Segmentation of cardiovascular tree in 4D from PC-MRI images

Peter Öberg

April 8, 2013

Abstract

Investigating the dynamics of blood flow is essential to understanding the cardiovascular system and its pathologies. Phase Contrast Magnetic Resonance Imaging (PC-MRI) enables acquisition of 4-dimensional time resolved velocity experiments. Slice-by-slice analysis of the data is time consuming and cumbersome. In this master thesis a simple but effective algorithm for fully automatic visualization of the entire cardiovascular tree based on 4D PC-MRI velocity data. The achieved visualization portrays the cardiovascular vessel tree.The purposed segmentation algorithm is implemented in MATLAB and combined with the existing software Segment and Fourflow for visualization. Clinical data shows very promising visualization of the cardiovascular tree.

Contents

1 Introduction 5

2 Background 7

2.1 Magnetic Resonance Imaging . . . 7
2.2 Phase contrast MRI . . . 9
2.3 The Human heart . . . 12
2.3.1 Systole and diastole . . . 13
2.4 Vessel segmentation . . . 14
2.5 Segment . . . 14
2.6 Fourflow . . . 15

3 Aim 17

4 Method 19

4.1 Notations . . . 19
4.2 Definition . . . 20
4.3 Local features . . . 20
4.3.1 Local Deviation . . . 20
4.3.2 3D flow vector analysis . . . 20
4.4 Implementation of final algorithm . . . 23

5 Results 27

5.1 Features . . . 27
5.1.1 Local deviation . . . 27
5.1.2 Angular spread . . . 30
5.1.3 Tensor . . . 31
5.1.4 Projected velocity magnitude . . . 32
5.2 Final algorithm . . . 33
5.3 Validation . . . 35
5.3.1 Time resolution . . . 35
5.3.2 Isosurface values . . . 36
5.3.3 Previous method . . . 37
5.3.4 Applications . . . 38

6 Discussion and Conclusions 41

6.1 Discussion . . . 41
6.1.1 Different features . . . 41
6.1.2 Accuracy of final visualization . . . 42

6.1.3 Further improvement 42
6.2 Conclusion 43

Chapter 1

Introduction

Over the years many different forms of medical imaging have been developed; Ultrasound, Computed Tomopgraphy (CT) and Magnetic Resonance Imaging (MRI). All of these are capable of presenting tremendous amount of information. Finding internal hemorrhaging, infarction or a life threatening aneurysm can now be done within minutes completely noninvasive without risking harm to the patient.

L Wigström [1] introduced three-dimensional Phase Contrast MRI (PC-MRI) with which time resolved, three dimensional velocity data can be acquired. The acquisition technique of such data is relatively established, but analysis of the data in order to use it for diagnostics or research is still inadequate.

Manual segmentation is time consuming and operator dependent. Therefore, introducing automatic segmentation in images can be of great clinical use to achieve consistent and complete assessment of the data. When studying flow images there is a need to display the cardiovascular tree and flow simultaneously. Automatic visualization of the entire cardiovascular tree from flow data would be of great value.

Previous methods [2] have focused on segmentation of single vessels and have not made full use of the characteristics of cardiovascular flow. Therefore, the aim of this thesis is to attempt to visualize the entire cardiovascular tree using flow properties in 4D PC-MRI flow data.

In this thesis a straightforward algorithm to create a three dimensional visualization of the cardiovascular tree using flow properties of velocity magnitude and direction in PC-MRI flow images is developed.

Chapter 2

Background

2.1 Magnetic Resonance Imaging

In 1952, Felix Bloch and Edward M. Purcell [3] were awarded the Nobel Prize in Physics for their discovery of the nuclear magnetic resonance phenomenon. The fact that nuclei possess a spin and an angular momentum was later used as a source for image construction [4]. This angular momentum creates a magnetic moment when affected by an external magnetic field, B_o, see Figure 2.1. The angular velocity ω_0 is given by [5]

$$\omega_0 = \gamma B_0 \tag{2.1}$$

where γ is the gyromatic ratio for the nucleus and B_0 is the external field. The frequency at which the nuclei precess is called the Larmour frequency.

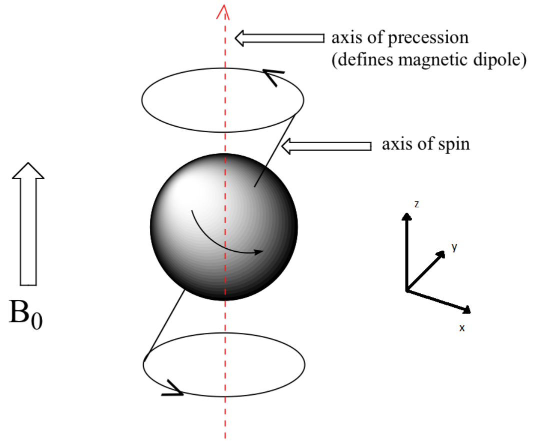


Figure 2.1: A proton precessing around the axis of precession. Adopted from [6]

The nuclei align themselves with the external magnetic field B_0. Parallel alignment takes less energy than anti parallel, thus slightly more nuclei choose the parallel state. On a macroscopic scale the summation of all the magnetic moments within the volume yields a net magnetization vector M_z in the same direction as B_0, usually referred to as the z-axis.

As the nuclei have a specific Larmor frequency we can excite them with a radio-frequency (RF) signal. If the RF signal frequency coincides with the Larmor frequency of the nuclei they will absorb energy and their magnetization vector will flip 90 (degrees) to the transverse xy-plane. This results in a reduction of the longitudinal magnetization M_z and the occurrence of a transversal magnetization M_{xy} as seen in Figure 2.2.

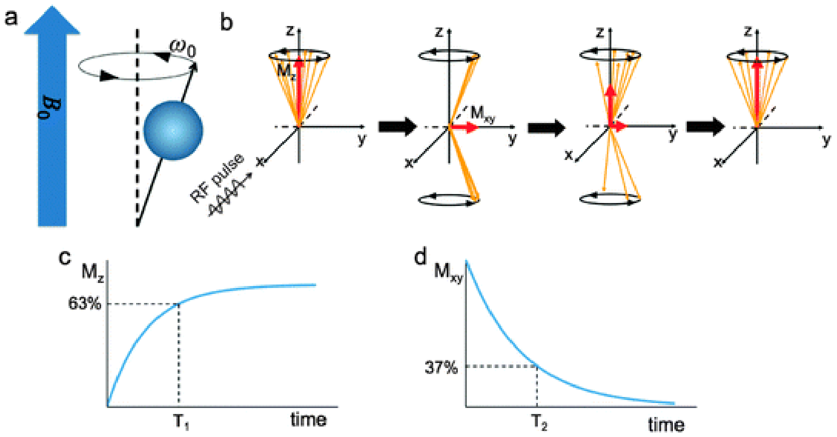


Figure 2.2: The RF pulse flips the longitudal magnetization M_z and creates a transversal magnetization M_{xy}. [7]

When the RF signal is turned off the longitudinal magnetization M_z starts to go back to its initial value and M_{xy} fades away due to the nuclei realigning themselves to B_0. The time it takes for M_z to return to 63% of its original value is called T_1, while the time for M_{xy} to dissipate to 37% of its largest value is called T_2. During this realignment process, called the relaxation period, a RF signal is sent out which induces a signal in a coil within the MRI machine which is the source MR signal which yields the actual images.

T_1 and T_2 are tissue specific and alteration in the imaging process can be used to adjust the contrast of the images acquired. This possibility of adjusted contrast in order to highlight certain types of tissue preprocessing is one of the strengths with MRI.

In order to record the released MR signal from the nuclei the exact origin of this particular signal needs to be matched with a location within the volume. This is done by introducing three gradient fields; G_z, G_y, G_x. The first of these gradients applies a linear gradient along the G_z-axis. The result of this is that a certain position along the z-axis will correspond to a certain Larmor frequency, essentially this is the slice selection. G_y, called the phase encoding gradient, desynch the phase of the nuclei within the slice so that each position along the y-axis corresponds to a certain phase. Finally G_x, called the frequency encoding gradient, yields a certain frequency return of the signal for each step in x-direction. This is illustrated in Figure 2.3.

1. The RF signal is sent while the G_z gradient selects the slice.

2. G_y encodes the phase within the slice.
3. G_x encodes the frequency within the slice and the received signal is recorded.

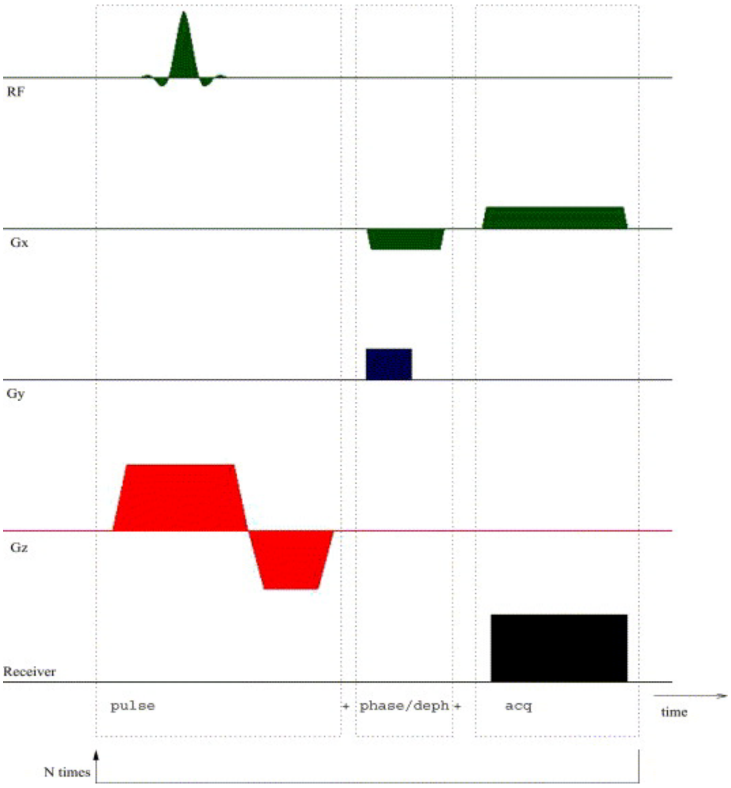


Figure 2.3: The most common pulse sequence for MRI-imaging [8]

With the use of these gradients each voxel is bound to a certain phase and frequency of the received signal. The received RF signal is the inverse Fourier transformed into a complex number. These complex numbers in each voxel represent the image, see Figure 2.4. Notice in this image the same four chambers of the heart illustrated by Figure 2.6. In general the modulus is used for imaging but the phase of the image can also be used, particularly to observe flow within the volume.

2.2 Phase contrast MRI

A uniform motion of protons along a magnetic field gradient results in a change of the MR signal phase, ϕ. This change is proportional to the velocity of the tissue, v. Using this phase shift it is possible to construct an image detailing the velocity in any specified direction and slice. The phase of the received signal from a single voxel is given by [5]:

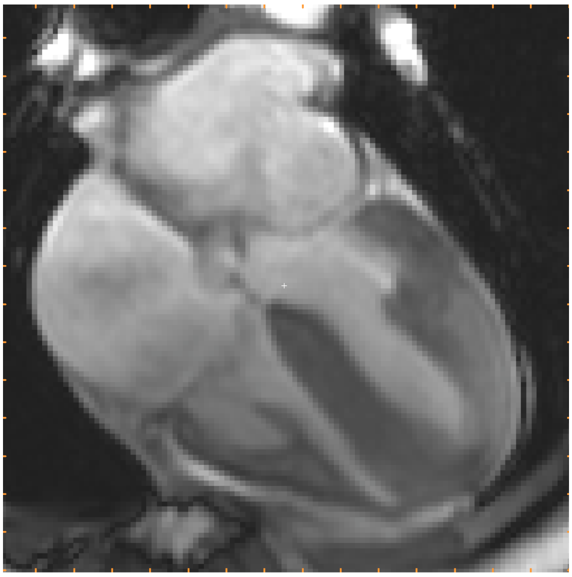


Figure 2.4: The most common image representation is the magnitude image in the through plane direction. This is the 4 chamber view of the heart.

$$\phi(r,T) = \gamma B_0 T + \gamma v \cdot \int_0^T \mathbf{G}(r,t) t dt \tag{2.2}$$

$$= \gamma B_0 T + \gamma v \cdot \overline{\mathbf{G}} \tag{2.3}$$

where γ is the gyromagnetic ratio, T is the time, B_0 is the external magnetic field and $\mathbf{G}(r,t)$ is the magnetic field gradient. By applying gradient pairs, which sequentially dephase and rephase, two datasets with different phases depending on how far the tissue has moved during the recording are acquired. By subtracting these phases

$$\phi_1 - \phi_2 = \gamma v \cdot (\overline{\mathbf{G}}_1 - \overline{\mathbf{G}}_2) \tag{2.4}$$

the velocity along the $(\overline{\mathbf{G}}_1 - \overline{\mathbf{G}}_2)$-gradient direction is attained. The gradient pairs are usually applied to produce velocity information in the x-, y- and z-direction. This way the 3D velocity for each individual voxel in one slice is recorded simultaneously.

The collected data from PC-MRI scans presents us with a multidimensional data set; for each slice in the volume there are three different phase images, one for each gradient direction, where the amplitude of the image shows the velocity. See Figure 2.5. Furthermore, these images are time resolved and thus we have a fourth dimension, 3D + T. The size of these type of dataset is large and handling them can be cumbersome. This is visualized by Figure 2.5.

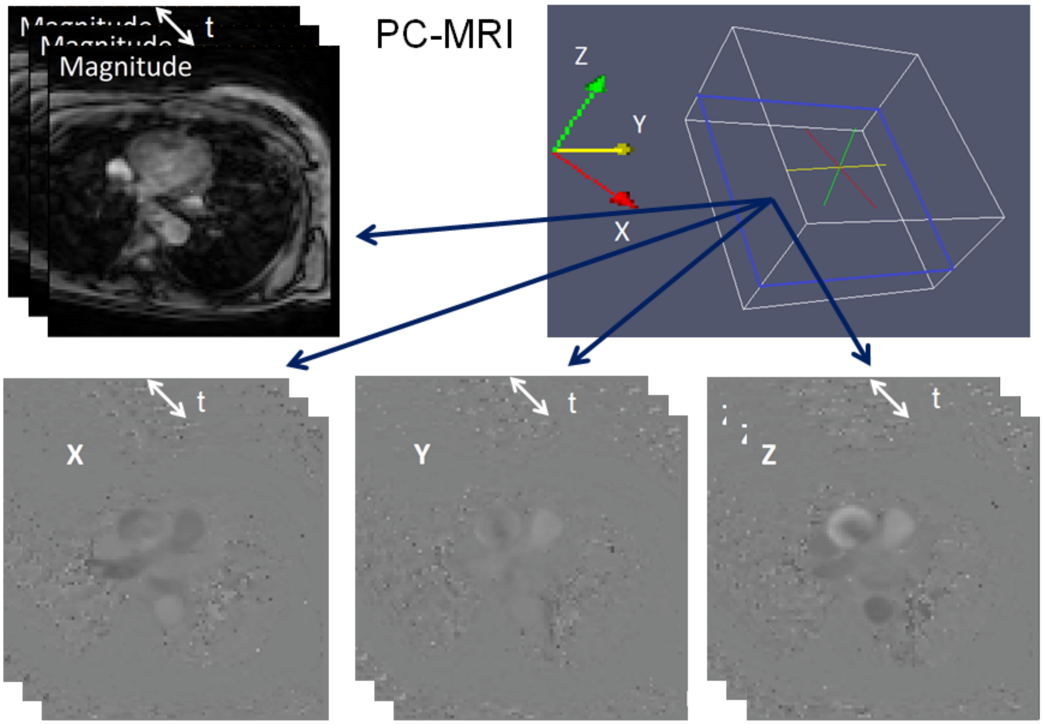


Figure 2.5: Schematic image of PC-MRI data. Each slice within the volume contains four time resolved images: The magnitude image and one phase image for each direction.

2.3 The Human heart

At the centre of our circulatory system sits the human heart. This self regulating biological pump is a wonder of nature. It is divided into four separate chambers; left atrium (LA), right atrium (RA), left ventricle (LV) and right ventricle (RV). Between the atria and ventricles are valves. These valves allow blood to flow in one direction without flowing back. The mitral valve separates LA from LV and the tricuspid valves the right side. The pulmonary valve controls the outlet of blood from RV to the lungs and the aortic valve the outlet from the LV to the body. A schematic view of the heart is displayed in Figure 2.6.

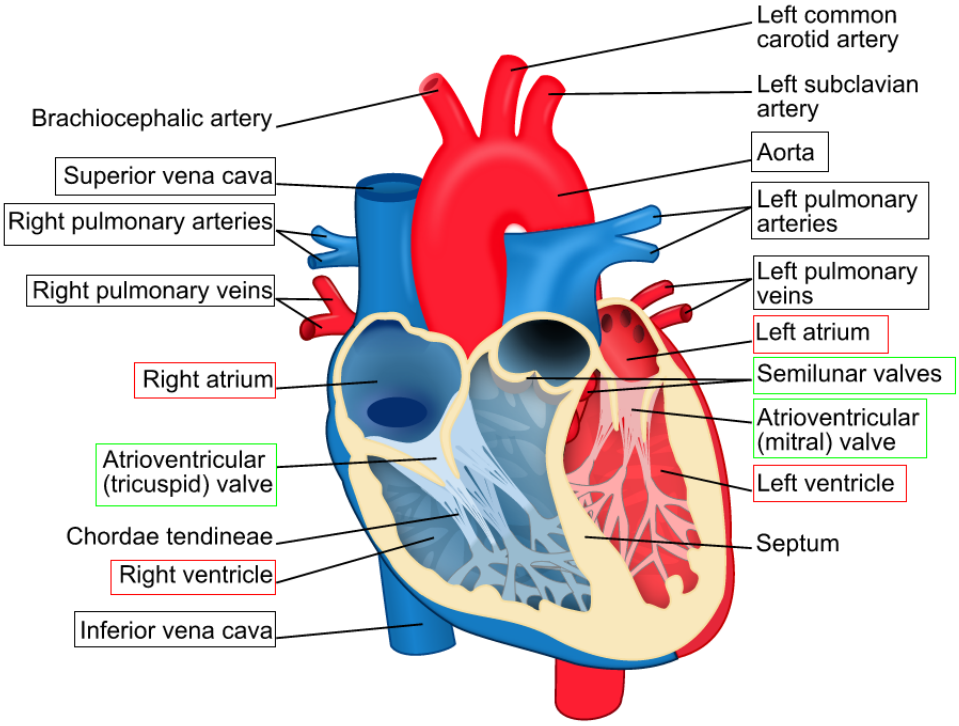


Figure 2.6: The human heart. Black boxes: Vessels visualized by algorithm. Red Boxes: Atria and ventricles. Green boxes: Valves. Adopted from [9]

The compression of the heart leads to flow in the blood vessels throughout the body. The flow follows the same pattern each cycle and can be broken down into a few steps. Starting in the right atrium:

1. Deoxygenated blood flows in from the body into the RA.
2. The blood flows into the RV through the tricuspid valve.
3. The heart contracts and the blood is pumped into the lungs where it deposits carbon dioxide and absorbs oxygen.
4. The oxygenated blood flows into the LA through the pulmonary valves.
5. The blood flows down into the LV through the mitral valve.

6. Finally the oxygenated blood is pumped out to the body.
7. Back too step 1.

2.3.1 Systole and diastole

During systole the ventricles in the heart contract, pumping blood in left and right ventricles out into the body and lungs. The relaxing part of the cardiovascular cycle is called diastole where blood flows into the left and right atria and ventricles. In general in the cardiovascular system, the velocities are high during systole and low during diastole. An important assumption in this thesis is the fact that during systole the flow is high within the blood vessels compared to surrounding tissue, while the flow during diastole is similar in the blood vessels and tissue. This means that distinguishing whether or not an arbitrary flow in a certain point is within a blood vessel or tissue is difficult during diastole. In this thesis flow information during systole is used to achieve a 3D visualization of the cardiovascular tree. Important to note is that this means that visualization will not be a 100 % correct representation of the cardiovascular tree during a certain point of the cardiovascular cycle. In reality the vessels differ both in size and position during the heartbeat, [10]. For the sake of calculation the duration of systole is approximated to 300 ms. This is more or less true in the case of increased heart rate, in which case the diastole is shortened. Furthermore 4D flow cardiac MRI is performed at rest. Figure 2.7 shows measured flow within the aorta accompanied by division of the cardiac cycle into systole and diastole.

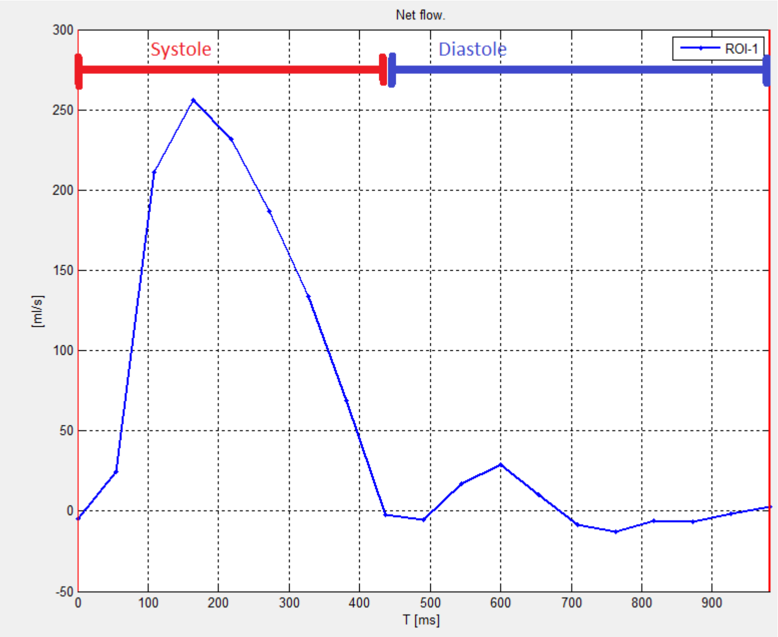


Figure 2.7: Measured flow in aorta ascending shows the dynamics of flow in blood vessels during the heartbeat.

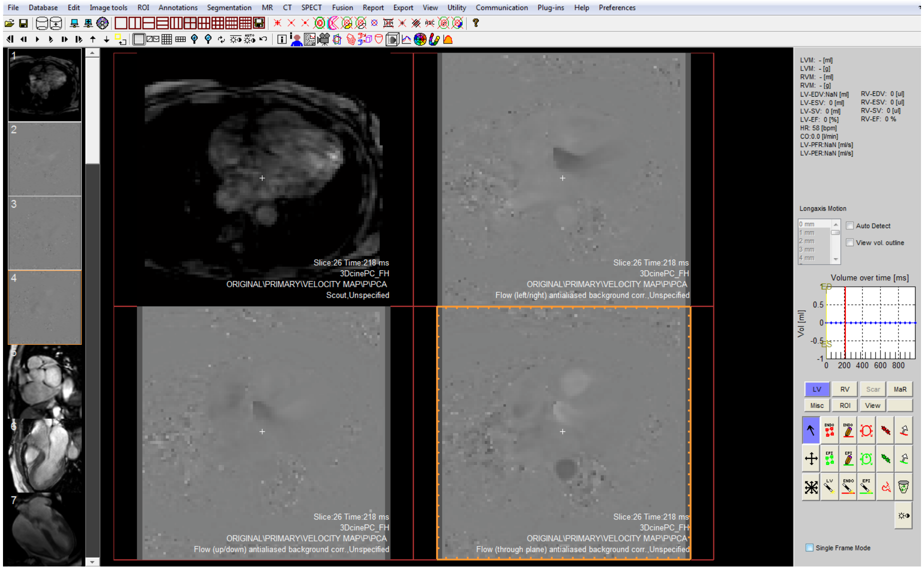
2.4 Vessel segmentation

There is a need for automatic segmentation of blood vessels from flow data. This segmentation can be useful to analyse the data. There are many different methods presented for segmentation of blood vessels in PC-MRI data. Level set functions [2], snakes and active contours [11] have been proposed but these are often limited by initialization parameters, thus only semi automatic, and only to one or two blood vessels. Using streamlines [12] it is possible to visualize the flow, but this does not effectively segment the entire cardiovascular tree and requires a lot of user input.

2.5 Segment

In this thesis the software package Segment was used. Segment [13] is a large open source software developed in MATLAB by Einar Heiberg and co workers Lund Cardiac MR Group at the Department of Clinical Physiology at Lund University and in collaboration with Medviso AB [14] under the supervision of Einar Heiberg.

The software is capable of analysing and quantifying data from many different medical images; MRI, CT, SPECT and PET. The tools in this software include features such as quantification of MRI flow and segmentation of the left ventricle. The source code for the software is available on the project homepage http://segment.heiberg.se for research or educational purposes.



LVM: - [ml]
LVM: - [g]
RVM: - [ml]
RVM: - [g]
LV-EDV:NaN [ml]
LV-ESV: 0 [ml]
LV-SV: 0 [ml]
LV-EF: 0 [%]
HR: 58 [bpm]
CO:0.0 [l/min]
LV-PFR:NaN [ml/s]
LV-PER:NaN [ml/s]
RV-EDV: 0 [ul]
RV-ESV: 0 [ul]
RV-SV: 0 [ul]
RV-EF: 0 %

Slice:26 Time:218 ms
3DcinePC_FH
ORIGINAL\PRIMARY\VELOCITY MAP\P\PCA
Scout,Unspecified

Slice:26 Time:218 ms
3DcinePC_FH
ORIGINAL\PRIMARY\VELOCITY MAP\P\PCA
Flow (left/right) antialiased background corr.,Unspecified

Slice:26 Time:218 ms
3DcinePC_FH
ORIGINAL\PRIMARY\VELOCITY MAP\P\PCA
Flow (up/down) antialiased background corr.,Unspecified

Slice:26 Time:218 ms
3DcinePC_FH
ORIGINAL\PRIMARY\VELOCITY MAP\P\PCA
Flow (through plane) antialiased background corr.,Unspecified

Figure 2.8: Segment with 4D PC-MRI data loaded

2.6 Fourflow

Fourflow [15] is an open source software for quantification and visualization of 4D PC-MRI data that enables development of new quantitative analysis tools. It was developed by Christopher Green in Lund Cardiac MR Group at the Department of Clinical Physiology at Lund University. Like Segment, this software and its source code is available for download on the project homepage http://heiberg.se/fourflow.

Chapter 3

Aim

The aim of this thesis is to automatically visualize the full cardiovascular tree in 3D PC-MRI images in order to simplify clinical research and studies in 4D flow images.

The specific aims are the following:

1. Develop an automatic method for detecting the cardiovascular vessel tree from 4D PC-MRI flow images.

2. Implement the algorithm in an existing clinical research program (Segment).

3. Export detected blood vessel data to a dedicated visualization software (Fourflow).

4. Evaluate the performance by comparing with previous existing methods.

Chapter 4

Method

The noise introduced in the phase images by air in the lungs and around the body presents a problem when trying to identify coherent flow in the data. To filter out this noise the application of a simple low pass filter seemed to be an easy solution. Although in theory the noise presented by air in phase images should be close to white noise this was not applicable in clinical data. This is most likely due to the fact that the number of time frames in the datasets is small. The number of time frames range between 19 and 40 frames covering one cardiac cycle. Therefore, several preprocessing methods were tested and although only a few are used in the final algorithm several of these methods will be explained below to help further studies in this area.

The central idea was that with sufficient preprocessing of the data filtering out noise and tissue in the phase images would lead to segmentation of cardiovascular flow. Since most of the essential flow dynamics occur during the systolic part of the heart cycle only these time frames, T_s, were used to visualize the cardiovascular tree.

4.1 Notations

1. x - Notation of scalars using lower case italic characters.

2. $\hat{x}$ - Notation of scalar field using lower case italic hat characters.

3. H - Upper case bold characters are filters.

4. $\vec{v}$ - Notation for vectors.

5. $\vec{\mathbf{v}}$ - Vector fields are denoted by bold face letters.

4.2 Definition

The velocity in a voxel in the dataset is defined by

$$D \subset R^4 \tag{4.1}$$

$$\hat{v}_g : D \rightarrow [0,1] \tag{4.2}$$

$$\hat{v}_g(x,y,z,t) \in [0,1] \tag{4.3}$$

$$g = G_1, G_2, G_2 \tag{4.4}$$

the velocity $\hat{v}_g = \hat{v}_g(x,y,z,t)$, where z is the coordinate of a particular slice. For each iteration of the features z is kept fixed. $\hat{v}_g$ is the voxel intensity in the four dimensional space D corresponding to a gradient direction g.

4.3 Local features

4.3.1 Local Deviation

The first proposed feature was observing local deviations around each pixel in a separate phase image. This was to identify the erratic behaviour of noise. The spatial resolution of the neighbourhood was 3 × 3 pixels while temporal resolution was 3 time frames. The value of each pixel in the 3 × 3 × 3 volume was compared to the mean of the volume in order to express local deviation around the pixel.

$$\hat{M}_{avg} = \mathbf{H} * \hat{v}_g \tag{4.5}$$

$$\hat{p}_{ld}(x,y,t) = \sum_{i=1}^{n} |\hat{M}_{avg}(x,y,t) - \hat{v}_g(x_i,y_i,t)| \tag{4.6}$$

where $\mathbf{H}$ is an average filter of size 3×3, $*$ denotes convolution, n is the number of pixels in the neighbourhood. The local deviation, $p_{ld}(x,y,t)$ in each pixel is given by Eq (4.6). This process is repeated for each pixel and averaged over systole in the phase image v_g.

4.3.2 3D flow vector analysis

By combining the three phase images, G_1, G_2, G_3, the velocity can be expressed in vector form $\vec{v} = (v_{G_1}, v_{G_2}, v_{G_3})$.The magnitude of the velocity is expressed by

$$|v| = \sqrt{v_{G_1}^2 + v_{G_2}^2 + v_{G_3}^2} \tag{4.7}$$

Both the magnitude and direction of this velocity field tell us important information about what is coherent flow and what is not. The noise in phase images may present some similar linear components to the flow, but in general the vector field in these areas shows a larger angular spread and less coherent magnitude. With the assumption that coherent flow will have a more uniform vector field in both direction and magnitude than tissue and noise in the image three different coherence measures were tested.

Angular Spread

By creating a $3 \times 3 \times 3$ voxel neighbourhood around each voxel and comparing the angle of each individual 3D vector with the median angle of the 3D vector neighbourhood an expression for the angular spread is achieved. The angles seen in Figure 4.1 are calculated b taking the arcustangens of the x, y and z directions. The angular spread is defined by:

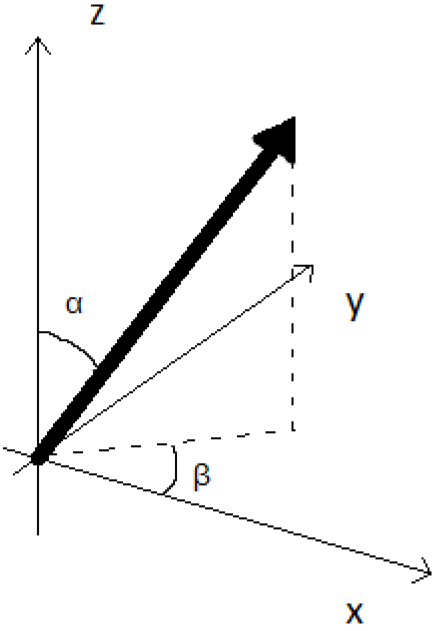


Figure 4.1: α and β angles

$$\vec{\mathbf{v}}_{gC} = \mathbf{H} * \vec{\mathbf{v}}_{g} \tag{4.8}$$

$$\vec{\mathbf{v}}_{C} = (\vec{v}_{G1C}, \vec{v}_{G2C}, \vec{v}_{G3C}) \tag{4.9}$$

$$\hat{p}_{ang}(x,y,t) = \frac{1}{n}\sum_{i=1}^{n} |ang(\alpha,\beta)| \tag{4.10}$$

where $\mathbf{H}$ is a two dimensional mean filter, $*$ denotes the convolution, $\vec{\mathbf{v}}_C$ is the filter output for all three gradient directions, n is the number of pixels in the neighbourhood, $a\hat{n}g_v(\alpha,\beta)$ is defined as the angles for each 3D vector within the neighbourhood as presented by Fig. (4.1), z is kept fixed for each iteration. This process is repeated for each pixel in each slice of the dataset. The normalized sum for systole of this neighbourhood around each voxel is used as an indicator of whether or not the voxel is part of coherent blood flow. In order to follow the suit of other features where a larger value means a larger probability for coherent flow the values of $\hat{p}_{ang}$ where modified as follows:

$$\hat{p}_{ang}(x, y, t) = |\hat{p}_{ang} - \sqrt{2}\pi| \tag{4.11}$$

Structure tensor

By constructing a structure tensor and evaluating its eigenvalues and corresponding eigenvectors it is possible to derive information about what direction of flow within a volume is most prominent. M_v is a positive semi definite matrix as defined by:

$$M_v = \vec{v}\vec{v}^T = \begin{bmatrix} v_{G_1} \\ v_{G_2} \\ v_{G_3} \end{bmatrix} \begin{bmatrix} v_{G_1} v_{G_2} v_{G_3} \end{bmatrix} = \begin{bmatrix} v_{G_1}^2 & v_{G_1}v_{G_2} & v_{G_1}v_{G_3} \\ v_{G_1}v_{G_2} & v_{G_2}^2 & v_{G_2}v_{G_3} \\ v_{G_1}v_{G_3} & v_{G_2}v_{G_3} & v_{G_3}^2 \end{bmatrix} \tag{4.12}$$

$$\hat{M_v} = \mathbf{H} * M_v \tag{4.13}$$

where $*$ denotes the convolution, **H** is an average filter, z and t is fixed for each iteration. $\hat{M_v}$ has 3 eigenvalues; λ_1, λ_2, λ_3, one for each direction in space. The size of the eigenvalue is directly related to the amount of flow in the direction of its eigenvector. If all flow would flow in the same direction there would only be one large eigenvalue, $\lambda_1 = 1$ and its eigenvector would be in the direction of the flow. However there is usually flow in more than one direction, but in general one eigenvalue is larger than the others.

For this method the largest eigenvalue was assessed as this corresponds to the most dominant direction for flow within the neighbourhood. If there no dominant direction in the neighbourhood the value of all eigenvalues will be low, approximately 0.4. If there is coherent flow in the neighbourhood a single direction should be more dominant than the others. This will yield one eigenvalue much higher than the others. Thus, voxels with coherent flow will yield a higher max eigenvalue than a voxel with incoherent flow.

$$\hat{p}_{tens} = \frac{1}{n} \sum_{i=1}^{n} max(eig(\hat{M_v})) \tag{4.14}$$

where n is the size of the neighbourhood. The value for this eigenvalue was averaged during systole for each pixel within the slice. This is then repeated for each pixel for each slice. A larger value of the feature indicates coherent flow.

Projected Velocity Magnitude

The last and most successful feature uses a combination of angle and magnitude. By letting all the 3D vectors within the neighbourhood project on the median vector of the same neighbourhood an expression for projected velocity magnitude, $\hat{p}_{proj}(x, y, t)$, is achieved which is an indication of coherent flow.

$$\vec{\mathbf{v}}_{gC} = \mathbf{H} * \vec{\mathbf{v}}_g \tag{4.15}$$

$$\vec{\mathbf{v}}_C = (\vec{v}_{G1C}, \vec{v}_{G2C}, \vec{v}_{G3C}) \tag{4.16}$$

$$\hat{p}_{proj}(x, y, t) = \frac{1}{n} \sum_{i=1}^{n} \vec{\mathbf{v}}_i \cdot \vec{\mathbf{v}}_C \tag{4.17}$$

where $\mathbf{H}$ is a two dimensional mean filter, $*$ denotes the convolution, $\vec{\mathbf{v}}_C n$ is the filter output for all three gradient directions, n is the number of pixels in the neighbourhood, $\vec{\mathbf{v}}_i$ is the a vector in the neighbourhood, $\vec{\mathbf{v}}_C$ is the median 3D vector and $\cdot$ denotes the scalar product of two vectors. This process is afterwards repeated for all pixels and averaged over systole. If many of the vectors within the neighbourhood are aligned in a similar direction they will all project much of their magnitude on the median vector direction. However if vectors within a neighbourhood are spread in different directions most of these will project poorly onto the median vector and thus resulting a small output for $\hat{p}_{proj}$. During systole the magnitude of the coherent flow within vessels is large compared to its surroundings which further increases the effectiveness of the feature.

4.4 Implementation of final algorithm

In the algorithm used for visualization the product of the projected velocity magnitude feature and the actual magnitude image is used to finalize a probability map of the cardiovascular tree. It runs through all the slices of the input dataset and returns a new dataset. Figure 4.2 illustrates the data flow.

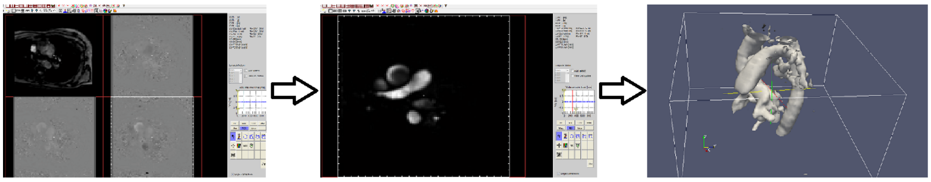


Figure 4.2: Figure demonstrating the data flow through the different interfaces. Starting from the left data is loaded into segment. In the middle the Vessel Probability Module has been run creating a probability map. In the last picture the data has been exported to the software Fourflow where and isosurface has been applied to finalize visualization.

After the raw data has been acquired by the MR scanner three phase images and one magnitude image are loaded into Segment [?]. Segment unwraps the data and does some background correction of the images. The Vessel Probability Module, which has been developed with this thesis, is a plugin to Segment. When the data is loaded into Segment the plugin has two menu options:

1. Probability Map - This generates a new image stack with probability map for vessel visualization.
2. Export to Fourflow - This option exports the data for vessel visualization and subsequent quantification.

The first option initializes the velocity probability calculations. This calculates the product of the projected velocity magnitude feature and the magnitude image. These are combined into a probability map, created in a new image stack within the dataset. Once this is done running the second option, "Export to Fourflow", runs a script to export the data into a format that fits Fourflow. Note that both the velocity data from the three phase images and the probability map is exported.

In Fourflow a simple isosurface is applied to the probability map in order to visualize it in 3D space. The user can adjust the isosurface value up or down depending on how important visualization of smaller blood vessels. A higher value will allow visualization of small blood vessels within the lungs, however it might lead bigger blood vessels to be visualized as joined together. For further details see Section 5.3.2. Transparency of the visualized blood vessels can also be altered and coloured in many ways if the operator chooses to.

With the visualization at hand further quantification of the flow data can be done by for example simulating streamlines or particles within the volume at a certain time and letting them follow the flow introduced by the velocity data.

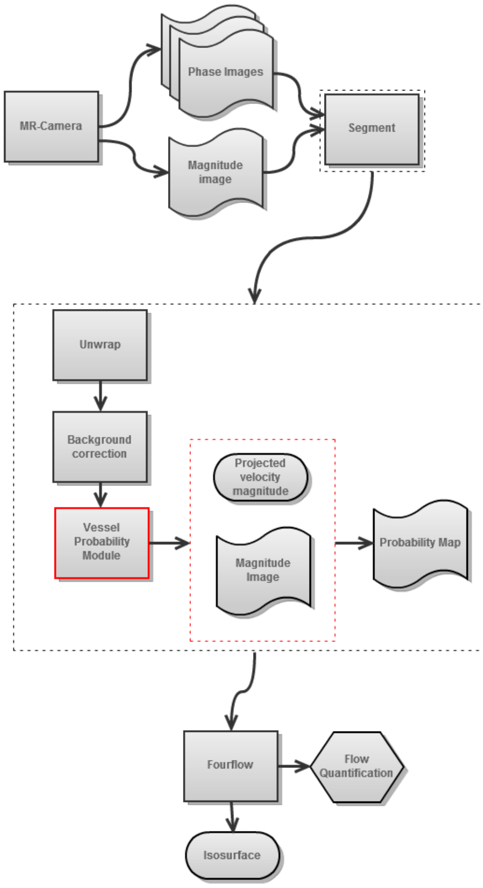


Figure 4.3: Schematic figure of the implementation showing the data flow from start to finish. This thesis developed parts in red borders.

Chapter 5

Results

5.1 Features

The results in this section are all show from the same clinical data from a healthy volunteer and where possible the same slice and time frame. The size of the data was $80 \times 80 \times 48$, 48 being the number of slices in the dataset. Temporally there are 15 time frames. The data was acquired at Lund University Hospital from healthy volunteers with a 1.5 Tesla MRI machine with a voxel resolution of $3 \times 3 \times 3$ mm.

5.1.1 Local deviation

The local deviation was used primarily as an attempt to identify noise. Figure (5.2) shows the noise in lungs and around the patient being highlighted with the sum of deviations during diastole. In Figure 5.3 the maximum deviation during systole is displayed. The low resolution causes the blood vessels to be included along with the noise since the deviation in magnitude between 3 frames during systole is huge, see Figure 2.5 in the section 2.2. These images are from the phase image in z-direction.

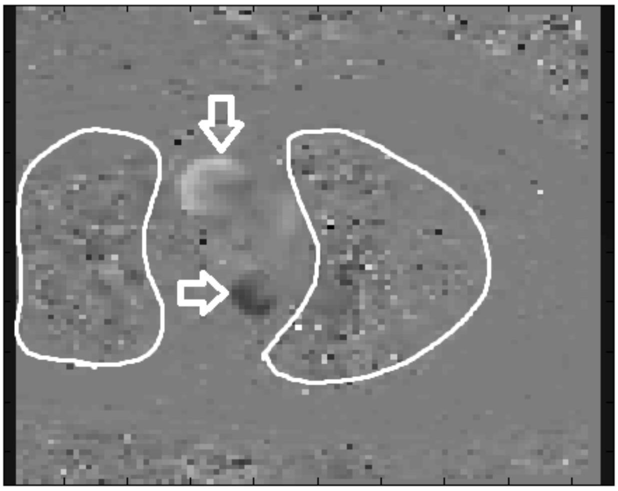


Figure 5.1: Original phase image in z-direction. Notice the substantial noise in the lungs and on the edges in the picture. This noise occurs because it is air. The most noticeable vessel is the aorta. Both the ascending and descending part of the aorta are marked with the white arrows.

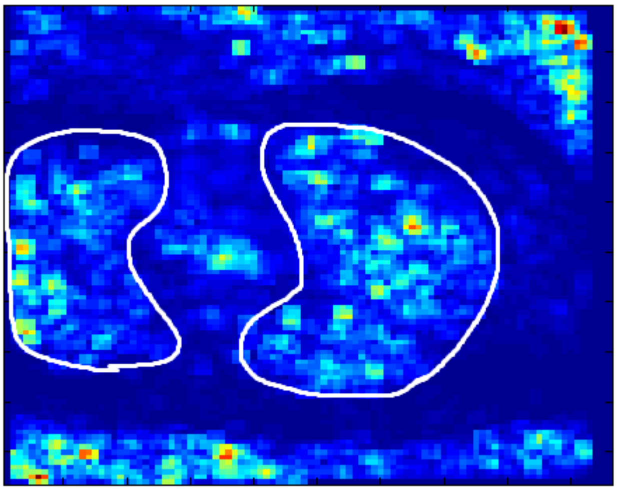


Figure 5.2: The sum of deviations during diastole. The noise in the lungs and around the patient caused by air is highlighted by the feature without including any of the blood vessels.

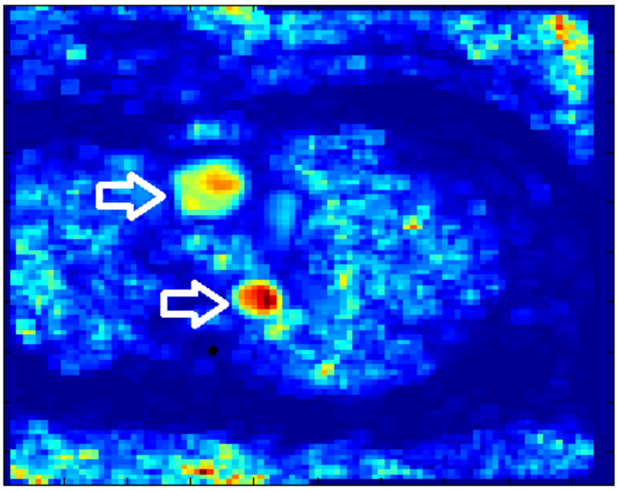


Figure 5.3: The max deviation during systole. The low time resolution leads the aorta to be included when observing max deviation during systole, marked by whit arrows.

5.1.2 Angular spread

When visualizing this feature in Fourflow an isosurface threshold of 0.9 was used. It is clearly visible in Figure 5.5 that this feature introduces a lot of noise due to flow within tissue.

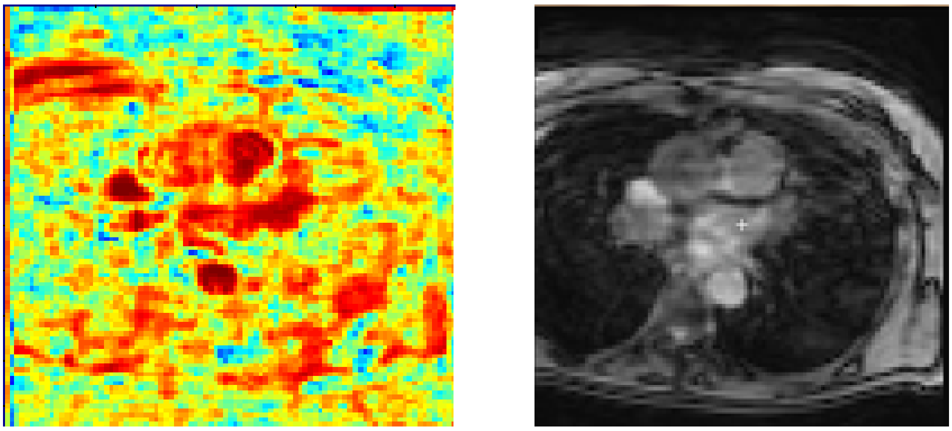


Figure 5.4: Illustration of the angular spread feature. Left: A higher lower value corresponds to a coherent flow. Right:original magnitude image.

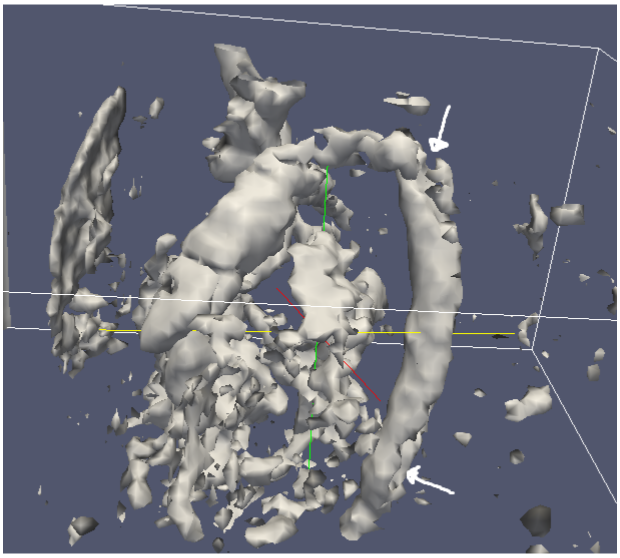


Figure 5.5: Visualisation in Fourflow using the angular spread feature. Notice the hollow points in aorta marked by the arrows. Isosurface is 0.9

5.1.3 Tensor

As with angular spread, visualizing this feature in Fourflow an isosurface threshold of 0.9 was used. This feature yields a much smoother visualization than the angular spread feature. However, it is filled with noise introduced by flow in tissue. Notice how the pectoral musculature is visible to the left and that the cardiac musculature around the ventricles is visualized, Figure 5.7.

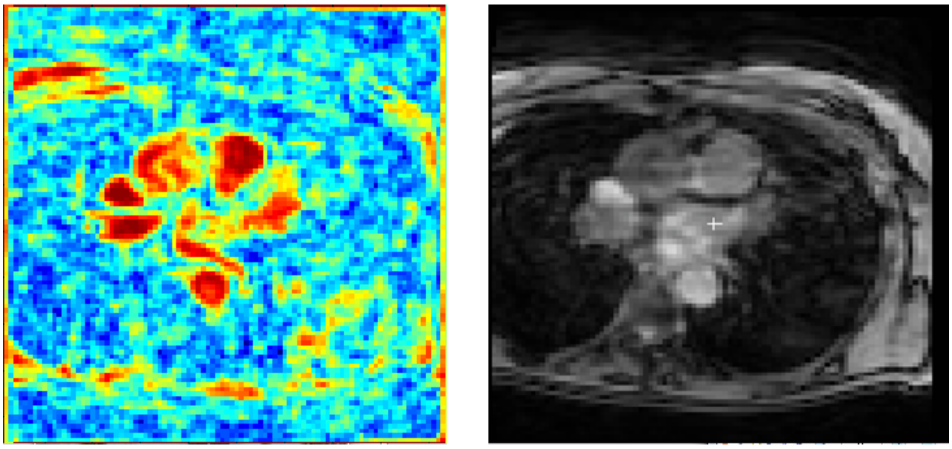


Figure 5.6: Left: Image displaying the tensor feature. A higher value corresponds to a coherent flow. Right: The original magnitude image

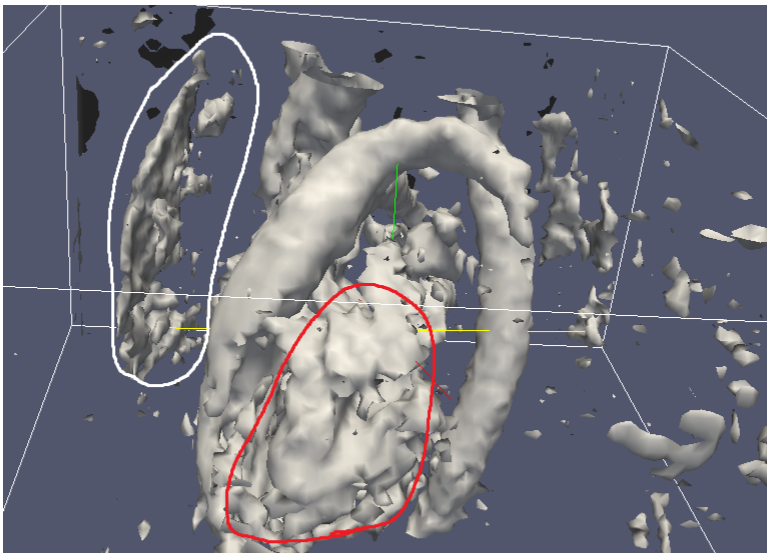


Figure 5.7: Visualisation in Fourflow using the tensor feature. Isosurface is 0.9. Notice visualization of vessel and the noise from pectoral (white) and cardiac tissue (red).

5.1.4 Projected velocity magnitude

By projecting onto the mean direction within a neighbourhood the projected velocity magnitude feature identifies the cardiovascular vessel tree. The isosurface for this feature is set to 0.1 which is considerably lower than the angular spread and tensor features. Notice how the noise from surrounding tissue has been eliminated and the cardiovascular tree is clearly visualized.

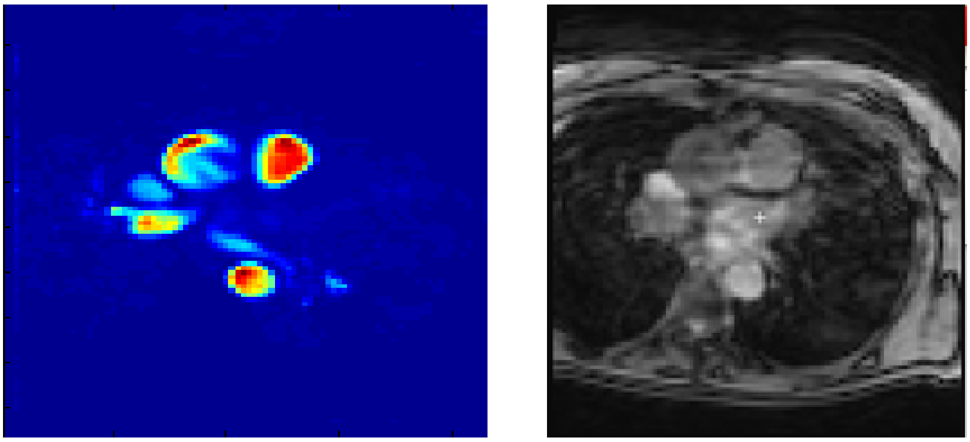


Figure 5.8: Left: Image displaying the tensor feature. A higher value corresponds to a coherent flow. Right: The original magnitude image.

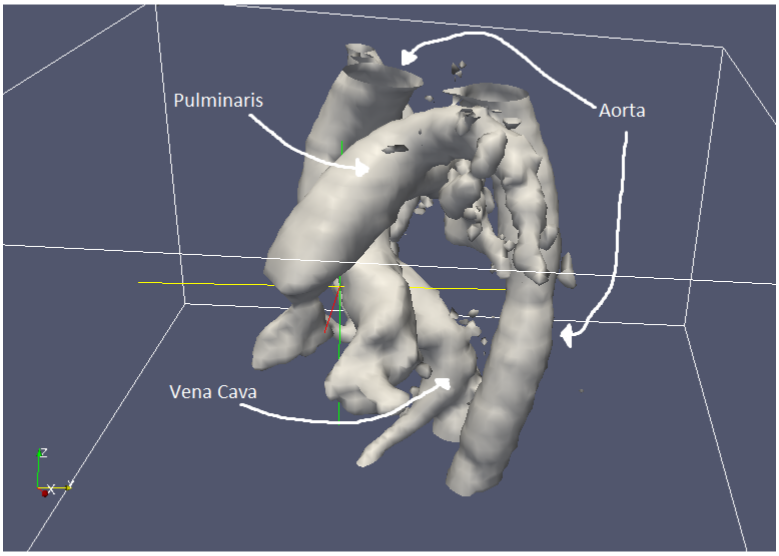


Figure 5.9: Visualisation in Fourflow using the projected velocity magnitude feature. Isosurface is set to 0.1.

5.2 Final algorithm

For the final algorithm the projected velocity magnitude feature is complemented by combining it with information from the magnitude image. The result is further filtering of visual artefacts introduced by tissue and noise. Although the spatial resolution is similar for all datasets, the time resolution can vary to a much higher degree. Using the algorithm on datasets with higher time resolution yields much better visualization of the smaller vessels within the lungs.

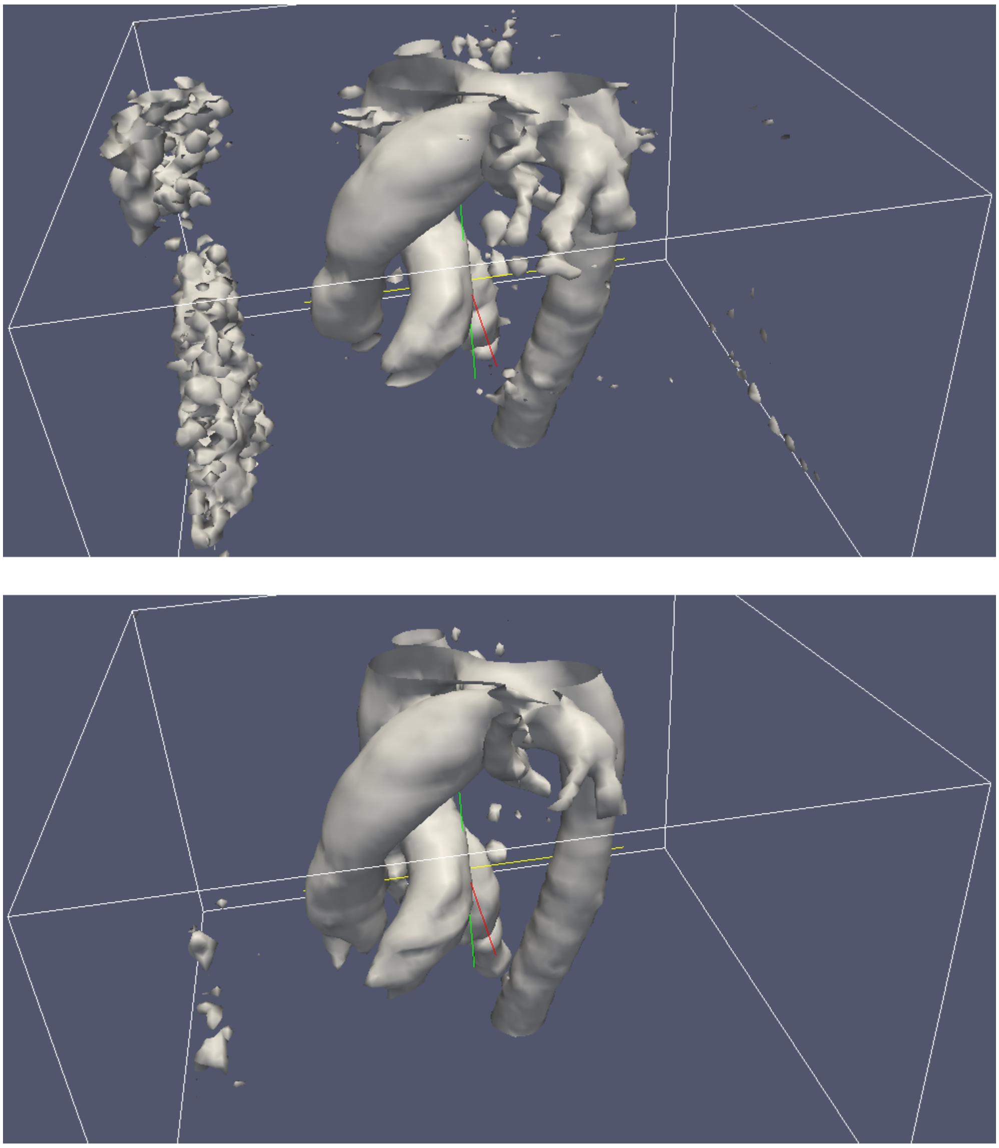


Figure 5.10: Top: Projected velocity magnitude weighted down with information from the magnitude image. Bottom: Projected velocity magnitude feature.

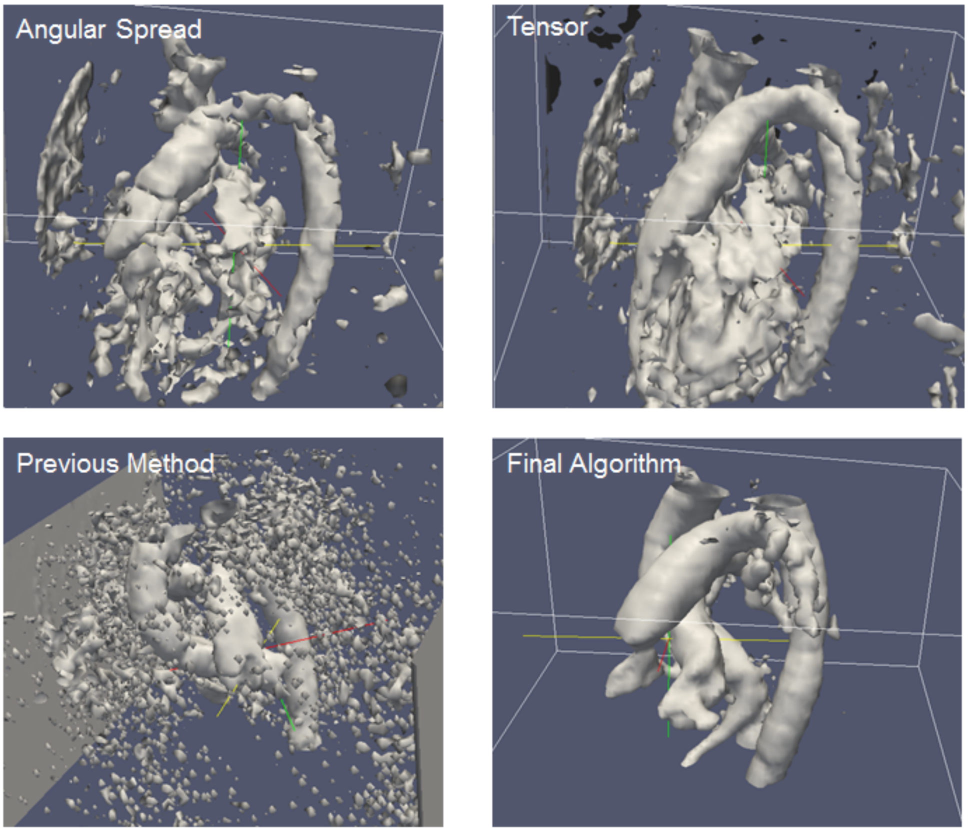


Figure 5.11: Comparing different features. Top left is visualization with angular spread. Top Right is visualization with tensor feature. Bottom right is visualization with the final algorithm and the Bottom left is the previous method.

5.3 Validation

5.3.1 Time resolution

A larger time resolution yields better results with low isosurface values. Below is an image of the same patient data but with different temporal resolution. The low resolution had a time window of 15 while the high resolution data had 40 time frames. Spatial resolution 80 × 80 × 48.

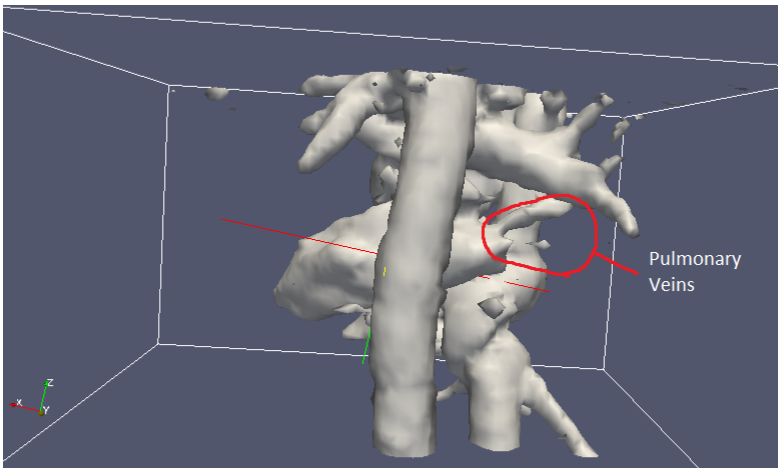


Figure 5.12: Data with low temporal resolution. Isosurface set to 0.03

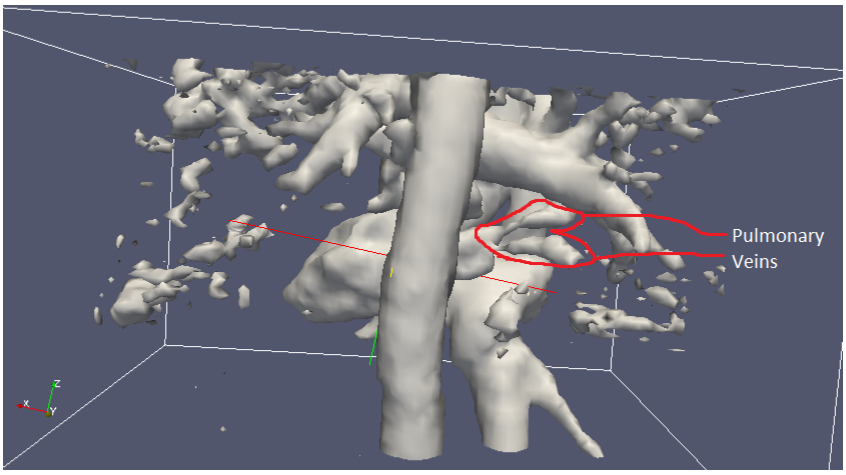


Figure 5.13: Data with high temporal resolution. Isosurface set to 0.03

5.3.2 Isosurface values

It is possible to decide in how much detail of the cardiovascular tree gets visualized by setting different thresholds. Lowering the value will lead to more information gathered by the vessel probability algorithm to be included in the visualization. Lowering this value makes it possible to identify small bloodvessel, but the image can feel bloated.

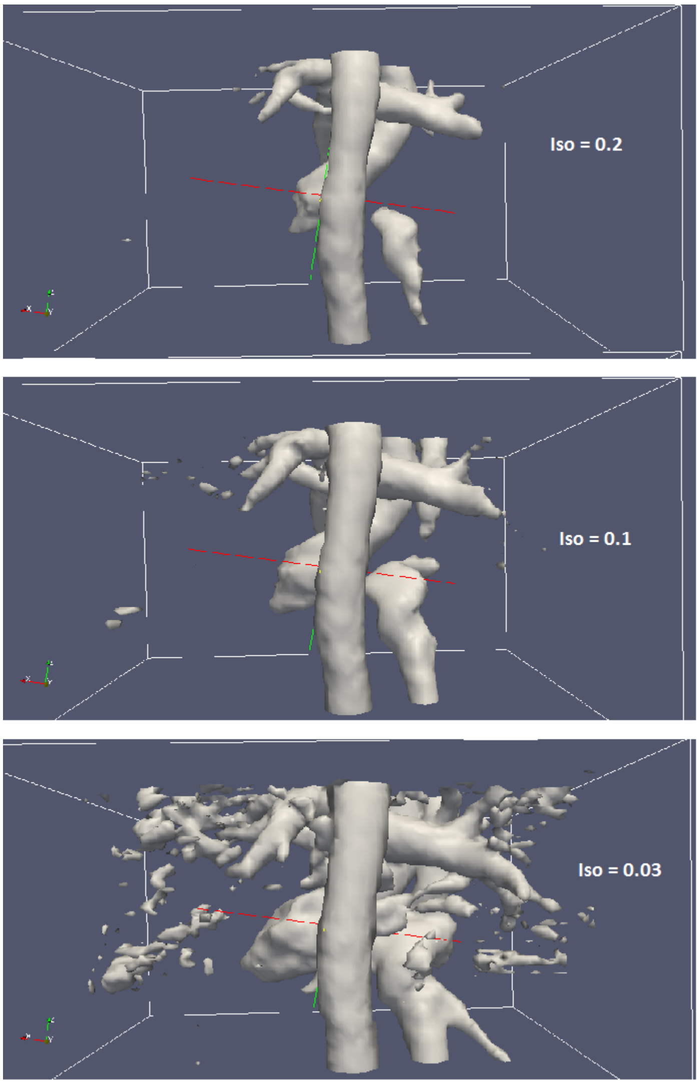


Figure 5.14: The same data with different values of the isosurface.

5.3.3 Previous method

The previous visualization utilized an isosurface with a threshold set to the square sum of the phase images. The user would then find an appropriate time frame and isosurface threshold, usually during systole, to try and achieve visualization. Below are images acquired with this method that have been tweaked to the best possible visualization regarding isosurface threshold and time selection.

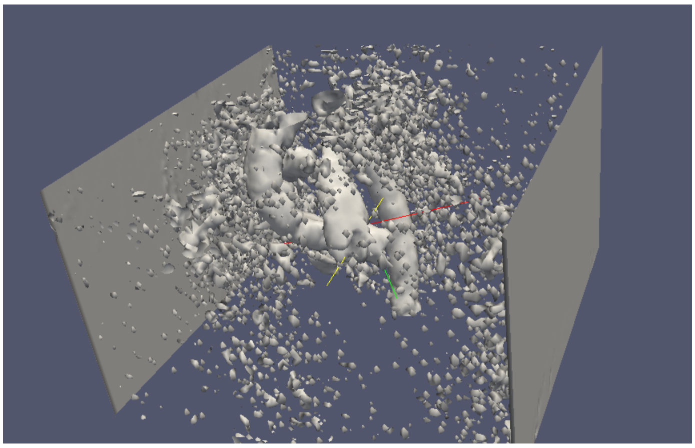


Figure 5.15: Previous visualization in Fourflow. Isosurface is set to 0.1 and the time is the peak of systole.

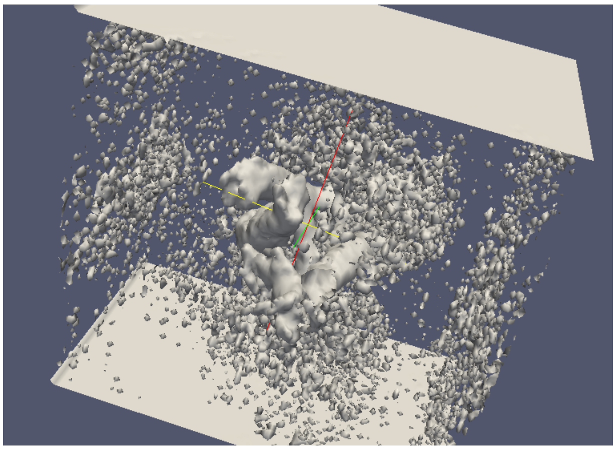


Figure 5.16: Previous visualization in Fourflow. Isosurface is set to 0.1 and the time is the peak of systole.

5.3.4 Applications

It is often of great use to use transparent visualization of the cardiovascular tree in conjunction with flow quantification tools within Fourflow. Streamlines can be introduced which will be influenced by the flow data. Using this tool shows how all the flow is moving within the visualized blood vessels.

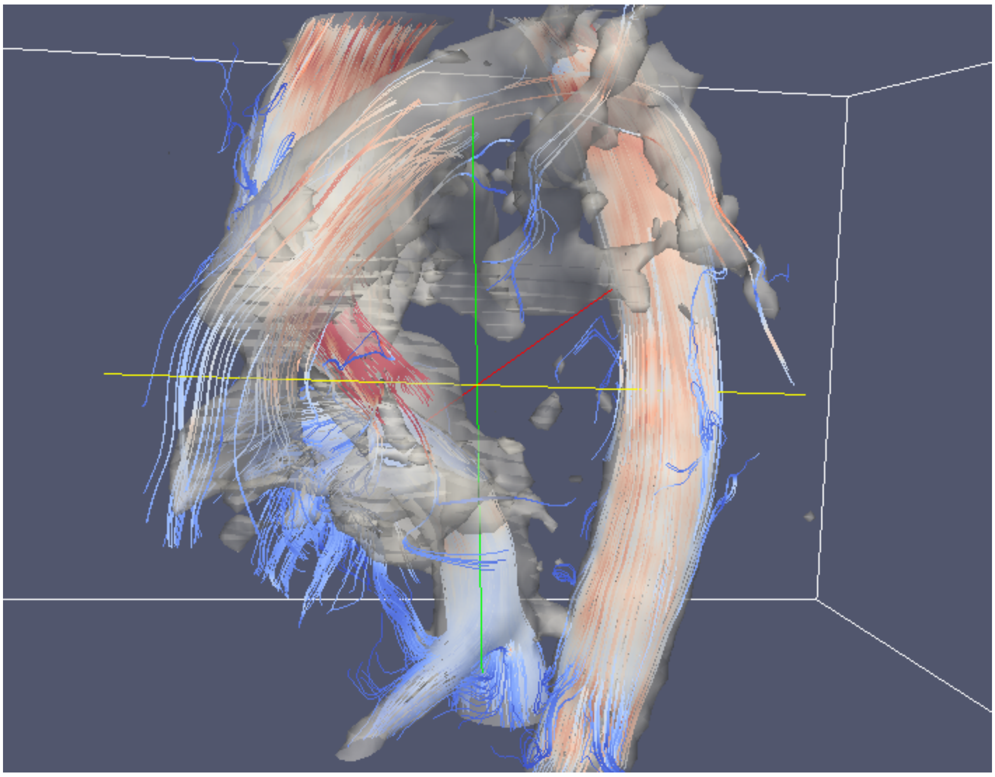


Figure 5.17: Visualizing streamlines within the cardiovascular tree during one systole timeframe

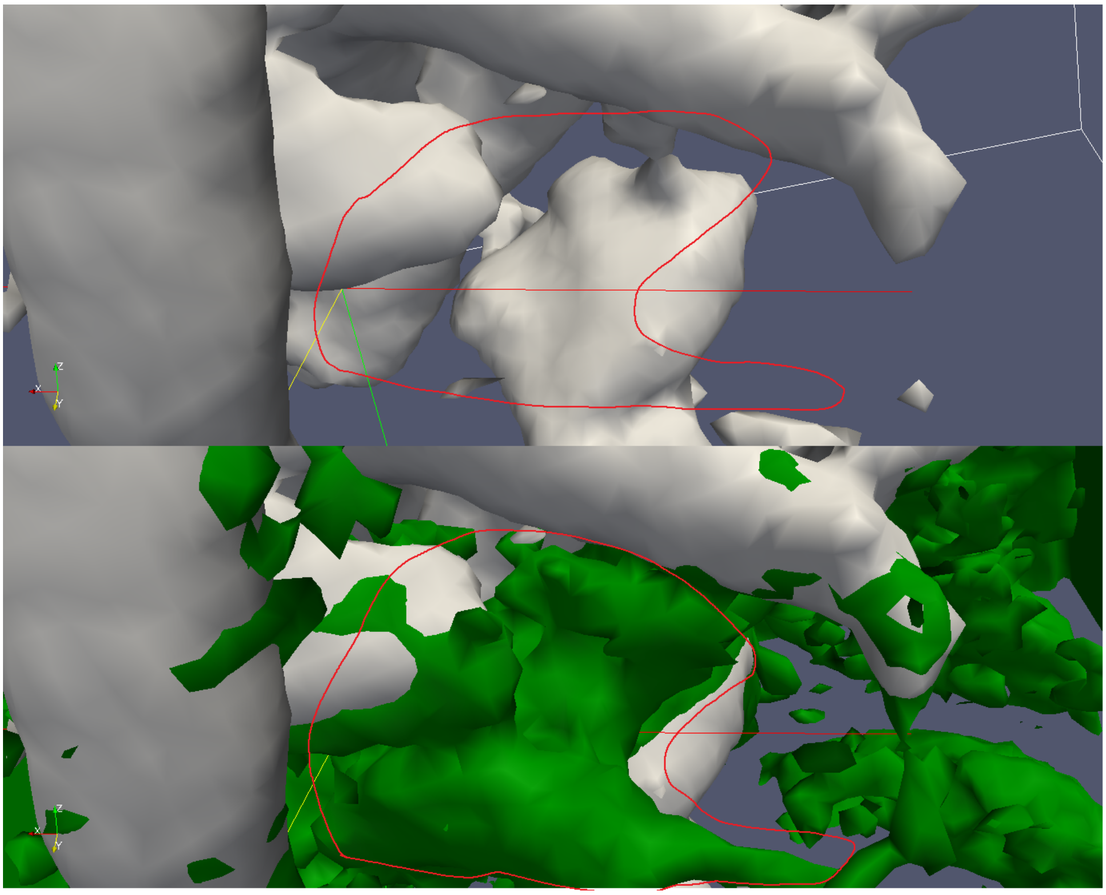


Figure 5.18: Top: Projected velocity magnitude feature. The marked red area is the left atrium which is not visualized. Bottom: Tensor feature in green showing more noise but a more pronounced visualisation of flow in the atrium and the connecting veins from the lungs

Chapter 6

Discussion and Conclusions

6.1 Discussion

The previous method of using the magnitude of the combined phase, Figure 5.15-16, manages to visualize parts of the aorta and pulmonaris, but there is a substantial amount of noise making identification and orientation difficult. The algorithm developed in this thesis manages to automatically visualize the entire vessel tree to an acceptable accuracy.

However the algorithm has some weak spots. Being restricted to measuring during systole certain vessels which don't see much flow during this time are not very prominent. Looking at results from the tensor feature in Figure 5.18 it shows a more clear visualization of the pulmonary veins than the final algorithm. However, it performs much worse for the rest of the cardiovascular system as seen in Figure 5.7.

6.1.1 Different features

Several of the different features introduced in this thesis show some interesting results. The simple local deviation feature failed to identify coherent flow. However it did succeed rather well in identifying noise introduced by air in the images when applied during diastole. This could be used as a pre-preprocessing method to further improve visualization.

All of the 3D vector analysis methods show rather promising results. They all effectively identify the blood vessels to a varying degree. It is clear that the helical flow within the aorta presents an issue when using the angular spread feature as the resulting visualization is hollowed out in the aorta descending. However the angular spread and tensor features have tenancies to include tissue. These two features only take the direction of flow into account. Inside tissue there is some sort of coherent flow which is most likely the reason they show up with these methods.

Projected velocity magnitude works well because it also takes the amplitude of the flow into account. During systole there is a large amount of flow within the blood vessels. While there is still coherent flow within surrounding tissue the magnitude of this flow is but a fraction of that within the blood vessels. This way the simple feature makes use of two of the main identifying characteristics of flow within the blood vessels.

6.1.2 Accuracy of final visualization

The results shown in this thesis presents a fairly complete visualization of the cardiovascular tree. When the sensitivity of the isosurface is high even the smaller branches of the pulmonary arteries and veins are visible. Although these visualizations do not give a fully accurate image due the fact that it freezes the time frame this is acceptable considering the purpose is only to automatically visualize the cardiovascular tree. These visualizations themselves are not a direct part of any quantitative analysis and thus the fact that the vessels at a certain part of the cardiac cycle might not be at the exact same location and size of the visualization is irrelevant.

6.1.3 Further improvement

Combining features

By combining several of the features the vessel probability map might be improved. The local deviation could be used to subdue noise prior to preprocessing. The tensor feature shows a higher sensitivity to noise and an ability to more easily visualize vessels that do not have any prominent flow during systole. A combination of the tensor and projected velocity magnitude features could lead to better visualization of smaller blood vessels, but it would require to remove noise from the tensor feature introduced by surrounding tissue. However the combination of features would also lead to a substantial increase in computation time. The possible gain from combining features has to outweigh the increase in computation time.

Computation time

The biggest weakness of the algorithm is the computation time. For a dataset with time resolution of 15 frames it takes roughly 4.5 minutes to run the algorithm. The data in this dataset is gathered prospectively. If the data is gathered retro prospectively you have up to 40 time frames. this almost triples the size of the dataset and the computation time for a dataset this large takes around 12 min. The code of the algorithm can optimized for faster execution and less memory consumption. It is written completely in MATLAB and if one was optimize the code in for example C or any other lower level language computation time would probably be reduced significantly.

User interface

At the moment the user has to:

1. Load data into Segment and run the Vessel Probability Module.
2. Export the probability map created in Segment new Fourflow compatible files.
3. Load data and apply the isosurface.

It would be beneficial to include the Vessel Probability module and file exportation in Fourflow to avoid having to use two different programs. The final visualization is used to aid the user in Fourflow and thus the commands to achieve visualization should be within Fourflow. This is doable without too much work, but something that there was not time to implement with this thesis.

6.2 Conclusion

The algorithm developed in this thesis shows great potential in visualization of the cardiovascular tree from 4D PC-MRI images. From Figure 5.13 all of the major vessels can be located and identified. This visual information allows the user to fast and accurately focus on regions of interest for further quantification and analysis of flow data. Further work may include combining features and optimizing the code for optimal performance.

Bibliography

[1] L. Wigström, L. Sjöqvist, and B. Wranne. Temporally resolved 3D phase-contrast imaging. *Magnetic Resonance in Medicine*, 36:800–803, 1996.

[2] M. Persson, J.E. Solem, K. Markenroth, J. Svensson, and A. Heyden. Phase Contrast MRI Segmentation Using Velocity and Intensity. 2005.

[3] F. Bloch and E.M. Purcell. Nuclear Induction. *Phys Rev*, 70:460–474, 1946.

[4] P.C. Lauterbur. Image formation by induced local interactions. Examples employing nuclear magnetic resonance. *Nature*, 242:190–191, 1973.

[5] N. J. Pelc, T.J. Herfkens, A. Shimakawa, and D. Enzmann. Phase contrast cine magnetic resonance imaging. *Magnetic Resonance Quarterly*, (4):229–254, 1991.

[6] ”The origin of the NMR signal”, http://chemwiki.ucdavis.edu.

[7] http://www.rsc.org/ej/CS/2012/c1cs15248c/c1cs15248c-f2.gif.

[8] T.H. Jochimsen and M. von Mengershausen. ODIN Object oriented Development Interface for NMR. *Journal of Magnetic Resonance*, 170(1):67–78, 2004.

[9] Collaborative. ”The Human Heart”, http://en.wikipedia.org.

[10] B. Jonson and P. Wollmer. Klinisk fysiologi med nuklearmedicin och klinisk neurofysiologi. 2005.

[11] P. Makrowski, T. Sangild Sorensen, S.V. Therkildsen, A. Materka, H. Stodkilde-Jorgensen, and E.M. Pedersen. Two-phase active contour methid for semiautomatic segmentation of the heart and blood vessels from MRI images for 3D visualization. *Computerized Medical Imaging and Graphics*, 26(1):9–17, 2002.

[12] S. Napel, D.H. Lee, R. Frayne, and B.K. Rutt. Visualizing three-dimensional flow with simulated streamlines and three-dimensional phase-contrast MR imaging. *Journal of magnetic resonance imaging*, 2(2):143–153, 2005.

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

[14] ”Segment”, http://segment.heiberg.se/fourflow. 2013.

[15] ”Fourflow”, http://segment.heiberg.se/. 2013.