SIGRAD 2016
M. Hayashi (Editors)

Interactive 4D MRI blood flow exploration and analysis using line predicates

J. Jankowai†[1], R. Englund[1], T. Ropinski[1,2], I. Hotz[1]

[1]Department of Science and Technology, University of Linköping, Sweden
[2]Institute of Media Informatics, Ulm University, Germany

Abstract

We present an interactive exploration tool for 4D PC-MRI blood flow data that incorporates established rendering and filtering methods and combine them into one application. These methods include advanced line illumination, interactively adjustable spatial context visualization and blood flow analysis using line predicates.

Categories and Subject Descriptors (according to ACM CCS): I.3.8 [Computer Graphics]: Applications—

1. Introduction

The presented system is a tool for interactive exploration and analysis of cardiac blood flow including enhanced rendering and filtering- and grouping-methods. According to the WHO's statistic [Wor15], coronary artery disease, or ischaemic heart disease, was the leading cause of death worldwide with a 13% share in 2012. An early diagnosis can help initiating preventive treatment and prognosis in time.

In a healthy heart, the interplay of hemodynamics and cardiac morphology are very well attuned to one another, resulting in an efficient blood transfer from the heart into the body [KYW*00]. In the case of a cardiovasular disease, one of those two factors may change, leading to characteristic changes in flow patterns in the vessels and heart. In return, these alterations can further change the morphology of the heart which can e.g. lead to heavy deformations of the vessels as it is the case in an aneurism. Such unwanted flow patterns include an increased amount of vortices, increased shear forces on the hearts' vessels or blood remaining in a chamber for more than 1 cycle (increased residence time). Visual data exploration can support the detection and evaluation of such flow patterns. Whether or not a certain flow pattern indicates a potential risk depends on factors such as prevalence, position, severity, and size. In addition, an explorative tool such as ours allows for the examination of data from patients that are known to have a certain cardiovascular disease and discover flow characteristics of that disease.

The data utilized in this work was obtained by flow-sensitive **p**hase-**c**ontrast **m**agnetic **r**esonance **i**maging (4D PC-MRI). In this method, vessel morphology information and 3D time-resolved blood flow velocities are measured in-vivo and simultaneously over a set of full cardiac cycles. This process is repeated for approximately 10-20 minutes and the results of all measured cycles averaged [SAG*14].

4D PC-MRI measurements allow for the creation of cardiac blood flow data giving additional insight into the hemodynamics rather than only morphology. This data can be visualized and analyzed in a 3D context using pathlines. A pathline can be thought of as the path a particle takes in a flow field. Commonly, physicians examine blood flow on 2D cross-sections. These 2D cross-sections are either used as projection or as seeding planes for pathlines by some applications. This only allows for examination in a region of interest and can be very time-consuming and arduous.

In practice, it is very cumbersome and inefficient for a physician to look at a plethora of lines obstructing and cluttering one another since relevant flow structures remain hidden in the excess of information. Pathline predicates [SGSM08, SS06, BPMS12] can be used to filter out the noise mentioned above and also extract and group lines with characteristic features. A pathline predicate is a user-defined property or criterion by which a set of pathlines can be classified and filtered (either the pathline fulfills the predicate or not). Then, each pathline group represents a subflow with respect to a certain behaviour of interest. These include

† Master student at University of Linköping, Sweden

flow paths, velocity, vorticity, or residence time. In the case of blood flow analysis, the physician usually wants to combine some of these properties to a more complex query. In addition to creating these queries, the presented system is extensible so that more predicates can be added easily.

Apart from the challenge of semantic filtering our system deals with some technical and perceptional challenges. A general problem when it comes to rendering pathlines is clutter due to the high number of lines. Without proper lighting, this cluttering results in a loss of spatial perceptibility. Many graphics libraries, such as OpenGL, do not provide automatic access to lighting for polylines. To aid this situation and improve spatial perception, a lighting technique for polylines has been implemented.

The software framework used in this project is called **In**teractive **Vi**sualization **Wo**rkshop (**Inviwo**) and is an extendable open-source framework written in C++ for rapid prototyping of interactive applications [InV15].

In summary, the contributions include:

- Straight-forward and computationally cheap anatomical context visualization
- Enhanced spatial perception and decluttering through line illumination and tube representatives
- Blood flow filtering and analysis based on line predicates

2. Related work

With 4D MRI being a young imaging technique it has not yet found its way into mainstream clinical routines. So far, mostly 4D MRI experts have been concerned with visualizing and analysing the obtained data in the domain of research.

Standard flow visualization techniques like streamline and pathlines [BB99, Buo98] as well as colour-coded 2D planes, vector plots or velocity profiles [MCA*03] have been adopted for 4D MRI blood flow analysis. Even though these methods have been advanced [BBLM10, HSU*10, MKE11] they still require much parameter tuning and in-depth user-knowledge in order to find specific flow patterns in the data.

Over the past few years, 4D MRI blood flow visualization has become an increasingly interesting research area. The main objective is to help the user by increasing usability and comprehensiveness of exisiting methods. To aid this situation, van Pelt et al. introduced a virtual probing approach that allows for flow exploration by interactive seed injection onto the flow field and examination of the flow using illustrative rendering and animation [vPBB*11]. These rendering techniques had been presented in previous work and include arrow-trails to depict time-dependent blood flow dynamics and exploded planar reformats to connect 3D and 2D views of the flow. In addition, they proposed a method to simplify the selection of 2D vessel cross-sections [vPBB*10].

Addressing occlusion and clutter when morphology and blood flow behaviour are visualized in the same context, Gasteiger et al. introduce a ghosted view method displaying the vessel surface whilst revealing the underlying blood flow depending on the orientation between viewer and surface [GNKP10]. Neugebauer et al. present an approach to encode a multitude of parameters on the 2D plane isolating the main vessel from an aneurism. In addition, interaction widgets customized to the needs of examining cerebral aneurisms are introduced [NJB*11].

There is an apparent emergence of approaches to improve the depiction of 3D integral lines or flow parameters on 2D planes. This, however, does not solve the problem of interpreting the flow in terms of flow patterns and overall flow behaviour, leaving its solution mostly to the user and making the analysis outcome heavily dependent on the user.

An approach for assisting the user in this task has been proposed by Heiberg et al. where swirling flow is automatically detected using a vector pattern matching technique [HEWK03]. Krishnan et al. visualize blood flow with similar paths by segmenting integral lines starting from a 2D plane based on their anatomical target area. The clusters are displayed on the emitter planes [KGG*12]. Targeting the quantification of blood flow, Eriksson et al. cluster pathlines according to their start and target regions into specific groups and derive the volume of the different compartments from this [ECD*10].

The aforementioned methods deal with one specific flow behaviour while a more flexible approach that allows to structure the flow according to several properties can be useful. Targeting this, Salzbrunn et al. present a line predicate approach for flow analysis [SGSM08, SS06] which is closely related to the pathline attributes introduced by Shi et al. [STH*09], linking views to select pathlines with specific parameters in non-medical data. Born et al. utilize line predicates to sort the set of integral lines capturing the complete flow dynamics into bundles with similar properties [BPMS12]. Our method builds on these ideas.

With respect to the enhancement of perception of line bundles, Banks et al. first proposed a method for lighting line primitives by treating them as infinitesimal cylinders and applying the maximum reflection principle [Ban94]. Mallo et al. later improved on this idea by simplifying the Blinn/Phong model for said cylinders and thereby improving the diffuse reflection model [MPSS05].

Our method for anatomical context visualization is based on the idea of Cipolla et al. defining the contour or silhouette of an object as the set of points on the object where the view direction is orthogonal to the surface normal [CG00].

3. Flow Visualization

In the following we shortly describe the underlying principles used for the flow visualization.

Unsteady velocity fields are defined as: $\mathbf{v} : D \times I \to \mathbb{R}^3$ with $\mathbf{v} \subset \mathbb{R}^3$ being a 3D vector field such that

$$\mathbf{v}(g,\tau) \quad = \quad (v_x(g,\tau), v_y(g,\tau), v_z(g,\tau))^T$$

where $g \in D$, and current time $\tau \in I = [t_0, t_n]$.

Pathlines depict the path of a virtual particle placed into an unsteady flow field at a certain time. It describes the curve $L(g,\tau)$ which is tangent to the vector field everywhere for a point at time τ. This means

$$\dot{L}(g,\tau) \quad = \quad \mathbf{v}(g,\tau) \tag{1}$$

Utilizing this definition, a pathline l passing point a at time τ can be defined as

$$\begin{aligned} l_{a,\tau} : I_{a,\tau} &\rightarrow D, \\ t &\mapsto l_{a,\tau}(t), \\ l_{a,\tau}(\tau) &= a, \\ \frac{\partial l_{a,\tau}}{\partial t}(t) &= \mathbf{v}(l_{a,\tau}(t),t) \end{aligned} \tag{2}$$

where $\mathbf{v}$ denotes the vector field and $I_{a,\tau} \subset I$ the maximal lifespan of the particle in D during I [SGSM08]. The last two terms in Eq. 2 describe an initial value problem to which a pathline can be considered a numerical solution.

Line Predicates are boolean functions that determine whether or not a pathline $l \in \mathcal{P}$ fulfills a certain criterion:

$$\begin{aligned} P : \mathcal{P} &\rightarrow \{\text{true, false}\}, \\ l &\mapsto P(p) \end{aligned}$$

where P is the line predicate and $\mathcal{P}$ represents the set of all integral lines to be examined.
Multiple predicates can be combined logically in order to create more complex queries. Consider a set set of predicates $\mathcal{S}$ containing predicates P_1 and P_2. These can be evaluated successively, connected by a logical operator, as follows:

$$\mathcal{S} = \{P_1 \wedge P_2, P_1 \wedge \overline{P_2}, \overline{P_1} \wedge \overline{P_2}, P_1 \wedge \overline{P_2}\} \tag{3}$$

4. Data

The data set at hand was provided by the **C**enter for **M**edical **I**mage Science and **V**isualization (CMIV) at Linköping University [BPE*15] and contains several discrete volume data sequences of length 41, where each item in the sequence represents one timestep in the cardiac cycle. A complete cardiac cycle in this data set takes about 1036.9 milliseconds, yielding a temporal resolution of 25.291 milliseconds. Of these sequences, two are needed to create the spatial context and the pathlines and are hence the most relevant ones for the application. The first data sequence contains one volume per timestep storing velocity (direction and length) vectors in a uniform grid with dimensions $112 \times 112 \times 48$, the blood flow at a certain point and time. The physical dimensions of the volume are $297.32mm \times 297.31mm \times 131.61mm$, resulting in a spatial resolution (voxel size) of $2.65mm \times 2.65mm \times 2.74mm$. The second relevant data sequence contains a binary volume per timestep with the same dimensions, defining a mask for the whole heart ($0 \mapsto$ outside the heart, $1 \mapsto$ inside the heart). In addition, the data set contains mask sequences for every vessel/chamber of the heart. This can be useful for separate rendering of the vessels/chambers (see Fig. 7) or region-based line filtering.

While the temporal resolution is high, spatial data is sparse in comparison to simulation-obtained data and includes noise, especially outside the heart. The noise outside the heart can be eliminated by a voxel-wise multiplication of the velocity volume with the binary mask. For the elimination of noise for measurements inside the heart more advanced techniques need to be utilized.

5. System

Our system is comprised of several components that were combined in order to provide an interactive tool that meets the requirements for 4D blood flow exploration and analysis. These components include customizable visualization for the anatomical context, line illumination that enhances spatial perceptibility of large sets of lines using colour to emphasise additional properties of the blood flow, and finally, we offer a predicate approach to line filtering and clustering for flow exploration and examination. An interaction interface is linked to a rendering window to support the exploration. A central aspect of the entire system is flexibility, it is easily extendable integrating new properties of interest. The single components of the system and the underlying methods are described below.

5.1. Spatial context visualization

We extract the heart's morphology from the binary mask volume using the marching cubes algorithm. The obtained geometry is then used to render the spatial context for the pathlines in form of a contour using illustrative rendering to improve the spatial perception of the shape of heart (see Fig. 1). We have chosen to use a silhouette and contour enhancement. Similar methods have been used e.g. in [DFRS03]:

Consider the view vector $\mathbf{V}$ from the camera towards the point p to be lit and the normal vector $\mathbf{N}$ of the surface at the fragment to be lit, both normalized and in view space. According to Cipolla and Giblin [CG00], the contour of a surface S consists of the set of those points fulfilling

$$\mathbf{N}(p) \cdot \mathbf{V}(p) = 0 \tag{4}$$

with $p \in S$. This method can be extended by calculating the angle $\theta = \arccos(\mathbf{V} \cdot \mathbf{N})$ between the view vector $\mathbf{V}$ and the surface normal $\mathbf{N}$ and rendering the contour as a gradient. The smaller the angle, the higher $\cos\theta$ will be. To render the contour of the mesh as spatial context we apply $1 - \cos\theta$ as alpha value $c_{\mathbf{a}}$. The contour can then easily be widened or narrowed by applying an adjustable expontent κ to $c_{\mathbf{a}}$

$$c_{\mathbf{a}} \quad = \quad (1 - \cos\theta)^{\kappa} \tag{5}$$

For a more illustrative rendering of the contour we can define an alpha threshold λ for $c_{\mathbf{a}}$ which will dismiss all values for $c_{\mathbf{a}}$ below λ and therefore create a binary contour similar to Cipolla's and Giblin's approach with the difference that here

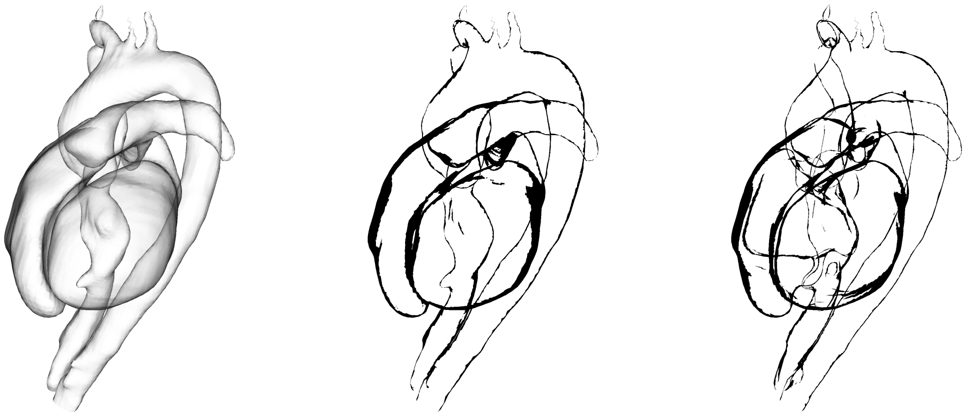
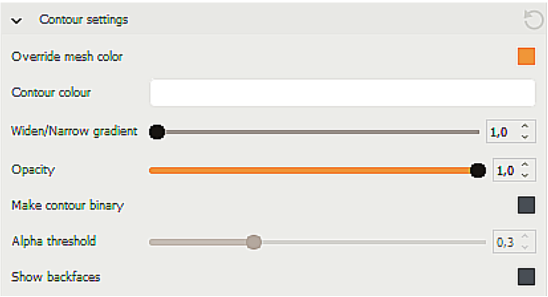


Figure 1: *Different rendering methods for the anatomical context. Left: Contour rendered using the standard method described by Eq. 5 with $\kappa = 1.0$ and full opacity. Center: Contour rendered binary according to Eq. 6 with $\lambda = 0.4$, full opacity and not displaying hidden structures. Right: Contour rendered binary showing occluded structures with $\lambda = 0.3$ and full opacity.*

the thickness of the contour can be controlled through λ:

$$F_{\mathbf{w}}(c_{\mathbf{a}}) = \begin{cases} 1 & \text{if } c_{\mathbf{a}} > \lambda \\ 0 & \text{if } c_{\mathbf{a}} < \lambda \end{cases} \tag{6}$$



Contour settings

Override mesh color

Contour colour

Widen/Narrow gradient 1,0

Opacity 1,0

Make contour binary

Alpha threshold 0,3

Show backfaces

Figure 2: *Settings to control the spatial context rendering: If desired, the user can override the mesh colour. For non-binary contour rendering the exponent κ can be adjusted. For binary rendereing the alpha threshold λ controls the contour thickness. In order to reveal hidden structures, the user can also select to render hidden frontfaces.*

5.2. Line illumination

When rendering a large amount of lines, perceptability of orientation and arragement in space is often reduced. Applying proper lighting to the lines is used to remedy the problem. As one rendering option we have chosen to use illuminated streamlines as proposed by Banks et al. [Ban94].

With $\mathbf{P}$ as the position of the point to be lit, position $\mathbf{C}$ of the camera, and position $\mathbf{S}$ of the light source, we get the view vector $\mathbf{V} = \mathbf{C} - \mathbf{P}$ and the light vector $\mathbf{L} = \mathbf{S} - \mathbf{P}$. According to the Phong's lighting model [Pho75], light intensity $\mathbf{I}$ can be calculated as follows:

$$\mathbf{I} = \mathbf{I}_a + \mathbf{I}_d + \mathbf{I}_s = k_a + k_d \mathbf{L} \cdot \mathbf{N} + k_s (\mathbf{V} \cdot \mathbf{R})^n \tag{7}$$

with $\mathbf{R}$ as the reflection of $\mathbf{L}$ at $\mathbf{N}$, specular exponent n, and k_a, k_d, and k_s as ambient, diffuse, and specular coefficients.

Since a curve in 3D space does not have a uniquely defined normal at point $\mathbf{P}$ it needs to be calculated from the vectors $\mathbf{V}$, $\mathbf{L}$, and tangent $\mathbf{T}$ at $\mathbf{P}$. As the velocity vector defines the tangent of the integral lines the volume can simply be sampled at a desired position and time step in order to obtain the tangent needed for the aforementioned calculation.

A local coordinate frame $(\mathbf{T}, \mathbf{N}, \mathbf{B})$ is calculated from $\mathbf{V}$ and $\mathbf{T}$ [MPSS05], such that

$$\begin{pmatrix} \mathbf{T} \\ \mathbf{N} \\ \mathbf{B} \end{pmatrix} = \begin{pmatrix} \mathbf{T} \\ \mathbf{B} \times \mathbf{T} \\ \mathbf{T} \times \mathbf{V} / \|\mathbf{T} \times \mathbf{V}\| \end{pmatrix}.$$

The ambiguity in the choice of the normal vector is resolved by treating curves as infinitesimal cylinders and choosing the respective surface normal that maximizes the dot products for the specular and diffuse terms in Eqn. 7 [Ban94].

The diffuse reflection is independent of the viewing direction and according to Phong's model calculated by the dot product of the light vector $\mathbf{L}$ and the facet normal $\mathbf{N}_\theta$ with

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_T \\ \mathbf{L}_N \\ \mathbf{L}_B \end{pmatrix} = \begin{pmatrix} \mathbf{L}_T \\ \sqrt{1 - \mathbf{L}_T^2} \cos\alpha \\ \sqrt{1 - \mathbf{L}_T^2} \sin\alpha \end{pmatrix}, \quad \mathbf{N}_\theta = \begin{pmatrix} 0 \\ \cos\theta \\ \sin\theta \end{pmatrix}$$

in $\mathbf{TNB}$ space and with α as the angle between $\mathbf{N}$ and the projection of $\mathbf{L}$ onto $\mathbf{NB}$. It is $\mathbf{L} \cdot \mathbf{N}_\theta = \sqrt{1 - \mathbf{L}_T^2} \cos(\theta - \alpha)$.

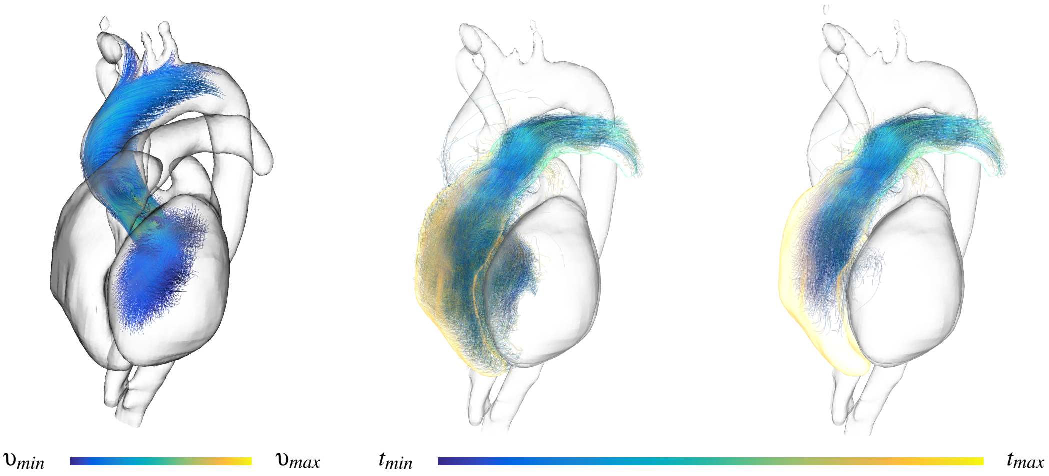


Figure 3: *Left: The lines in this figure start in the left heart chamber and depict the flow durnig systole. Short lines filtered out. Middle: Blood flow over a whole cardiac cycle originating from the right ventricle without applied filtering. The contour has reduced opacity, left ventricle is coloured yellow and pulmonary artery coloured green. Colour-coding by time. Right: Line set from the middle filtered with a ROI predicate, showing only the flow that reaches the pulmonary artery. Same contour and colour-coding.*

With $\alpha = \theta$, at the maximum, the diffuse term becomes:

$$\mathbf{I}_d = k_d\sqrt{1-\mathbf{L}_T^2}.$$

The specular term is given by the dot product of the view vector

$$\mathbf{V} = \begin{pmatrix} \mathbf{V}_T \\ \sqrt{1-\mathbf{V}_T^2} \\ 0 \end{pmatrix}$$

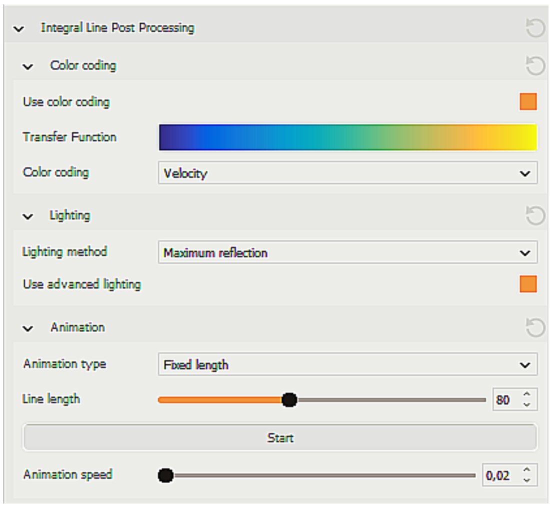
and the reflection vector $\mathbf{R}_\theta$ of $\mathbf{L}$ at $\mathbf{N}_\theta$

$$\begin{aligned} \mathbf{R}_\theta &= -\mathbf{L} + 2(\mathbf{L}\cdot\mathbf{N}_\theta)\mathbf{N}_\theta \\ &= \begin{pmatrix} -\mathbf{L}_T \\ \sqrt{1-\mathbf{L}_T^2}\cos(2\theta-\alpha) \\ \sqrt{1-\mathbf{L}_T^2}\sin(2\theta-\alpha) \end{pmatrix}. \end{aligned}$$

Since the thrid component of $\mathbf{V}$ is zero, the maximum for the specular reflection is reached at $\theta = \alpha/2$ which gives

$$\mathbf{I}_s = k_s\left(-\mathbf{V}_T\mathbf{L}_T + \sqrt{1-\mathbf{V}_T^2}\sqrt{1-\mathbf{L}_T^2}\right)^n$$

for the calculation of the specular term with n as the specular exponent.



Integral Line Post Processing

Color coding

Use color coding ■

Transfer Function

Color coding: Velocity

Lighting

Lighting method: Maximum reflection

Use advanced lighting ■

Animation

Animation type: Fixed length

Line length: 80

Start

Animation speed: 0,02

Figure 4: *Setting to control line illumination: If desired, the user can colour-code the pathlines according to an adjustable transfer function applied to selected line properties. Advanced lighting can be enabled/disabled, the lighting model can be chosen and pathlines can be animated, see 5.4.*

5.3. Line predicates based filtering

We use line predicates as central means for the filtering of the pathlines. We have exemplarily implemented some basic predicates. However, this set can be easily extended if required. The implemented predicates depend on line parameters, such as line length, maximum velocity and mean velocity. Vorticity $\omega = \nabla\times\mathbf{v}$ is related to turbulence in a flow field and therefore of special interest for flow analysis. The **R**egion **o**f **I**nterest (ROI) predicate checks if a line passes through a specified region. The presented parameters for am integral line l with velocity $\upsilon_i(l)$ at vertex $x_i \in \{0,\ldots,n\}$ at time t_i are defined below along with their respective predi-

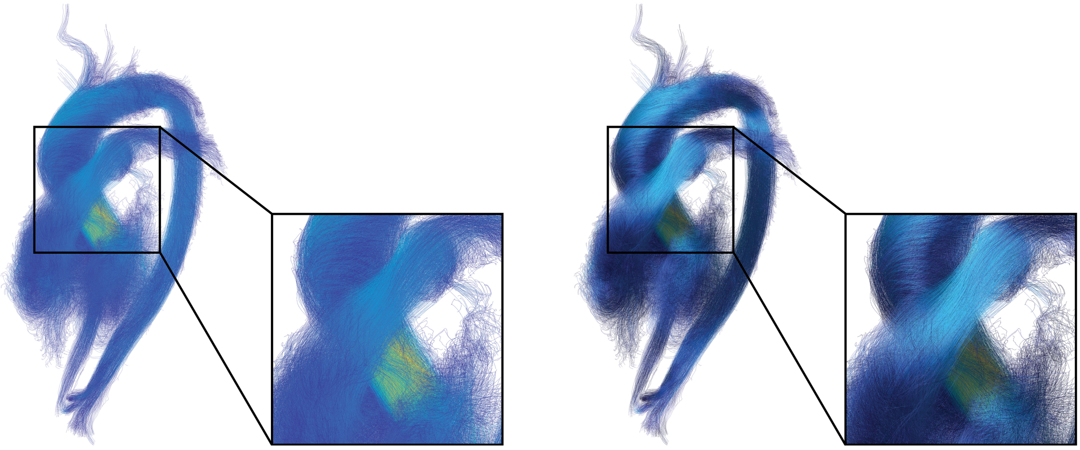


Figure 5: *Comparison of a line set directly rendered as line primitives by OpenGL and therefore without lighting (left) and the same set of lines rendered with applied lighting according to Sec. 5.1 (right)*

cate.

Length. The length of line l during time span $[t_1, t_2]$ is calculated as follows:

$$len(l) = \sum_{i=1}^{n} \|x_i(l) - x_{i-1}(l)\|$$

Mean velocity. The length of a line is direclty related to the mean velocity which is defined as the average velocity of l during time span $[t_1, t_2]$.

$$\upsilon_{mean}(l) = \frac{len(l)}{t_1 - t_2}$$

Maximum velocity. During the time span $[t_0, t_{n-1}]$ the maximum velocity of a line is defined as:

$$\upsilon_{max}(l) = max(\upsilon_0(l, t_0), ..., \upsilon_{n-1}(l, t_{n-1}))$$

If we set one of these functions as $f(l)$ we can formulate a general evaluation function, or predicate, $P(l)$

$$P(l) = \begin{cases} true & \text{if } f(l) \in [f_1, f_2] \text{ during } [t_1, t_2] \\ false & \text{otherwise} \end{cases}.$$

Basically, $P(l)$ will evaluate to true if a line meets the user-defined criterion $f(l)$ within $[t_1, t_2]$.

Region of interest. The ROI predicate evaluates if a pathline l runs through a region, or set of points, $R = [x_0, x_n]$ at t_i:

$$P(l) = \begin{cases} true & \text{if } x \in l(t_i) \text{ and } x \in R \\ false & \text{otherwise} \end{cases}.$$

To provide an easy way to define new queries the system in cludes a parser that enables logical combination of predicates in order to create more complex queries. This could be queries like “Which lines reach a high velocity but have a short length?” or “Which lines reach a high vorticity during the systole and either have a low mean velocity or do not reach the aorta?”. The interface for the filtering consists of a pulldown menu for the available predicates and a second pulldown menu indicating the logical combination between two predicates (see Fig. 6). The options `STRONG AND` and `STRONG OR` were added so that a hierachy can be established that would be represented by brackets in a logical equation, for example

$$\mathcal{S} = (P_1 \wedge P_2) \vee P_3 \qquad (8)$$

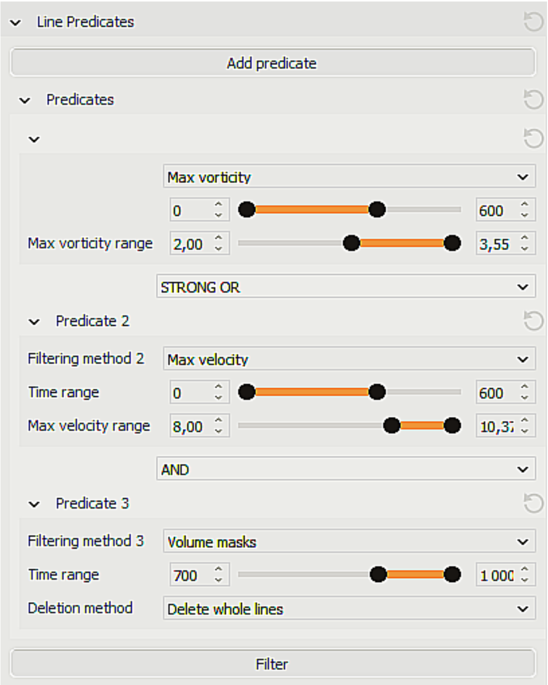
In our application, the logical operator between P_1 and P_2 would be a `STRONG AND` followed by an `OR` operator between P_2 and P_3.

5.4. Implementation Details

The implementation and data flow is designed to find a balance between complexity and interactivity. Therefore, the system can be divided into two parts: Relatively time-consuming pre-calculations performed on the CPU and interactive rendering on the GPU.

The pathline integration is done on the CPU using either the Euler, Heun, or 4th-order Runge-Kutta method. Adjustable parameters available in this process are the temporal step size, the number of steps forward and backwards, and the integration method. Also performed on the CPU is the pathline filtering using line predicates. Once the pathlines have been calculated the user can define the desired predicates and initiate the filtering (see Fig. 6).

Lighting, colour-coding and pathline animation are performed interactively using shaders programmed in GLSL. As shown in Fig. 4, the user can define the transfer function and set the colour-coding to velocity, time, or vorticity. Maximum values for velocity, length and vorticity are available per line on the GPU and used to ensure that the full range of the transfer function is used ideally.



Max vorticity
0 – 600
Max vorticity range 2,00 – 3,55
STRONG OR
Filtering method 2 Max velocity
Time range 0 – 600
Max velocity range 8,00 – 10,37
AND
Filtering method 3 Volume masks
Time range 700 – 1 000
Deletion method Delete whole lines

Figure 6: *Settings available for the line predicates. The user can select predicates from a pulldown menu, their bounding values, evaluation time interval, and the logical combination between two predicates from a second pulldown menu. The options* `STRONG AND` *and* `STRONG OR` *were added so that a hierarchy can be established that would be represented by brackets in a logical equation (see Eq. 8).*

Advanced lighting of the pathlines can be turned off if more performance is needed or be set to the maximum reflection principle or the cylinder averaging approach presented in [Ban94]. Animation of pathlines can be done either in an evolving manner, meaning that the pathline will be drawn successively, or as pathlets, meaning short lines with a fixed line length, moving as a kind of wave along the pathlines. Indirectly, the flow can be shown as a particle stream if a very short length is set for the pathlets.

Adjustments to lighting, colour-coding, or animation are directly visible in the rendering at interactive framerates.

6. Results and Conclusion

The presented exploratory system was developed as a module for InViWo using the C++ programming language, supported by programmable shaders implemented in GLSL. The exploratory purpose of the system heavily depends on the accomplished interactivity. The GPU is used for the majority of the visualization methods and related calculations, allowing real-time interaction and customisation.

We have examplarily evaluated calculation times for pathline generation and line filtering for one 4D MRI blood-flow data set. The calculation time for pathline integration depends on the step size, number of steps, integration method and number of seedpoints. The generation of approximately 13,000 pathlines for a full cardiac cycle with a step size of 0,001 (ergo 1000 steps) took about 33.4 seconds using Euler's method and 61.6 seconds using the 4th-order Runge-Kutta integration scheme. We have then applied a combination of two and three predicates to this line set. Filtering took 20.7 seconds and 23.3 seconds, respectively.

Different options for the context rendering can be seen in Fig. 1. Fig. 7 depicts different modes for line rendering (tubes and illuminated lines). The illuminated lines are more appropriate for dense line representation, while the tubes can be used to show flow structures more representatively. An example for filtering using the length predicate can be seen in Fig. 3 as well as different colour-coding examples. Additionally, comparing the middle and left image in Fig. 3 it can be observed that not all blood from the left ventricle reaches the pulmonary artery within one cardiac cycle. This may be indicative of a heart muscle insufficiency.

In conclusion, we present a system that provides interactive 4D blood flow visualization, exploration and analysis. In particular, we offer an application that allows the user to filter out and search for flow structues of interest and helps the user to make sense of the data in a spatio-temporal context through depth-cues, anatomical context and animation.

References

[Ban94] Banks D. C.: Illumination in Diverse Codimensions. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1994), SIGGRAPH '94, ACM, pp. 327–334. 2, 4, 7

[BB99] Bogren H. G., Buonocore M. H.: 4D magnetic resonance velocity mapping of blood flow patterns in the aorta in young vs. elderly normal subjects. *Journal of Magnetic Resonance Imaging 10*, 5 (1999), 861–869. 2

[BBLM10] Barker A., Bock J., Lorenz R., Markl M.: 4D flow MR imaging. *MAGNETOM Flash* (2010), 46–52. 2

[BPE*15] Bustamante M., Petersson S., Eriksson J., Alehagen U., Dyverfeldt P., Carlhall C. J., Ebbers T.: Atlas-based analysis of 4D flow CMR: automated vessel segmentation and flow quantification. *J Cardiovasc Magn Reson 17* (2015), 87. 3

[BPMS12] Born S., Pfeifle M., Markl M., Scheuermann G.: Visual 4D MRI blood flow analysis with line predicates. In *Pacific Visualization Symposium (PacificVis), 2012 IEEE* (Feb 2012), pp. 105–112. 1, 2

[Buo98] Buonocore M. H.: Visualizing blood flow patterns using streamlines, arrows, and particle paths. *Magn Reson Med 40*, 2 (Aug 1998), 210–226. 2

[CG00] Cipolla R., Giblin P.: *Visual Motion of Curves and Surfaces*. Cambridge University Press, New York, NY, USA, 2000. 2, 3

[DFRS03] DeCarlo D., Finkelstein A., Rusinkiewicz S., Santella A.: Suggestive Contours for Conveying Shape. *ACM Transactions on Graphics (Proc. SIGGRAPH) 22*, 3 (July 2003), 848–855. 3

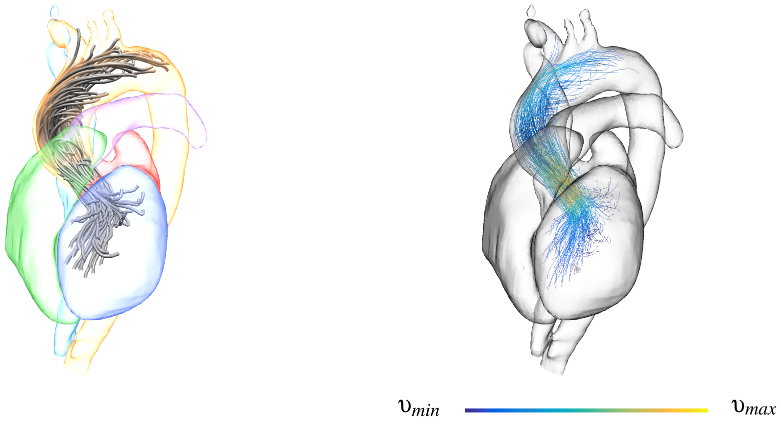


Figure 7: *Examples of different rendering and colouring options for the spatial context as well as the lines. The lines in this figure start in the left heart chamber and depict the flow durnig systole. Left: Each chamber and vessel is assigned a user-defined colour for better distinguishability of the heart structure. Lines are rendered as tubes to represent basic flow structures. Right: Anatomical context rendered rather subtly as black silhouette. Flow is represented by illuminated lines and colour-coded according to velocity.*

[ECD*10] Eriksson J., Carlhall C. J., Dyverfeldt P., Engvall J., Bolger A. F., Ebbers T.: Semi-automatic quantification of 4D left ventricular blood flow. *J Cardiovasc Magn Reson 12* (2010), 9. 2

[GNKP10] Gasteiger R., Neugebauer M., Kubisch C., Preim B.: Adapted Surface Visualization of Cerebral Aneurysms with Embedded Blood Flow Information. In *Eurographics Workshop on Visual Computing for Biology and Medicine* (2010), Bartz D., Botha C., Hornegger J., Machiraju R., Wiebel A., Preim B., (Eds.), The Eurographics Association. 2

[HEWK03] Heiberg E., Ebbers T., Wigstrom L., Karlsson M.: Three-dimensional flow characterization using vector pattern matching. *IEEE Transactions on Visualization and Computer Graphics 9*, 3 (July 2003), 313–319. 2

[HSU*10] Heiberg E., Sjogren J., Ugander M., Carlsson M., Engblom H., Arheden H.: Design and validation of Segment–freely available software for cardiovascular image analysis. *BMC Med Imaging 10* (2010), 1. 2

[InV15] InViWo: InViWo Overview, 2015. 2

[KGG*12] Krishnan H., Garth C., Guhring J., Gulsun M. A., Greiser A., Joy K. I.: Analysis of time-dependent flow-sensitive PC-MRI data. *IEEE Trans Vis Comput Graph 18*, 6 (Jun 2012), 966–977. 2

[KYW*00] Kilner P. J., Yang G.-Z., Wilkes A. J., Mohiaddin R. H., Firmin D. N., Yacoub M. H.: Asymmetric redirection of flow through the heart. *Nature 404*, 6779 (Apr 2000), 759–761. 1

[MCA*03] Markl M., Chan F. P., Alley M. T., Wedding K. L., Draney M. T., Elkins C. J., Parker D. W., Wicker R., Taylor C. A., Herfkens R. J., Pelc N. J.: Time-resolved three-dimensional phase-contrast MRI. *J Magn Reson Imaging 17*, 4 (Apr 2003), 499–506. 2

[MKE11] Markl M., Kilner P. J., Ebbers T.: Comprehensive 4D velocity mapping of the heart and great vessels by cardiovascular magnetic resonance. *Journal of Cardiovascular Magnetic Resonance 13*, 1 (2011), 1–22. 2

[MPSS05] Mallo O., Peikert R., Sigg C., Sadlo F.: Illuminated lines revisited. In *Visualization, 2005. VIS 05. IEEE* (Oct 2005), pp. 19–26. 2, 4

[NJB*11] Neugebauer M., Janiga G., Beuing O., Skalej M., Preim B.: Anatomy-Guided Multi-Level Exploration of Blood Flow in Cerebral Aneurysms. *Computer Graphics Forum 30*, 3 (2011), 1041–1050. 2

[Pho75] Phong B. T.: Illumination for Computer Generated Pictures. *Commun. ACM 18*, 6 (June 1975), 311–317. 4

[SAG*14] Stankovic Z., Allen B. D., Garcia J., Jarvis K. B., Markl M.: 4D flow imaging with MRI. *Cardiovascular Diagnosis and Therapy 4*, 2 (2014). 1

[SGSM08] Salzbrunn T., Garth C., Scheuermann G., Meyer J.: Pathline predicates and unsteady flow structures. *The Visual Computer 24*, 12 (2008), 1039–1051. 1, 2, 3

[SS06] Salzbrunn T., Scheuermann G.: Streamline Predicates. *IEEE Transactions on Visualization and Computer Graphics 12*, 6 (Nov 2006), 1601–1612. 1, 2

[STH*09] Shi K., Theisel H., Hauser H., Weinkauf T., Matkovic K., Hege H.-C., Seidel H.-P.: *Topology-Based Methods in Visualization II*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, ch. Path Line Attributes - an Information Visualization Approach to Analyzing the Dynamic Behavior of 3D Time-Dependent Flow Fields, pp. 75–88. 2

[vPBB*10] van Pelt R., Bescos J. O., Breeuwer M., Clough R. E., Groller M. E., ter Haar Romenij B., Vilanova A.: Exploration of 4D MRI Blood Flow using Stylistic Visualization. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (Nov 2010), 1339–1347. 2

[vPBB*11] van Pelt R., Bescos J. O., Breeuwer M., Clough R. E., Groller M. E., ter Haar Romenij B., Vilanova A.: Interactive Virtual Probing of 4D MRI Blood-Flow. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (Dec 2011), 2153–2162. 2

[Wor15] World Health Organization: Global Health Observatory visualizations, Causes of death - Ten leading causes of death, 2012, 2015. 1