any problems with the usability of the tool and if they have any additional remarks. These questions are meant to indicate if there are any issues that could affect the participants experience using the tool and workflow.

7

Results

Following the user study, an assessment is made of user experiences by analyzing the survey results and the recordings. From these results, the effectiveness and efficiency of the visualizations will be determined. Beginning with the user experiences, each of the participants have previously used some form of medical imaging and visualization tool as such tools are essential for their background, however only two participants indicated having experience with vector field visualization. One of these participant mentioned having used the medical visualization program Paraview [2] for some basic glyph and streamline visualizations. The other participant mentioned having used the MRI simulation program Sim4Life[32], although that was some time ago.

After using the tool, the participants were asked to rate their experience of using the tool by indicating how difficulty it was to use, the time they felt it took to use and how successful they felt in using the tool, each on a scale from one to five. The average ratings of the perceived experiences are shown in Table 7.1 below and Table 7.2 contains the complete ratings results for each participant. The low average rating for the difficulty and experienced time likely means that the tool is indeed efficient, and the successfulness rating aligns with the usefulness. When taking these ratings together, it seems to imply that the tool and workflow work as intended. However, some nuance is required when looking at the ratings individually, as the two participants that had spent less time on using the tool also found it more difficult to use than the rest and one of them felt it took some time to use.

Difficulty	2.2
Experienced time	2.4
Successfulness	4.4

Table 7.1: The average ratings for the perceived experiences of using the tool. The difficulty of using the tool was given an average rating of 2.2, where 1 means very easy and 5 means very difficult, the perceived time spent on getting results was given a 2.4, where 1 means very little time and 5 a lot of time, and the success rating in getting results.

Participant	Difficulty	Time	Success
1	2	2	4
2	4	2	4
3	1	2	5
4	3	4	4
5	1	2	5

Table 7.2: The ratings per category for each of the five participant.

Note that the experienced time is not the same as the total time spent on the tasks, but rather how long they felt it took them to complete tasks. The planned session time for each participant was 30 minutes, however for two participants extra time was used to complete the tasks. This extra time was partially due to the process simply taking longer than expected, and partially due to the participants wanting to explore the tool more in-depth. Participants two and four spent slightly less time on their session, as there was a tight scheduling. This also reflected on their experience ratings, being less positive than the rest. The full time spent by each participant on both parts of the study is given in Table 7.3 below.

Participant	Part 1: Tutorial	Part 2: Exploration
1	15m	23m
2	11m	14m
3	19m	31m
4	10m	17m
5	15m	30m

Table 7.3: The time spent in minutes on Part 1 (Introduction and tutorial) and Part 2 (Exploring the medical dataset) per participant. Two of the participants ended up spending extra time on exploring in part 2 as there was extra time in the schedule and one participant spent less time than the allotted slot.

The participants were then asked to analyze and describe the orientation of the CSF-Mobility relative to the blood vessel using the different visualization techniques. Four of the five participants indicated that it flows parallel to the vessel, although they also mentioned that this is mostly for a specific area. Participants three and four remarked that it “Depends where along the vessel, sometimes [there is] more random motion.” and "differend direction in different regions". Participants one and five answered similarly: “Flow along the vessel, turbulent in larger regions.” Participant two found that it was not clear if the flow was parallel to the vessel and instead indicated that the flow had more chaotic/random motion.

Next, the participants decided which visualization method made it easiest to interpret and if there were any confusing visualization methods. This part gave some different answers, as participant two found it "hard to say" which visualization was easiest to use, one finding glyphs and streamlines helpful when used together. Overall there is a slight preference to streamlines compared to the hedgehog plot, with two participants preferring streamlines. Additionally, a participant remarked that the glyphs missed a legend for the color mapping which made it unclear which color corresponds to which axis. Furthermore, three participants indicating that the straightening along the centerline helped to analyze the data (both with hedgehog plots and streamlines). The unfolding transformation had less positive results, as at least two participants said they need more time is needed to fully understand and the transformation, aprticipant stating "The streamlines with unfolding became a bit too abstract." and “... I need more time to appreciate it.”

In the additional remarks there were some comments on the usage of the 3D Slicer program and module, where the amount of back and forth between modules seemed to be somewhat annoying, as participants needed to toggle certain models on or off to get a better view of a specific visual. Furthermore, participant two found that there was not enough time to really explore and participant four, saying that the tutorial was "more confusing than enlightening because of the abstract data", but that they understood more when looking at the CSF Mobility data. This, combined with the description of participant two on the relative flow of CSF being more chaotic or random, is likely related to the fact that these participants were not able to spend as much time on both the tutorial and the exploration, but perhaps also due to the fact that they were not that familiar with the topic of CSF in the perivascular spaces as participant 2 was "... struggling to understand all the options and concepts...". Given their remarks, it makes sense that they would rated the difficulty in Table 7.2 higher than the other participants. Nevertheless they both felt relatively successful. Lastly, there were some minor loading/processing issues during the sessions, but nothing that significantly impacted their experience. The participants found the tool "rich in information" and were looking forward to using it again.

The results indicate that the tool can indeed be useful in finding out more on the orientation of the CSF flow relative to a vessel. However, a recurring remark from the recordings is that it would take more time to fully understand what each of the visualizations actually shows in detail and to understand the context of the data after it has been transformed. Specifically the unfolding transformation was considered to be more complicated and required extra explanation and time to fully understand both how it actually transformed and what this meant for the data. On the other hand, participants found that the straightening along the vessel actually improved their understanding of the orientation of the CSF-Mobility relative to the vessel with both the hedgehog plot and the streamlines visualizations.

8

Discussion & Conclusion

The results the user study and the tool itself are promising, however there are a few aspects that deserve attention. For example, the created tool from the implementation has specific limitations, both due to the software it is built upon and due to the way the tool was implemented. The tool was originally constructed using one separate python notebook that ran on the Slicer kernel, however this eventually needed to be converted into separate Slicer modules, one using python and one using C++ to have more control and customization, making the tool set somewhat disjointed. This split also required extra setup and implementation time, which meant that less time was available for exploring additional techniques. The implemented vector field visualization methods already show a lot of information, however there are many more visualization styles, techniques and variations that would be good to explore in future work.
Furthermore, using the VTK image processing library set for the visualization of the data provides a solid basis to work with, however VTK makes handling the data more difficult, requiring an image data or volume format for some purposes and polygonal models for others. Converting this data adds unnecessary complexity and wait time to the usage of the tool, however it was required for the functionality of certain parts. In the python implementation, this was sped up by first converting the data to a numpy format and then converting the result back to the appropriate VTK format, while the C++ implementation was fast enough on its own.
The tool is only able to handle a certain amount of data, although we did not test onthis specificly. Given the size of the medical dataset as discussed in Chapter 3.2, it was necessary to reduce the size by filtering out noisy vectors to keep both the calculation time and memory usage functional. Using the T2 volume as a mask with a threshold seems to be sufficient here, although users have to be careful not to filter out too many vectors. The synthetic dataset used in the tutorial of the user study was small enough to use in its complete form and therefore did not require filtering out any vectors.

Another limitation is that the user study has a small group size and each participant had limited time to explore the data. The study has a small group of participants that have similar backgrounds, making it important to go in depth into the answers that were given, instead of looking for a wide range of quantitative data. Furthermore, although the participants have some experience with visualizing data, they were not familiar with the 3D Slicer software, which meant that extra time was needed to introduce how this program worked before they could fully utilize the created tool set. The time slots were originally planned to be only 30 minutes, excluding filling in the survey, thus even with time extension this took some time away from analyzing, interpreting and comparing the different visualization methods in depth. Despite this, the user study showed that the tool, workflow and visualization methods have their merit.

Apart from whether the straightening of the vessel and the unfolding transformations actually assist with interpreting the data, the transformation of the data field changes the context and it changes how the relation between the vessel and the orientation of the CSF flow should be interpreted. From the results it became clear that the unfolding transformation made the context too complex to understand, even though the intention was that the cross-sectional view would simplify the context.

Furthermore, the transformations also warp the data further away from the vessel more. After the straightening transformation the vector field only makes sense when looking at an area close to the vessel itself. However, this is not necessarily an issue, since using an ROI already limits the scope of the visualizations and the unfolding transformation creates a specific cylindrical ROI around the vessel, keeping the context to the nearby flow.
Additionally, while the straightening transformation still seems to work on thin vessels, the unfolding transformation does not change the vector field around the thinner vessels as much as it does for larger vessels. This was not part of the user survey, so it is unclear if it is helpful or not, however judging by the generated images the unfolding does not add much extra insight here.

This thesis set out to explore visualization methods for the CSF flow in the perivascular space relative to blood vessels, accompanied by techniques to simplify interpretation of the data as well as a workflow to use these methods. To achieve this, a workflow was introduced for users to compare two main visualization methods, hedgehog plots and streamlines, combined with a few techniques to help simplify the visualizations.
From the user study it was found that the hedgehog plot and streamlines visualization methods are both useful and easy to interpret the dense flow data, especially when combined with the straightening transformation along the vessel centerline. Unfortunately, the unfolding transformation is makes it too complex to interpret, requiring some improvements and extra time and explanation before it can be effective. Overall, it is possible and useful to characterize the CSF flow by using the the visualizations and workflow, even though it still requires some time.
Whether the unfolding transformation is something that can be improved to be more effective, or if there are other transformations that can be combined with visualization techniques not tested in this thesis is something that can be explored in future works. In the end, these methods combined with the workflow have the capability to give researchers more insight into the orientation of the CSF flow in the brain.

References

[1] *3D Slicer image computing platform*. `http://slicer.org/`. Last accessed: 02-06-2025.

[2] James Ahrens et al. "36-paraview: An end-user tool for large-data visualization". In: *The visualization handbook* 717 (2005), pp. 50038–1. url: `https://www.paraview.org/`.

[3] Nazira J. Albargothy et al. "Convective influx/glymphatic system: tracers injected into the CSF enter and leave the brain along separate periarterial basement membrane pathways". In: *Acta Neuropathologica* 136.1 (May 2018), pp. 139–152. issn: 1432-0533. doi: `10.1007/s00401-018-1862-7`. url: `http://dx.doi.org/10.1007/s00401-018-1862-7`.

[4] Luca Antiga, Bogdan Ene-Iordache, and Andrea Remuzzi. "Centerline computation and geometric analysis of branching tubular surfaces with application to blood vessel modeling". In: (2003).

[5] S. Barone, A. Paoli, and A. V. Razionale. "CT segmentation of dental shapes by anatomy-driven reformation imaging and B-spline modelling". In: *International Journal for Numerical Methods in Biomedical Engineering* 32.6 (Oct. 2015). issn: 2040-7947. doi: `10.1002/cnm.2747`. url: `http://dx.doi.org/10.1002/cnm.2747`.

[6] Peter J. Basser et al. "In vivo fiber tractography using DT-MRI data". In: *Magnetic Resonance in Medicine* 44.4 (2000), pp. 625–632. doi: `https://doi.org/10.1002/1522-2594(200010)44:4<625::AID-MRM17>3.0.CO;2-O`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/1522-2594%28200010%2944%3A4%3C625%3A%3AAID-MRM17%3E3.0.CO%3B2-O`. url: `https://onlinelibrary.wiley.com/doi/abs/10.1002/1522-2594%28200010%2944%3A4%3C625%3A%3AAID-MRM17%3E3.0.CO%3B2-O`.

[7] S Born et al. "Illustrative Stream Surfaces". In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (Nov. 2010), pp. 1329–1338. issn: 1077-2626. doi: `10.1109/tvcg.2010.166`. url: `http://dx.doi.org/10.1109/TVCG.2010.166`.

[8] Cheng-Kai Chen et al. "An illustrative visualization framework for 3d vector fields". In: *Computer Graphics Forum*. Vol. 30. 7. Wiley Online Library. 2011, pp. 1941–1951.

[9] P. Eulzer et al. "Vessel Maps: A Survey of Map-Like Visualizations of the Cardiovascular System". In: *Computer Graphics Forum* 41.3 (June 2022), pp. 645–673. issn: 1467-8659. doi: `10.1111/cgf.14576`. url: `http://dx.doi.org/10.1111/cgf.14576`.

[10] Andriy Fedorov et al. "3D Slicer as an image computing platform for the Quantitative Imaging Network". In: *Magnetic Resonance Imaging* 30.9 (Nov. 2012), pp. 1323–1341. issn: 0730-725X. doi: `10.1016/j.mri.2012.05.001`. url: `http://dx.doi.org/10.1016/j.mri.2012.05.001`.

[11] Jun Han, Jun Tao, and Chaoli Wang. "FlowNet: A deep learning framework for clustering and selection of streamlines and stream surfaces". In: *IEEE transactions on visualization and computer graphics* 26.4 (2018), pp. 1732–1744.

[12] Ian F Harrison et al. "Non-invasive imaging of CSF-mediated brain clearance pathways via assessment of perivascular fluid movement with diffusion tensor MRI". In: *eLife* 7 (July 2018). issn: 2050-084X. doi: `10.7554/elife.34028`. url: `http://dx.doi.org/10.7554/eLife.34028`.

[13] Lydiane Hirschler et al. "Effects of the cardiac and respiratory cycles on CSF-mobility in human subarachnoid and perivascular spaces". In: *Joint Annual Meeting ISMRM-ESMRMB ISMRT 31st Annual Meeting*. ISMRM-ESMRMB. ISMRM. doi: `10.58530/2022/0320`. url: `http://dx.doi.org/10.58530/2022/0320`.

[14] Lydiane Hirschler et al. "High resolution T2-prepared MRI enables non-invasive assessment of CSF flow in perivascular spaces of the human brain". In: *Proceedings of the 27th Annual Meeting of ISMRM, Montreal, Quebec, Canada*. Vol. 746. 2019.

[15] Lydiane Hirschler et al. "Region specific drivers of cerebrospinal fluid mobility as measured by high-resolution non-invasive MRI in humans". In: (Oct. 2024). DOI: `10.21203/rs.3.rs-3178346/v1`. URL: `http://dx.doi.org/10.21203/rs.3.rs-3178346/v1`.

[16] Lydiane Hirschler et al. "The driving force of glymphatics: influence of the cardiac cycle on CSF-mobility in perivascular spaces in humans". In: *Proc of the Int Soc of Mag Res in Medicine* (2020).

[17] Joseph Hornak. "The basics of MRI". In: Jan. 2010.

[18] Shuang Hou et al. "Coronal Curved Planar Reformation for Vertebra Localization in Pelvic Radiotherapy CT Images". In: *2022 23rd ACIS International Summer Virtual Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD-Summer)*. IEEE, July 2022, pp. 125–128. DOI: `10.1109/snpd-summer57817.2022.00029`. URL: `http://dx.doi.org/10.1109/SNPD-Summer57817.2022.00029`.

[19] Jos Huigen. "Directionality of CSF-Mobility in the Human Subarachnoid and Perivascular Space". In: (2023).

[20] J.P.M. Hultquist. "Constructing Stream Surfaces in Steady 3D Vector Fields". In: *Proceedings of Visualization '92*. (also RNR Tech Report 92-025). Boston, MA, Oct. 1992, pp. 171–178.

[21] Jeffrey J. Iliff et al. "A Paravascular Pathway Facilitates CSF Flow Through the Brain Parenchyma and the Clearance of Interstitial Solutes, Including Amyloid β". In: *Science Translational Medicine* 4.147 (Aug. 2012). ISSN: 1946-6242. DOI: `10.1126/scitranslmed.3003748`. URL: `http://dx.doi.org/10.1126/scitranslmed.3003748`.

[22] A. Kanitsar et al. "CPR - curved planar reformation". In: *IEEE Visualization, 2002. VIS 2002.* VISUAL-02. IEEE. DOI: `10.1109/visual.2002.1183754`. URL: `http://dx.doi.org/10.1109/VISUAL.2002.1183754`.

[23] Gordon Kindlmann. "Superquadric tensor glyphs". In: *Proceedings of the Sixth Joint Eurographics - IEEE TCVG Conference on Visualization*. VISSYM'04. Konstanz, Germany: Eurographics Association, 2004, pp. 147–154. ISBN: 390567307X.

[24] Michael Markl et al. "4D flow MRI". In: *Journal of Magnetic Resonance Imaging* 36.5 (Oct. 2012), pp. 1015–1036. ISSN: 1522-2586. DOI: `10.1002/jmri.23632`. URL: `http://dx.doi.org/10.1002/jmri.23632`.

[25] Humberto Mestre et al. "Aquaporin-4-dependent glymphatic solute transport in the rodent brain". In: *eLife* 7 (Dec. 2018). ISSN: 2050-084X. DOI: `10.7554/elife.40070`. URL: `http://dx.doi.org/10.7554/eLife.40070`.

[26] R. van Pelt et al. "Comparative Blood Flow Visualization for Cerebral Aneurysm Treatment Assessment". In: *Computer Graphics Forum* 33.3 (June 2014), pp. 131–140. ISSN: 1467-8659. DOI: `10.1111/cgf.12369`. URL: `http://dx.doi.org/10.1111/cgf.12369`.

[27] Roy van Pelt et al. "Exploration of 4D MRI Blood Flow using Stylistic Visualization". In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (Nov. 2010), pp. 1339–1347. ISSN: 2160-9306. DOI: `10.1109/tvcg.2010.153`. URL: `http://dx.doi.org/10.1109/TVCG.2010.153`.

[28] Martin Kaag Rasmussen, Humberto Mestre, and Maiken Nedergaard. "Fluid transport in the brain". In: *Physiological Reviews* 102.2 (Apr. 2022), pp. 1025–1151. ISSN: 1522-1210. DOI: `10.1152/physrev.00031.2020`. URL: `http://dx.doi.org/10.1152/physrev.00031.2020`.

[29] Geir Ringstad, Svein Are Sirirud Vatnehol, and Per Kristian Eide. "Glymphatic MRI in idiopathic normal pressure hydrocephalus". In: *Brain* 140.10 (Aug. 2017), pp. 2691–2705. ISSN: 1460-2156. DOI: `10.1093/brain/awx191`. URL: `http://dx.doi.org/10.1093/brain/awx191`.

[30] Will Schroeder, Ken Martin, and Bill Lorensen. *The Visualization Toolkit (4th ed.)* Kitware, 2006. ISBN: 978-1-930934-19-1.

[31] Partho P. Sengupta et al. "Emerging Trends in CV Flow Visualization". In: *JACC: Cardiovascular Imaging* 5.3 (2012), pp. 305–316. ISSN: 1936-878X. DOI: `https://doi.org/10.1016/j.jcmg.2012.01.003`. URL: `https://www.sciencedirect.com/science/article/pii/S1936878X12000125`.

[32] *Sim4Life*. `https://sim4life.swiss/`. Last accessed: 02-06-2025.

[33] Matthew J. Simon and Jeffrey J. Iliff. "Regulation of cerebrospinal fluid (CSF) flow in neurodegenerative, neurovascular and neuroinflammatory disease". In: *Biochimica et Biophysica Acta (BBA) - Molecular Basis of Disease* 1862.3 (Mar. 2016), pp. 442–451. ISSN: 0925-4439. DOI: `10.1016/j.bbadis.2015.10.014`. URL: `http://dx.doi.org/10.1016/j.bbadis.2015.10.014`.

[34] Gilles Soulat, Patrick McCarthy, and Michael Markl. "4D Flow with MRI". In: *Annual Review of Biomedical Engineering* 22.1 (June 2020), pp. 103–126. ISSN: 1545-4274. DOI: `10.1146/annurev-bioeng-100219-110055`. URL: `http://dx.doi.org/10.1146/annurev-bioeng-100219-110055`.

[35] Vascular Modeling Toolkit. *Computing Centerlines | vmtk*. `http://www.vmtk.org/tutorials/Centerlines.html`. Last accessed: 26-06-2025.

[36] Anna Vilanova et al. "An introduction to visualization of diffusion tensor imaging and its applications". In: *Visualization and processing of tensor fields* (2006), pp. 121–153.

[37] A.V. Vilanova Bartroli et al. "Nonlinear virtual colon unfolding". In: *Proceedings Visualization, 2001. VIS '01.* VISUAL-01. IEEE. DOI: `10.1109/visual.2001.964540`. URL: `http://dx.doi.org/10.1109/VISUAL.2001.964540`.

[38] Tomaž Vrtovec, Boštjan Likar, and Franjo Pernuš. "Automated curved planar reformation of 3D spine images". In: *Physics in Medicine and Biology* 50.19 (Sept. 2005), pp. 4527–4540. ISSN: 1361-6560. DOI: `10.1088/0031-9155/50/19/007`. URL: `http://dx.doi.org/10.1088/0031-9155/50/19/007`.

[39] Qiuting Wen et al. "Assessing pulsatile waveforms of paravascular cerebrospinal fluid dynamics within the glymphatic pathways using dynamic diffusion-weighted imaging (dDWI)". In: *Neuroimage* 260 (2022), p. 119464.

[40] Lulu Xie et al. "Sleep Drives Metabolite Clearance from the Adult Brain". In: *Science* 342.6156 (Oct. 2013), pp. 373–377. ISSN: 1095-9203. DOI: `10.1126/science.1241224`. URL: `http://dx.doi.org/10.1126/science.1241224`.

[41] S. Zachow et al. "Visual Exploration of Nasal Airflow". In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (Nov. 2009), pp. 1407–1414. ISSN: 1077-2626. DOI: `10.1109/tvcg.2009.198`. URL: `http://dx.doi.org/10.1109/TVCG.2009.198`.

A

User study questionnaire

The following pages include the user study guide and the survey described in Chapter 6.

CSF vector visualization survey questionnaire

The tool consists of two modules: VectorVis and VectorStreamlines.

Part 1 - Example case

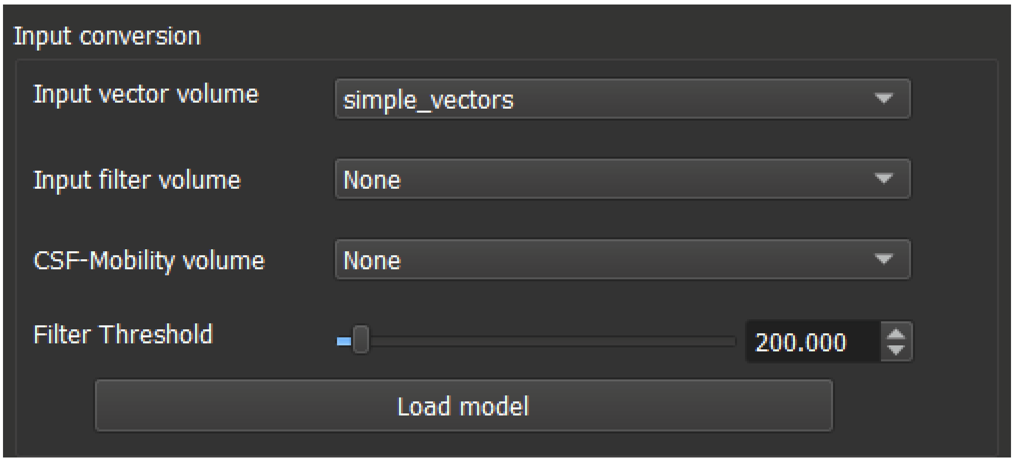
This part serves as a quick and basic tutorial for the visualization tool and workflow.

3D Slicer is loaded with the following dataset:
simple_vectorfield.nrrd → A 3D vector volume.
simple_centerline.mrk.json → A curve representing the centerline of the vessel.

Input

The first step is to convert the vector volume to a model.

It is possible to filter the vector volume using a separate scalar volume, such as a T2 Scan volume. The CSF-Mobility volume can be included here as well. This is not needed in this testing case.



Input vector volume: simple_vectors
Input filter volume: None
CSF-Mobility volume: None
Filter Threshold: 200.000

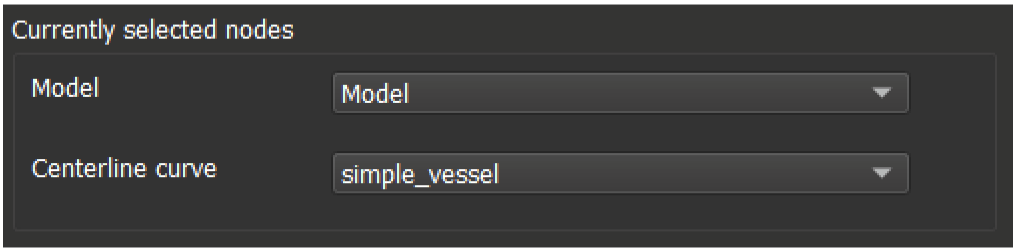
In the ‘Input conversion’ module, select the vector volume in the input section and press ‘Load model’.

Generating glyphs/hedgehog plots

1. In the ‘Currently selected nodes’ section, select the generated Model and select the simple_vessel as the centerline curve

2. Select the ‘Relative Orientation’ coloring mode

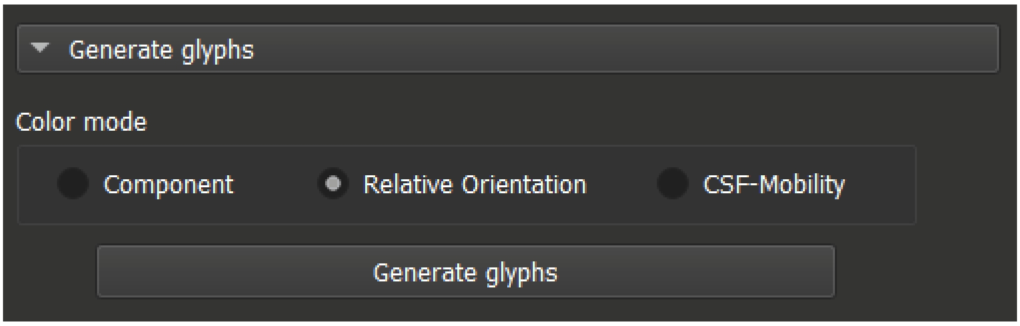
3. Press Generate glyphs’



Currently selected nodes

Model: Model

Centerline curve: simple_vessel



Color mode

○ Component ● Relative Orientation ○ CSF-Mobility

‘Component’: Uses xyz vector components as rgb input.
‘Relative Orientation’: Uses the orientation relative to the centerline curve (dot product).
‘CSF-Mobility’: Uses the CSF-Mobility volume data to color (requires loading with this volume).

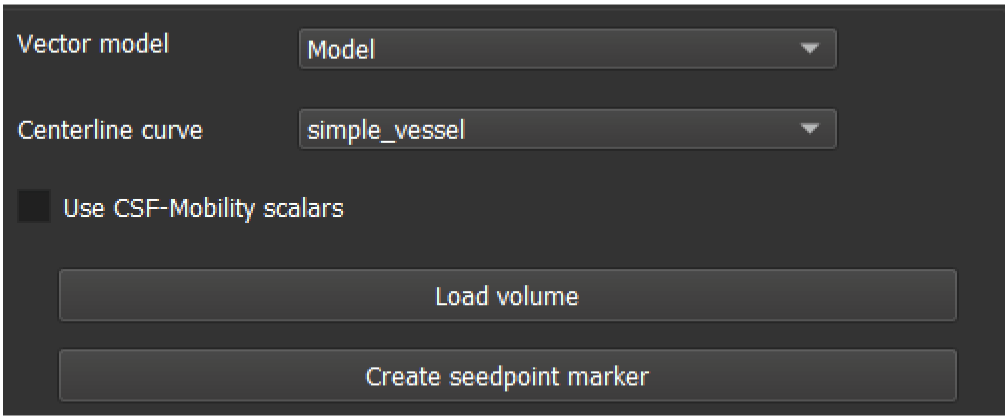
Streamlines

In the VectorStreamLines module.

We first need to load the data from our generated model

1. Select the Model and simple_vessel.

2. Press 'Load volume'.



Vector model: Model

Centerline curve: simple_vessel

☐ Use CSF-Mobility scalars

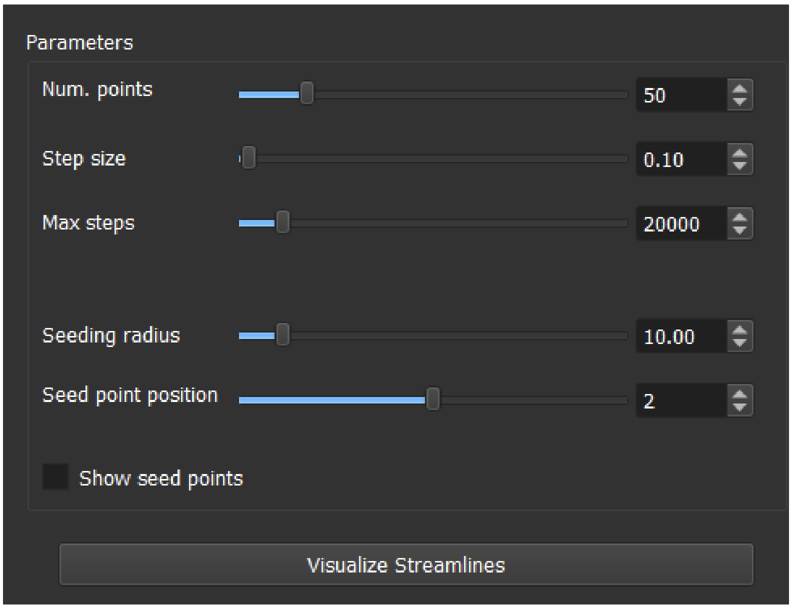
'Use CSF-Mobility scalars': requires loading the Model with the CSF-Mobility volume in the VectorVis module
'Create seedpoint marker': Creates a 'Magic lens' seedpoint marker. Not needed as one is automatically created.

3. You can optionally set some parameters (step size, max steps, etc.)

4. Use the 'Seed point position' slider to move the 'Magic lens' along the centerline

5. Optionally toggle 'Show seed points' to show the random seeding points.

6. Press 'Visualize streamlines'



Parameters

Num. points: 50

Step size: 0.10

Max steps: 20000

Seeding radius: 10.00

Seed point position: 2

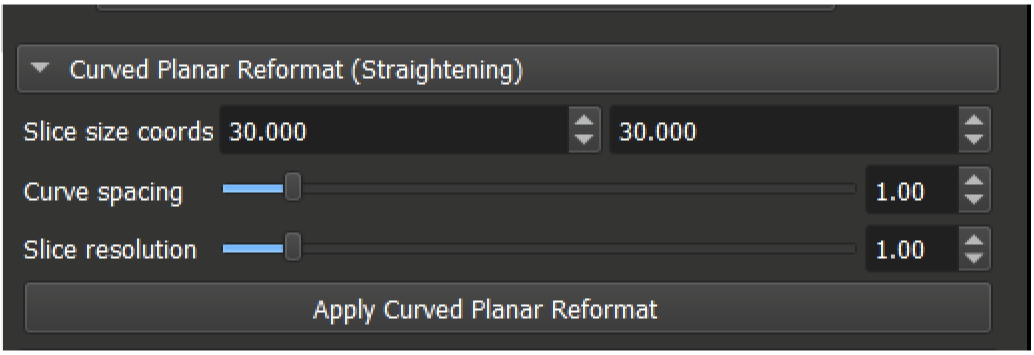
☐ Show seed points

Straightening

Go back to the VectorVis module.

1. With the Model and simple_vessel still as currently selected nodes, simply press ‘Apply Curved Planar Reformat’.

   This will generate a new Transformed Model.



‘Slice size coord’: Defines the area of the slices used in straightening the centerline.
‘Curve spacing’: Defines the resampling along the centerline curve.
‘Slice resolution’: Defines the sampling of the slices.

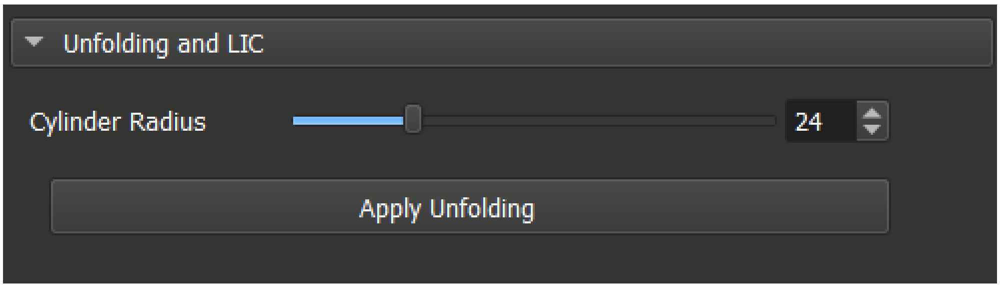
2. You can now generate glyphs or streamlines again by selecting the Transformed model as input in the ‘Currently selected nodes’.

Unfolding

1. With the Transformed Model selected select a radius.
   The radius defines a cylindrical region of interest around the centerline.

2. Press ’Apply Unfolding’. A new Unfolded Model will be created.
   Note: This can take some time.

3. Visualize again using glyphs and Streamlines



Part 2 - Test case and survey

In this part, we will let you explore the visualization methods freely. After doing so, try to answer the following questions to the best of your abilities.

3D Slicer is loaded with the following dataset:
vectorfield.nrrd → A 3D vector volume.
vessel_segment.stl → A cylinder model representing a small vessel.
Centerline curve.mrk.json → A curve representing the centerline of the vessel.
CSF-mobility.nii → CSF mobility

1. Have you used any other vector field visualization tools? (If you have, please elaborate.)

On a scale from 1-5, please answer these questions about the perceived workload.

2. How difficult do you feel it was to use the tool?

1 (very easy) 2 (easy) 3 (neutral) 4 (difficult) 5 (very difficult)

3. How much time do you feel it took you to use the tool?

1 (very little time) 2 (a bit of time) 3 (neutral) 4 (quite some time) 5 (a lot of time)

4. How successful do you feel in using the tool?

1 (Not successful) 2 (a bit successful) 3 (neutral) 4 (quite successful) 5 (very successful)

The following questions are about interpreting the visualization.

5. How would you describe the flow of the CSF relative to the blood vessel/centerline?

   a. It flows parallel to the vessel.
   b. It flows perpendicular to the vessel.
   c. The flow spirals around the vessel.
   d. The flow is chaotic or random
   e. Other / Not sure

6. Did you find any other trends or characteristics of the CSF flow?

7. Which visualization method made it easiest to interpret the flow?

8. Were there any visualization methods that gave confusing results?

The following questions are optional.

9. Did you encounter any problems with the usability of the tool?

10. Do you have any additional remarks?

This is the end of the survey. Thank you for participating.