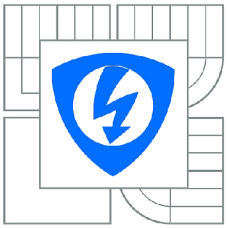
BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ



FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

SEGMENTATION OF BLOOD-VESSEL TREE IN WHOLE-BODY MRI DATA

SEGMENTACE CÉVNÍHO ŘEČIŠTĚ V CELOTĚLOVÝCH MRI DATECH

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR
AUTOR PRÁCE

KAROLÍNA GURICOVÁ

SUPERVISOR
VEDOUCÍ PRÁCE

doc. Ing. RADIM KOLÁŘ, Ph.D.

BRNO 2015

Abstrakt

Práce popisuje anatomii a vlastnosti cévního řečiště s charakteristickými znaky, na kterých je založena jeho segmentace. Nejprve jsou uvedeny metody segmentace 3D CT a MRI skenů. Více detailně jsou popsány základy segmentace založené na druhých derivacích a Hessově matici. K určení podobnosti cévám v původním obraze jsou spočítány vlastní čísla Hessovy matice každého voxelu. K vytvoření výsledného segmentovaného obrazu je navrženo více metod pro zpracování těchto vlastních čísel. Metoda je prakticky implementována v MATLABu. Vysegmentované arteriální řečiště je visualizováno pomocí knihovny VTK kódované v Pythonu. Dále je navrženo GUI, které umožňuje měření v zpracovaném objemu. Délky artérií jsou aproximovány lineárními úseky kopírujícími jejich cesty a výsledky tohoto měření jsou v práci prezentovány. Limitace této metody a návrh na poloautomatické měření jsou rozebrány na konci této práce.

Summary

The text briefly describes the anatomy and properties of the blood-vessel system to introduce the characteristics that enable segmentation. Firstly, some of the methods for segmentation of tubular structures from 3D CT or MRI scan are described. Moreover, the approaches using the second derivatives and Hessian matrix are elaborated. To determine the 'vesselness' in the original image, the eigenvalues of Hessian matrix of each voxel of the MRI data volume are computed. Several methods to produce the segmented output image based on these eigenvalues are suggested. The segmentation is implemented in MATLAB. The segmented arterial system is visualized with VTK encoded in Python. Additionally, a GUI is designed to allow measurements within the segmented volume. The lengths of the arteries are measured as a linear approximation of their paths and the results are presented in this work. The limitations of this method are described and further suggestions are made for the semiautomatic measurements.

Klíčová slova

segmentace cévní řečiště MRI Hessova matice druhé parciální derivace visualizace 3D délka cév

Keywords

segmentation blood-vessel tree MRI Hessian matrix second partial derivatives 3D visualisation artery length

GURICOVÁ, K.*Segmentation of blood-vessel tree in whole-body MRI data.* Brno: Vysoké učení technické v Brně, Faculty of Electrical Engineering and Communication, 2015. 40 s. Vedoucí doc. Ing. Radim Kolář, Ph.D.

I declare that I have elaborated my bachelor's thesis on the theme of "Segmentation of Blood-Vessel Tree in Whole-Body MRI Data" independently, under the supervision of the bachelor's thesis supervisor and with the use of technical literature and other sources of information which are all quoted in the thesis and detailed in the list of literature at the end of the thesis. As the author of the bachelor's thesis I furthermore declare that, concerning the creation of this bachelor's thesis, I have not infringed any copyright. In particular, I have not unlawfully encroached on anyone's personal copyright and I am fully aware of the consequences in the case of breaking Regulation S 11 and the following of the Copyright Act No 121/2000 Vol., including the possible consequences of criminal law resulted from Regulation S 152 of Criminal Act No 140/1961 Vol.

Karolína Guricová

I would like to thank my supervisor doc. Ing. Radim Kolář, PhD, for his patience and kind support. Another thanks to Ing. Ladislav Soukup for his cooperation on this project. Additionally, I would like to thank Matthias Menne, MSc, for his support and valuable advices.

Karolína Guricová

Contents

List of Figures

List of Tables

List of Symbols

f_n	partial derivative of first order in relation to n
$f_{n,m}$	partial derivative of second order in relation ot n and m
λ	an eigenvalue of a vector
H	the Hessian matrix

1. Introduction

The circulatory system of a human body is one of the most important systems that ensures body's function. Nowadays, with the possibilities of different imaging methods, many information about this system can be acquired. Segmentation of blood-vessel tree provides a lot of valuable details such as narrowing of blood-vessels, obstructions etc. The benefit of the whole-body MRI acquisition comes from the fact that MRI is a harmless modality. In comparison to CT, MRI does not use radiation for the data acquisition. It is therefore desirable to obtain sufficient information using MRI and that can be achieved with the help of image processing.

The aim of this thesis is to segment the blood-vessel tree with the possibility of measuring the distances within the segmented image. In this text the basis of the problematics of the segmentation in 3D data in desribed. Several methods are briefly introduced. The main idea of the blood-vessel tree segmentation is based on its properties that include its nearly circular shape and its continuous propagation throughout the human body. The approach that this thesis introduces in detail is based on the second derivatives of the image intensity. These second derivatives carry the information of structures in the volume. When organized into a Hessian matrix the eigenvalues can be computed. Those eigenvalues are processed further with different methods that allow the evaluation of voxel's 'vesselness'. This work also presents the outcome of the segmentation as a program that enables the measurements within the arterial system. Those measurements are manually selected linear approximation of the arteries' pathways. The limitations of this method are thorougly discussed and presented visually. Also further suggestions for semiautomatic measurements of arteries are described.

The implemetation is done in MATLAB and results are visualized with VTK encoded in Python. The GUI is designed with Qt Designer software.

2. Blood-Vessel Tree as MRI Data

This chapter introduces the properties of the blood-vessel tree. It also describes the characteristics of the MRI data and the acquisition process itself. Also to be able to enhance the desired blood-vessel tree, certain criteria must be met for the acquisition process.

2.1. Anatomy of the blood vessel tree

To be able to fully understand the segmentation of the blood vessel tree, the anatomy of the blood-vessels in a human body is briefly described. The human circulatory system consists of two parts. One is the pulmonary loop where blood is oxygenated. The second part distributes the oxygenated blood to the rest of the body. The system consists of arteries that bring the oxygenated blood to the tissues and the veins that take the deoxygenated blood back to the heart from where they enter the pulmonary loop. The exchange of the oxygen between the tissue and the blood happens in the capilaries. The capilaries have a smaller diameter in comparison to the arteries and veins and also have a thin wall to allow the exchange of oxygen. The diameter of larger arteries is greater than 10 mm and they are generally elastic. Their diameter changes with pressure and causes their cross-section to increase in size with the ejection phase of the heart cycle when blood is pumped into the system.[2]

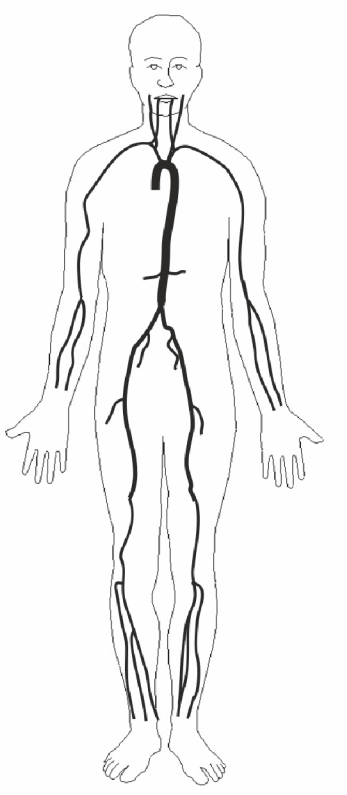


Figure 2.1: Arterial system of the human body

The implemented segmentation as described in Chapter 4 is based on the z-sliced planes. Vessels present nearly circular structures that show specific properties for the implemented detection. The desired blood-vessel tree that is to be segmented is shown in Figure 2.1. The blood-vessel tree with labeled arteries can be seen in 10.1.

2.2. MRI Data and Acquisition

In this section, the MRI data, the process of their acquisition and the used MRI scanner are introduced. For the most efficient segmentation it is necessary to assign the proper algorithm for the data that is being processed. Therefore it is desired to have a thorough information about the volume data set and to be able to understand the acquisition process itself. Also handling a big set of data requires detailed information, so the data volume and relations between separate parts of sequences are preserved. With this additonal information the image is reconstructable. The MRI scanner that is used for acquisition is the Discovery MR750 3.0T GE. Some of its parameters are displayed in the Table 2.2.[8]

Table 2.1: GE Discovery MRI750 3.0T parameters

Bore Size	60 cm
Maximum Field of View	$48 \times 48 \times 48$cm
Homogenity	0.003 ppm @10 cm, 0.05 ppm@20 cm 0.1@30 cm, 0.25@40 cm
Peak Gradient Amplitude	50 mT/m
Slew Rate	200 mT/m/s
Independant Receiver Channel	32

2.2.1. Data Origin and Acquisition

Data comes from a whole-body MRI acquisition. The acquisition is not held as one scan, however it is several scanning sequences. Those sequences are overlapping each other with small number of slices so that the changes that happen within the time gap between separate acquisition sequences are monitored. This helps to keep the whole-body image without any severe differences between single sequences. The information about the position and slice location is present in the DICOM file into which the image is saved. Therefore the reconstruction of the whole-body image is possible.

The acquisition process is based on the detection of blood vessels with a certain flow. The limitation parameters are the direction and the velocity of the flow. With this principle the acquired dataset is limited to the desired arterial structures. As seen in Figure 5, the desired arterial system consists of blood-vessels of greater diameter where detected flows reach higher velocity. To be able to detect the arterial system only, in the acquisition process, the direction is set to be anterior for blood-vessels above the heart and to be posterior for blood-vessels located below the heart for the thoracic cavity. This setting unfortunately causes problems with aortic arch where blood descends in the regions above the heart. For limbs the direction is always set to be distal. The acquisition is also timed with an ECG device that monitors the ejection phase of the heart. That is necessary because the acquisition process is dependent on the blood flow for the purpose of

highlighting the arterial system. Synchronized with the ejection phase, the image is taken when the pressure is the highest in the circulatory system. Therefore the arteries in the resulting image show the cross-section even when taken in different time moments. The acquired image is only dependent on their anatomy and does not vary in time. Without the proper timing, a slice that would be acquired in a different part of the heart cycle might show a vessel that appears to have a smaller diameter. That could lead to an incorrect detection of the narrowing of a blood vessel. By acquiring the image with the synchronization a certain uniformity of the blood-vessel tree is provided.

3. Segmentation of Blood-Vessel Tree in 3D data

In this chapter, different approaches to segmentation of blood vesel tree in 3D data is discussed. These methods are based on different data properties and therefore implement diverse algorithms. Many algorithms dealing with 3D volume use the second derivatives and therefore detect certain structures. However, other approaches are used where the data does not necessarily show a structural type. Those approaches, such as fuzzy connectedness, wave propagation or randomized Hough transform, are discussed briefly. The methods using second derivatives are elaborated further in Chapter 4.

3.0.2. Fuzzy Connectedness

This main idea of this method is the connection between the voxels of a volume. Firstly, a relation is assigned to two voxels to quantify the closeness of them. A function is defined that describes the adjacency of each voxel to the rest of the image. This function enhances the evaluation of the voxels that belong to the same region and therefore ensures a certain homogeneity. Secondly, the description of the resemblance of the intensity is expressed by another function. This function is already dependent on the position one and describes affinity to the region based on the intensity. The size of each region is computed After determining relation between each voxel and remaining voxels of the image, several paths between two points of the volume can be made. The paths are assigned a certain strength based on the connectedness of those two points. By dynamic programming the final segmented image is drawn.[5]

3.0.3. Wave Propagation

This method is based on the idea of a wavefront that propagates through the blood-vessel structure. Initially, a seed voxel is set as the beginning of the algorithm. The wave spreads in all directions and propagates so that the slice of the vessel is marked. This structure is a plane wave. Another property of this seed voxel is the normal vector that establishes the direction of the vessel. This vector is set by the connection between the centres of single plane waves. The axis that is represented by this vector helps with the constructuion of the blood-vessel tree. At each point of this vector, the plane wave is generated for the completion of the image. Additionally, bifurcations and different untypical properties are well segmented. Having the propagation direction, a seed voxel of a plane wave can connect to several centroids of different plane waves and therefore show the right pattern of the vessel structure. This method is used for example for surgical simulations. The segmented data represents the vessel diameter, the curvature and estimates the bifurcation well.[6]

3.0.4. Randomized Hough Transform

This approaches is used in curve detection. It supposes the presence of all possible curves in the image and then assigns a score based on their actual presence. This method is computionaly demanding because of big data volume that needs to be stored. Several

algorithms exist that help with the elimination of this drawback. Another extension of this algorithm is when selecting random points of the data to calculate possible structures. Such an approach with the decision algorithm is described in [3].

3.1. Second Derivative Methods

The first order derivatives of an image store the information about the magnitude and the gradient of the image. The second order derivatives additionally store the information about the shape. The detection of structures is based on the assumption that a blood vessel's profile in the cross-section is of a Gaussian shape. Furthermore, the intensity does not change much with the propagation of the vessel throughout the volume. The second order derivatives allow us to detect blood vessels, because we are provided with the information on the object curvature. A blood vessel is assumed to have a zero curvature along its center line and a large curvature in the section direction perpendicular to the z-axis.[4] The second derivatives are often organized into a Hessian matrix H. The Hessian matrix carries the information of the image regarding its magnitude, shape and orientation.[7] The information that is carried is shown in figure 3.1. Unfortunately, the use of the derivatives enhances the noise in the image. The second order derivatives approximately twice as much as the first order derivatives. To supress the noise a filter is used as mentioned in Chapter 4.

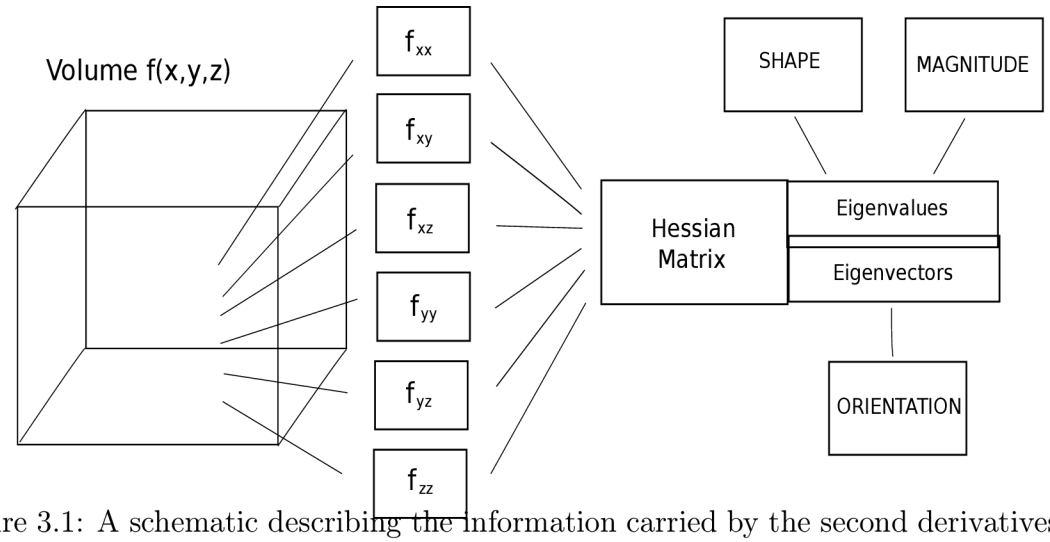


Figure 3.1: A schematic describing the information carried by the second derivatives and Hessian matrix

4. The Hessian Matrix Eigenvalues Method

This chapter intorduces the method used for the segmentation of tubular structures. This algorithm is based on the eigenvalues of the Hessian matrix. Following sections describe how the matrix is obtained from a discrete data set, the eigenvalues meaning and the evaluation of acquired results.

4.1. Hessian Matrix

Hessian matrix is the matrix of second derivatives and its form in a three-dimensional space is

$$H = \begin{pmatrix} f_{x,x} & f_{x,y} & f_{x,z} \\ f_{y,x} & f_{y,y} & f_{y,z} \\ f_{z,x} & f_{z,y} & f_{z,z} \end{pmatrix} \tag{4.1}$$

where $f_{i,j}$ is a partial derivative of a function *f(x,y,z)* with respect to i and j where $i, j \in \{x, y, z\}$.This matrix is obtained by drawing the partial derivatives of the function. However with the discrete set of data such acquisition is not possible. For the construction of a Hessian matrix of a discrete data set, numerical differentiation is applied. [1] For a partial derivative of a function *f(x,y,z)* the central difference formulas can be used. For the first partial derivative following formulas apply

$$\begin{aligned} f_x(x_i, y_j, z_k) &\approx \tfrac{1}{2h_x}(f_{i+1,j,k} - f_{i-1,j,k}) \\ f_y(x_i, y_j, z_k) &\approx \tfrac{1}{2h_y}(f_{i,j+1,k} - f_{i,j-1,k}) \\ f_z(x_i, y_j, z_k) &\approx \tfrac{1}{2h_z}(f_{i,j,k+1} - f_{i,j,k-1}) \end{aligned} \tag{4.2}$$

where f_n is a partial derivative with repect to n, h_n is the spacing in direction of n for $n \in \{x, y, z\}$ and $f_{i,j,k}$ is $f(x_i, y_j, z_k)$.
The second partial derivatives are

$$\begin{aligned} f_{x,x}(x_i, y_j, z_k) &\approx \tfrac{1}{h_x^2}(f_{i+1,j,k} - 2f_{i,j,k} + f_{i-1,j,k}) \\ f_{y,y}(x_i, y_j, z_k) &\approx \tfrac{1}{h_y^2}(f_{i,j+1,k} - 2f_{i,j,k} + f_{i,j-1,k}) \\ f_{z,z}(x_i, y_j, z_k) &\approx \tfrac{1}{h_z^2}(f_{i,j,k+1} - 2f_{i,j,k} + f_{i,j,k-1}) \end{aligned} \tag{4.3}$$

And the mixed partial derivatives

$$\begin{aligned} f_{x,y}(x_i, y_j, z_k) &\approx \tfrac{1}{4h_x h_y}(f_{i+1,j+1,k} - f_{i-1,j+1,k} - f_{i+1,j-1,k} + f_{i-1,j-1,k}) \\ f_{y,z}(x_i, y_j, z_k) &\approx \tfrac{1}{4h_y h_z}(f_{i,j+1,k+1} - f_{i,j+1,k-1} - f_{i,j-1,k+1} + f_{i,j-1,k-1}) \\ f_{z,x}(x_i, y_j, z_k) &\approx \tfrac{1}{4h_z h_x}(f_{i+1,j,k+1} - f_{i-1,j,k+1} - f_{i+1,j,k+1} + f_{i-1,j,k-1}) \end{aligned} \tag{4.4}$$

And because the calculated differences are the same the assumption of
$f_{x,y} = f_{y,x}, f_{y,z} = f_{z,y}, f_{z,x} = f_{x,z}$, can be made. Therefore is the Hessian matrix a symmetrical one and can be rewritten as

$$H = \begin{pmatrix} f_{x,x} & f_{x,y} & f_{x,z} \\ f_{x,y} & f_{y,y} & f_{y,z} \\ f_{x,z} & f_{y,z} & f_{z,z} \end{pmatrix} \tag{4.5}$$

This matrix is calculated for each voxel of a data set where $f(x, y, z)$ represents the intensity of each voxel.

4.2. Eigenvalues of the Hessian matrix

Let A be a linear transformation represented by the matrix A. The eigenvalue λ of an eigenvector $v \in R^n$ is a scalar such that

$$Av = \lambda\ v \tag{4.6}$$

The eigenvalues of a matrix can be calculated by substracting $\lambda\ I$ from the matrix of the linear transformation, I being the identity matrix, and computing the determinant of this expression. Let H be the Hessian matrix of a voxel, then following applies

$$(H - \lambda\ I)v = 0 \tag{4.7}$$

A linear system of equations that is represented by the expression 4.7 has a non-trivial solution if

$$det(H - \lambda\ I) = 0 \tag{4.8}$$

This equation is the characteristic polynomial of H.

By obtaining the eigenvalues of the Hessian matrix a new data volume is computed. Each voxel of this new volume consists of three parameters being the three eigenvalues of the calculated Hessian matrix. These eigenvalues $\lambda_1, \lambda_2, \lambda_3$ may however be permuted and the method must therefore implement the order of those parameters.

As mentioned in Chapter 2 the structures that are segmented are of a tubular shape. For the detection of objects with such properties, the rules of eigenvalue analysis can be applied. The following table shows image structure orientation based on the Hessian matrix eigenvalues. [4]

Table 4.1: Structure orientation dependancy on the eigenvalues

λ_1	λ_2	λ_3	structure orientation
L	L	L	noise
L	L	H	sheet-like structure
L	H	H	tubular structure
H	H	H	blob-like structure

where L stands for low and H for high value of the eigenvalue when sorted $\lambda_1 < \lambda_2 < \lambda_3$ and then taken as absolute values. By applying these rules onto a volume data set the signum of the eigenvalues of each voxel determines also the character of the data. Considering the high values of the parameter λ to be negative, the detected structure is of a higher intensity in comparison to the background and vice versa.

The desired structure orientation for the purpose of blood vessel segmentation is the set of eigenvalues with $\lambda_1 \approx 0, \lambda_2$ and λ_3 of a high negative value. Voxels that show this property are then detected as a tubular structure and segmented for the new volume dataset.

4.3. Processed 3D Image

The information that is drawn out of the sorted eigenvalues can be processed in several ways. In this section suggestions how to process the volume further are made. These different approaches are based on the desired result of the segmentation and can be also combined depending on the implementation.

4.3.1. Eigenvalue Threshold Image

To get a segmented blood-vessel tree is the desired result of the algorithm. A binary image representing the arterial structures can be calculated based on the eigenvalues sorting. If the set of eigenvalues for each voxel resembles the pattern for tubular structures in 4.2, voxel is distinguished from the background. The risk that comes with this method is a possible error of wrong-thresholding. This can occur if the original data set does not have a good resolution and therefore the error arises with the computation of the Hessian matrix.

4.3.2. Parametric Image

A binary image provides information about the blood vessel tree. However with medical data it can be desirable to obtain also information about the neighbourhood tissue and display the data that 'almost' made it to the binary image. Therefore forming a volume data set based on a parameter drawn out of the eigenvalues can be useful. The new volume data set is based on the resemblance to the properties of a tubular structure. The blood vessel tree is highlighted based on the fact that it corresponds to the tubular shape the most. Not that distinctively can be also other structures shown in the new image too. Such voxels acquire a value that is not extreme to be marked as a vessel but not insignificant to be marked as a background. Therefore a 3D image is shown where any tubular structures are marked with new intensities. Also a transparancy of unwanted objects can be implemented. The less transparent an objects appears the lower the intensity in the original image is. Therefore voxels with higher confidence in their 'vesselness' are shown as present structures. On the other hand, objects with low resemblence to the vessel shape are displayed less distinctively.

5. Implementation with MATLAB

The method of Hessian matrix eigenvalues implemented in MATLAB is described in this chapter. The goal of this realization is to segment the blood-vessel tree and to display the segmented image. Moreover, some examples of the realization of this method are displayed and further suggestions are made. The following figure 5 shows the block diagram of the image processing algorithm.

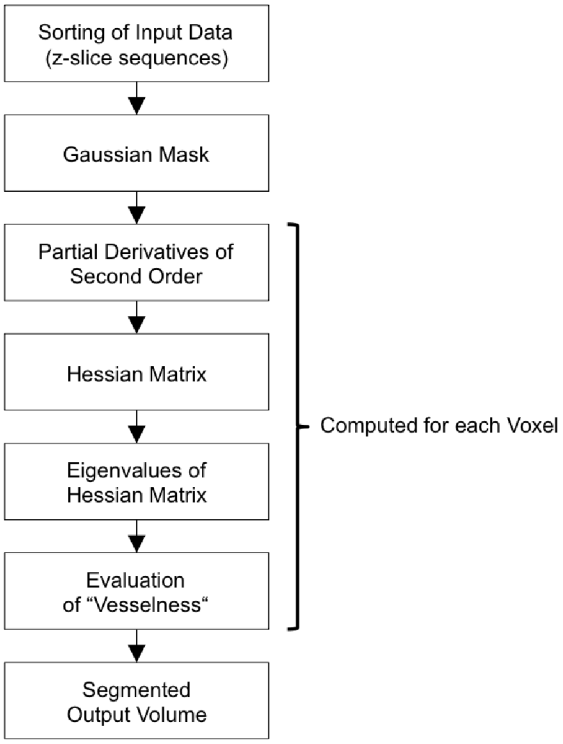


Figure 5.1: Block diagram of the implemented method

5.1. Input of the Data

The data set provided comes from a GE MRI scan as mentioned previously. This data is in DICOM image form. Each slice of the volume is provided with detailed information on the acquisition sequence, the spacing and the position of the slice in the whole image. From each sequence only the z-slices are taken into further computations based on the properties the vessels show in this direction.
The whole-body acquistion consists of roughly six uniform volumes as a result of the maximum field of view of the GE MRI. A typical voxel size is $1.875 \times 1.875 \times 4.5$ mm. The neighbouring volume sets are partially overlapping. In this region, they do not generally share the same z-slice coordinate. For optimal time efficiency, each volume is processed separately. Combining the volumes would lead to the need of adapting convolution masks (varying spacing) or to a denser grid created by interpolation to ensure a uniform voxel size. In case of multiple images with the same z-slice coordinate, an average is computed. Averaging may also help to eliminate artefacts.

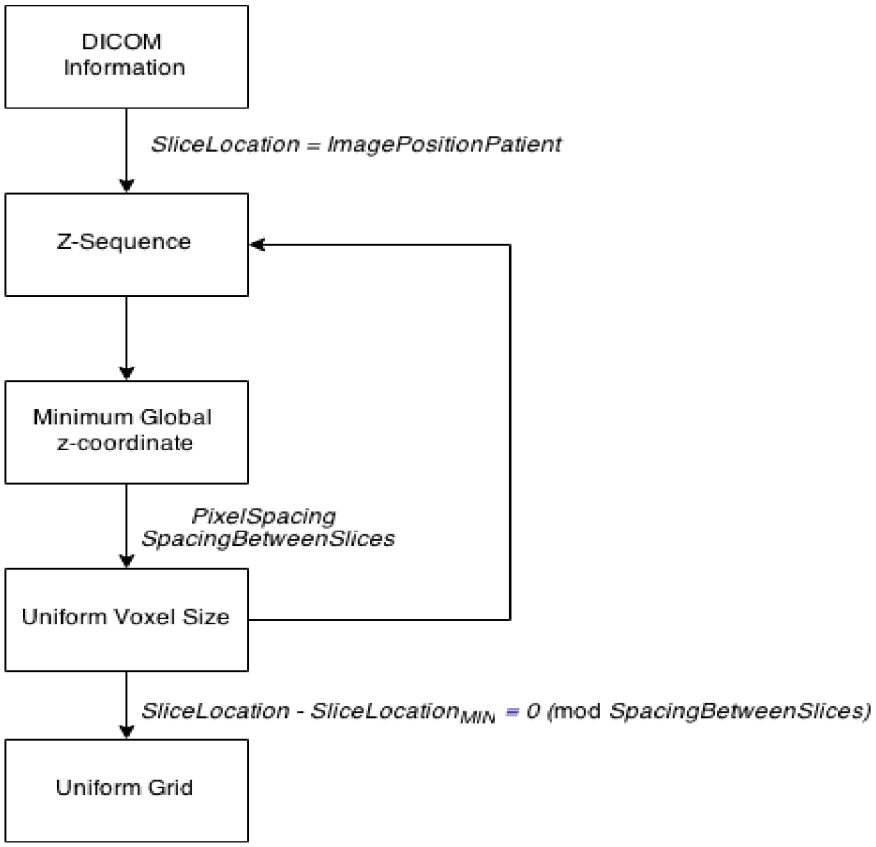


Figure 5.2: Schematic of the sorting of the input data

The sorting schematic is shown in the figure 5.1. In this description, the terms that represent fields of structure of DICOM information are written in italic. They are the names that enable the access to those fields if existent. For the acquisition of z-sequences only, the comparison of *SliceLocation* and z-coordinate of $ImagePositionPatient$ is made. In case of matching values (global z-coordinates), the image belongs to a sequence containing slices in desired direction and therefore is stored with the information as follows: name, voxel size, global z-coordinate. The information about the voxel size can be extracted out of the fields *PixelSpacing* and *SpacingBetweenSlices*. The sorting starts at the slice with the minimum global z-coordinate. Furthermore, the slices are separated according to the spacing. Volumes with the same voxel size do not generally represent a uniform grid if combined together. To ensure consistent structure, the global z-coordinates are compared to the z-coordinate of the starting point (in 5.1 described as $SliceLocation_{MIN}$). If a slice is a part of the uniform structure represented by the starting point in the global space its z-coordinate lies on the grid with the origin at the starting point. For graphical explanation, see 5.1. When a sigle uniform volume is extracted the recursive function is applied onto the remaining slices.

The uniform volumes that are smaller than 4 slices are discarted. They bring no significant information to the segmentation because the further processing that implements convolution with a 3×3 masks can be accurate only for the inner volume. Likewise the structures that are uniform however are partially missing some slices. In further process-

ing, those slices would be incorrectly visualized and they would additionally influence the segmentation process.

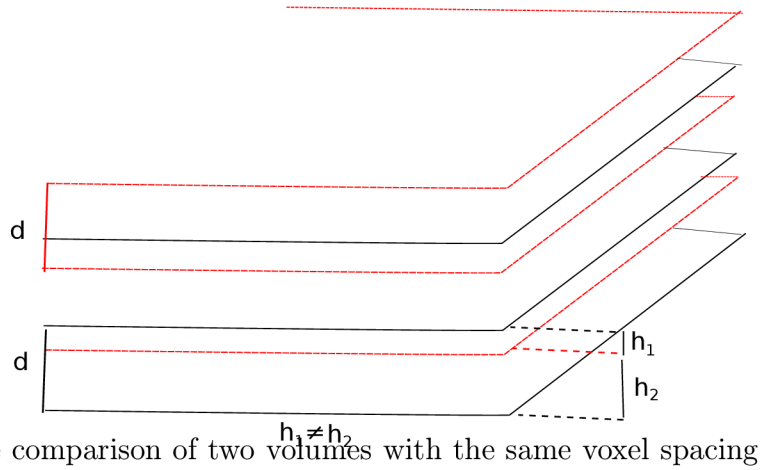


Figure 5.3: The comparison of two volumes with the same voxel spacing that cannot be combined into one uniform volume

5.2. Pre-processing of the Data Volume

The preprocessing of the data volume is done with a Gaussian mask that causes the smoothening of the image. This preprocessing is necessary due to the second derivative computation that brings considerable noise into the image. The Gaussian mask is modelled according to the Gaussian distribution that has the 2-D form

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{5.1}$$

where σ is the standard deviation of the distribution, the mean of the distribution is $(0,0)$. Figure 5.2 shows the one dimensional Gaussian distribution for different σ values.

The use of an isotropic mask is possible assuming that the vessel cross-section is nearly circular and of a Gaussian shape. Additionally, the intensity should not change with the propagation of the vessel throughout the volume. The size of the mask takes in consideraton the resolution of the image and its size. Moreover, the value of σ is also derived from the radius of the arteries so the smoothening does not repress the intensity of the structure to be segmented. The radius size of main arteries is in the range 1 - 2 pixels. It is desirable to set the mask size to at least three times the deviation σ to approximate this distribution smoothly enough in a discrete space. Using a mask of the size 6×6 however supresses the smaller arteries and often decreases their intensity to the level that there are no longer visible in the segmented volume. For the purpose of this segmentation, the original volume is convolved with a Gaussian mask of the size 3×3. The figure 5.2 shows the 3×3 Gaussian mask. The weights of a Gaussian mask add up to one so the overall intensity stays unchanged.

Figure 5.2 shows the difference of the smoothening according to the size of the mask. Notable is the intensity of the vessels after the application of the mask. The table 5.2 demonstrates the contrast acquired with masks used also in 5.2. The intensity values do not differ greatly, nonetheless the figures support the theory of the small arteries suppression. Moreover, the smoothening of the mask of size 6×6 causes smaller contrast

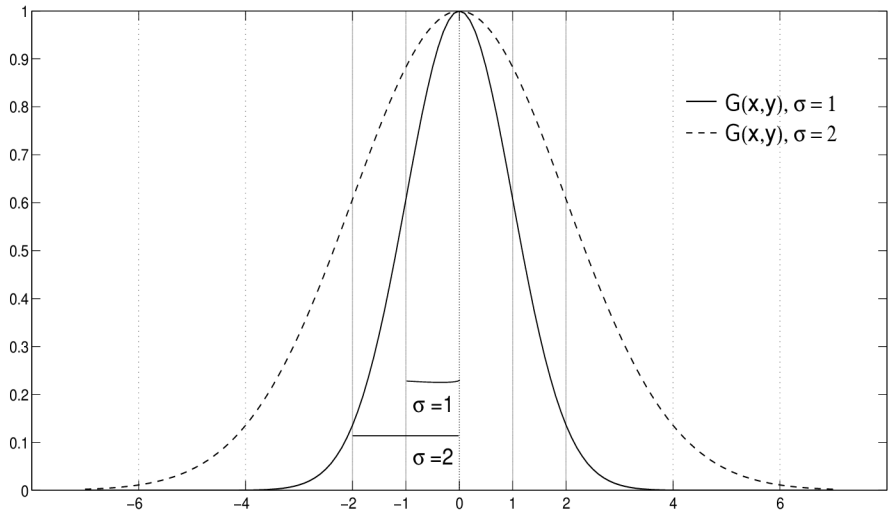


Figure 5.4: One dimensional Gaussian distribution for $\sigma \in \{1, 2\}$

0.0751	0.1238	0.0751
0.1238	0.2042	0.1238
0.0751	0.1238	0.0751

Figure 5.5: Gaussian mask of size 3 $\times$ 3

between the vessel cross-section and the surrounding. That causes additional influence on the perception of the intensity.

Table 5.1: The influence of smoothening on small arteries; values represent normalized intensities

Mask Size	Point 1	Point 2	Point 3	Point 4
No mask	0.584	0.416	0.244	0.506
3×3	0.290	0.267	0.153	0.345
6×6	0.235	0.231	0.129	0.302

5.3. Hessian matrix and Eigenvalues

The Hessian matrix is computed according to chapter 4. To compute partial derivatives of each voxel, mask operators are created. From symmetry of the Hessian matrix, only six operators are needed and realized through convolution. For non-mixed second derivatives, a mask seen in figure 5.3 on the left is applied. For mixed derivatives, a similar mask is created according to 4.4 (5.3 right). Each mask is additionally adjusted to the voxel size

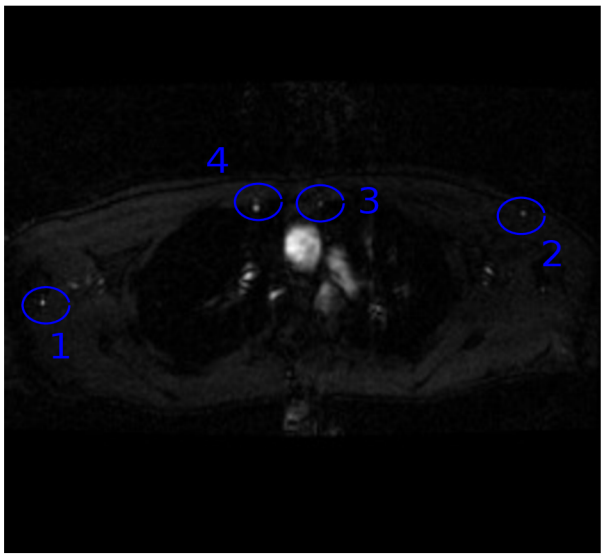


Figure 5.6: The original image before smoothening with highlighted arteries picked for table 5.2

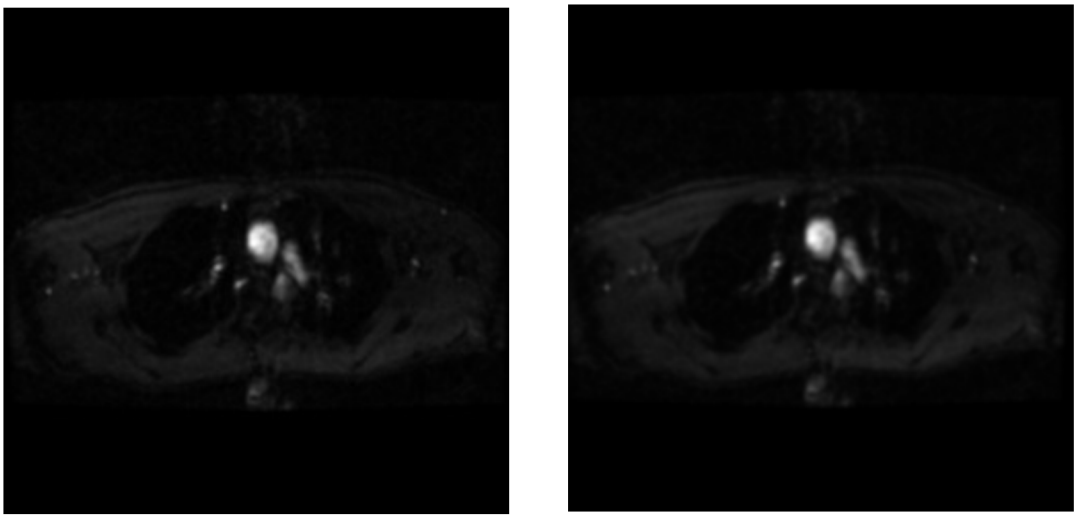


Figure 5.7: The application of the Gaussian mask; mask size 3×3, mask size 6×6

to consider different distances in each direction. For exact adjustment see equations 4.3. For each voxel, a Hessian matrix is created out of the computed partial derivatives of the second order. From the Hessian matrices the eigenvalues are computed and sorted in the descending order for each voxel. For a sorted vector of eigenvalues $u = (\lambda_1, \lambda_2, \lambda_3)$ of a voxel new volumes are created. The discussion of eigenvalues requirements is made in the following section 5.3.1.

0	1	0
0	-2	0
0	1	0

1	0	-1
0	0	0
-1	0	1

Figure 5.8: Mask operators; non-mixed partial derivatives (left), mixed partial derivatives (right)

5.3.1. Processing of the Eigenvalues

There are several ways how to create the new 3D image with segmented blood vessel tree. The suggested binary image mentioned in chapter 4.3.1 is made based on an assigned threshold for each of eigenvalues $\lambda_1, \lambda_2, \lambda_3$. For binary output image a combination of values for λ_1,λ_2 and λ_3 is set to wheteher mark a voxel as a vessel or as a background. Due to different characteristics of each acquisition sequence, implementing this method with only one threshold does not bring any satisfactory results. Making the threshold a function dependent on properties of each acqusition sequence can be considered for further improvement and will be discussed as an option for a user to select.
The segmentation output may be also represented by each of the eigenvalues. The output volumes are based on the three values of the sorted vector $u = (\lambda_1, \lambda_2, \lambda_3)$. Each eigenvalue represents some properties of the segmented volume. The prosessed volumes represented by eigenvalues are normalized.

λ_1 As seen on following figure 5.3.1, the highest intensity of voxels is in the surrounding area of tubular structures. More specifically, the arteries.

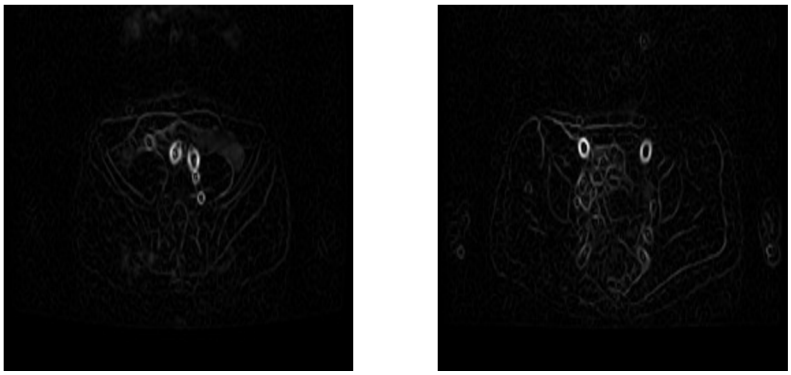


Figure 5.9: The parametric image based on the eigenvalue λ_1

λ_2 Parametric image based on the second eigenvalue results in a 2-D image where tubular structures are enhanced. Additionally to the desired segmentation, some structures like edges of the limbs or chest. It is displayed in the combination with the last eigenvalu in the figure 5.10.

λ_3 Segmented volume reresented by the eigenvalue λ_3 enhances the desired tubular structures. However, the surrounding appears to be a noisy non-uniform image. The result in 2-D can be seen in figure 5.10 in the combination with λ_2.

The combination of the eigenvalues that draws the expected output volume is made as a parametric image and is described below.
In code *loadFile.m* is in the sorted vector $sortVal = (\Lambda_1, \Lambda_2, \Lambda_3)$ the last eigenvalue $\Lambda_3 \approx 0$. The reason is the relationship between the tubular structure (bright) and the background (low intensity). This leads to the values of Λ_1 and Λ_2 to be negative. In this text, the theoretical assumption applies 4.2 for general values of the eigenvalues.

The following figure shows a 2-D image of thorax. The output volume that assignes the intensity of the voxels according to the function is shown in the figure 5.10.

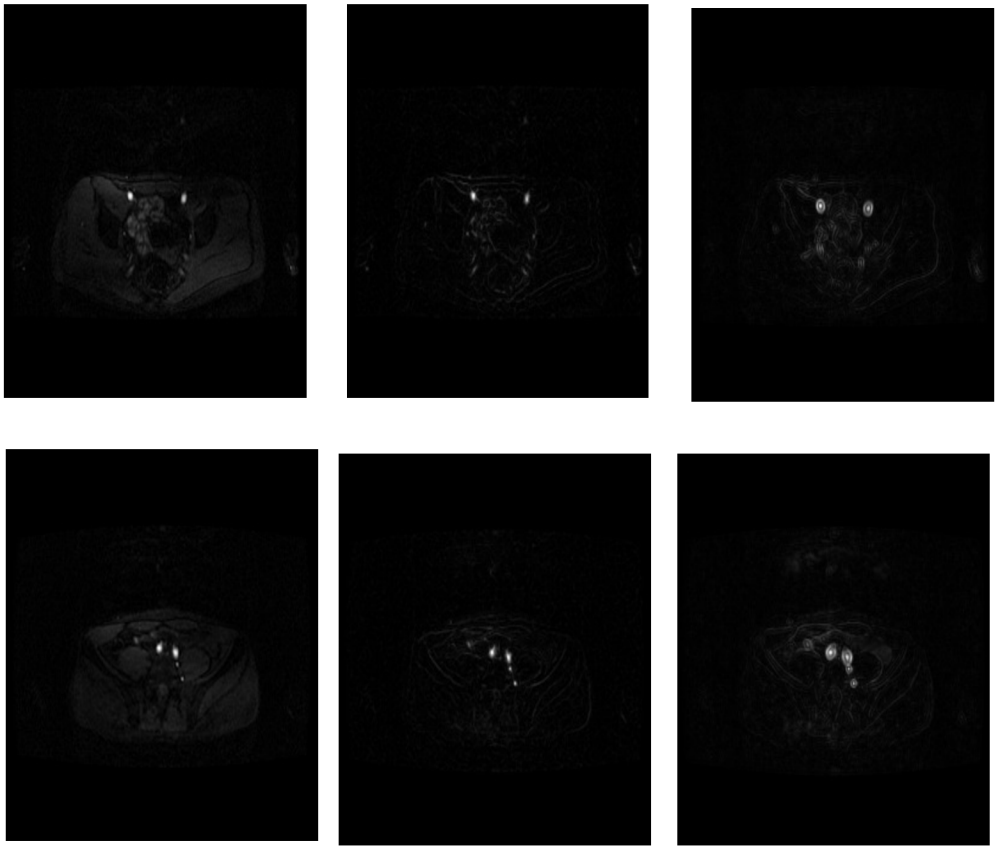


Figure 5.10: Original slice of the MRI acquired volume (left) and the processed output λ_p paramteric image (center), image with function including the omitted eigenvalue λ_1 (right)

For the parameter image method described in 4.3.2, an output image, $\lambda_p(\lambda_1, \lambda_2, \lambda_3)$, is assigned a new value. The parameter represents the vesselness of the original voxel and is the output of an assigned function. For the λ_p function the following is applied

$$\lambda_p = \sqrt{\lambda_2^2 + \lambda_3^2} \tag{5.2}$$

where λ_p is the λ -parameter value and λ_2 and λ_3 are negative values of the eigenvalues. The λ_p is a function of only two variables λ_2 and λ_3. The absence of λ_1 is based on its insignificance in the region of the inside of a tubular structure where $\lambda_1 \approx 0$. Moreover, its addition to the function λ_p in 5.2 would cause a seemingly blurred image in the area around the arteries. However, the information the λ_1-segmented volume provides can be

benefitial in some problematic areas. This is discussed more thoroughly in section 8. The combination of the values λ_2 and λ_3 brings the enhancement of the inner region of the original tubular structure and supresses the unsought detection of the body's edges that is only present in the λ_2 based volume. The information of the body shape is nevertheless still visible and allows a better orientation throughout the volume.

5.4. Data Transfer to VTK

The visualization of the processed data is done using a VTK library. As described in section 5.1 the whole-body sequence consists of more uniform volumes that are processed separately. Also the storage of the data in VTK files is done separately for each of the volumes. The normalized data is recalculated into the *unsingedchar* format that is in the range $(0, 255)$. The information that is stored additionally to the voxel intensity is the voxel size and the global z-coordinate for correct position of the volume within the global 3-D space.

6. Visualization with VTK

The Visualization Toolkit is an open-source software for visualization. It consists of a C++ class library, however interprets also Python that is used for the visualization in this work. Moreover, the VTK supports GUI toolkits, such as Qt that is used for the interface design. The visualization process is elaborated in this chapter. Additionally, the Qt design is introduced and the user's interaction is described. Figure 6 shows a whole-body segmentation of the arterial system with and withou the body outline.

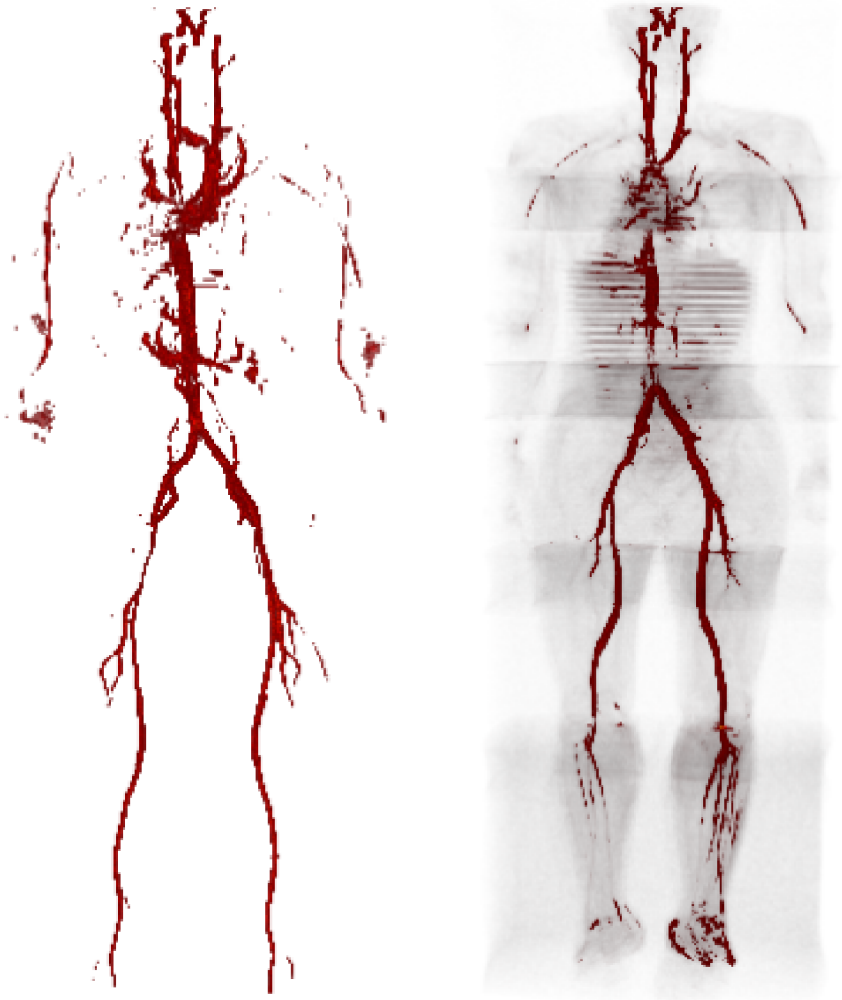


Figure 6.1: The arterial system of the whole body shown with and without the body outline

6.1. Data Input

The processed data is stored in the form of a VTK file. VTK supports several formats. The segmented volumes are stored as a *structuredpoints* formation. This form enables to visualize structured grid of user defined uniform size. Additionally, the origin is defined to set the position in the global volume. These information are extracted from DICOM

information during the processing in MATLAB and saved into the *VTK* header of each file.

6.2. Visualization

The visualization is done with the VTK software and is encoded in Python. The data is loaded from the *VTK* file as structured points dataset. The data undergoes a process of visualization such as mapping and rendering. For more elaborated information see the attached CD with the file *program.py*.

Properties of the visualized volume

- *Color Transfer Function*
  The color transfer function enables the highlighting of regions of varying intensities with different colors. The goal is to enhance the segmented vessel tree against the chosen background. Color choice can influence the information significantly. Moreover, a balance between the desired segmented structure and the body outline should be found. The blood-vessel tree without the information about the position within the body can provide less information. Figure below 6.2 presents the results for two different color choices and what influence they have on the visualization. The arteries are usually represented by a small interval of intensity values. The

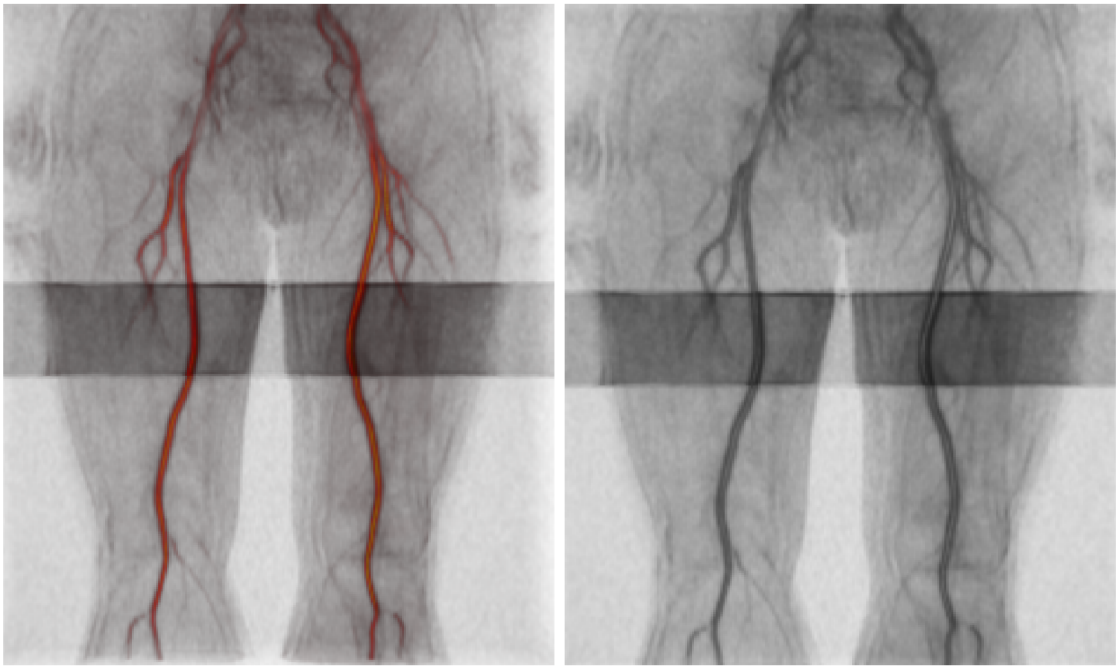


Figure 6.2: Visualization with different transfer color functions; *heat* (left), black and white (right)

  same applies for the body outline. To distinguish the difference between these two it is favourable to use a dynamic color scale.

- *Opacity*
  The opacity enables the suppression of unwanted segmented objects of low intensity. A very typical object that is set to be transparent is the outline of the human body.

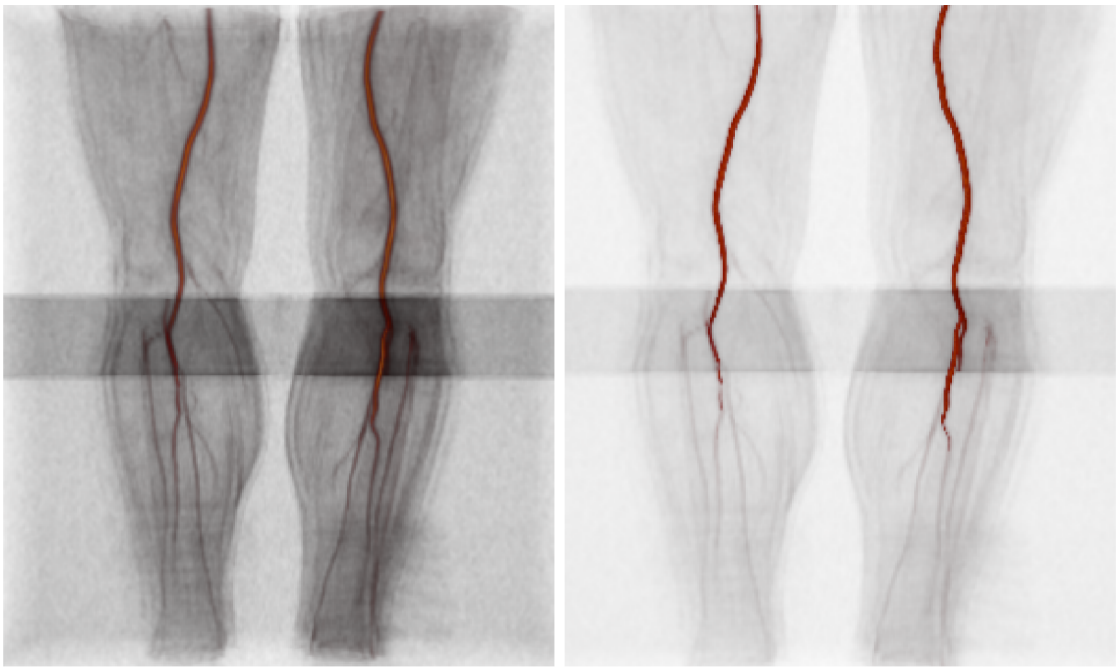


Figure 6.3: The volume visualized with low transparency (left) and with high transparency (right)

The request for a good *piecewise* function representing the volume's opacity is to set the outline to be transparent with slight visibility and to set the arteries to be opaque. The assumption that gives the means for this separation is different intensities of those two structures.

Figure 6.2 shows volumes with different opacities. The volume with higher opacity helps visualize small arteries within the legs. The volume with lower opacity enhances the main arteries.

6.3. Graphical User Interface

The GUI for the user's interaction is designed through *QtDesigner*. Qt is a cross-platform application framework used for the development of application software. The design of GUI is exported as an *ui*-file and can also be generated or converted into the Python encoding. The framework is saved as a separate file (*.py*) and is exported into the main program (*bc.py*) for the desired layout.

6.3.1. Interaction

The interaction with the volume, such as zooming, rotating and shifting is implicitely set as a function of the *Render Window Interactor*. Additionally, the VTK library enables to *pick* points within visualized volume. Depending on the type of the *Picker* the information about the volume region can be stored. The user obtaines two information from the interaction with the data

- *Point Coordinates*
  Point coordinates are the global coordinates of the picked point. The storage of

the DICOM information and its correct interpretation is necessary for a precise visualization. The global coordinates of a point are saved in a table that can be accesed and processed further as described in the following section 6.3.2. The picked point is the first voxel (point of a structured points structure) that intersects the picker's path.

- *Volume Set*
  The visualized blood-vessel tree is composed of several uniform volume sets. The picker enables to establish the volume set in which the point is being picked. Moreover, the extraction of the z-slice is possible as *an extraction of volume of interest*. This allows the user to see exact location of the artery in the inner body.

6.3.2. Measurements of Arteries

The lengths of arteries are calculated as euklidiean distances of two points in a volume. Therefore the distance between point A and B with the coordinates $A = (a_1, a_2, a_3)$ and $B = (b_1, b_2, b_3)$ is computed according to the formula

$$\bar{AB} = \sqrt{(b_1 - a_1)^2 + (b_2 - a_2)^2 + (b_3 - a_3)^2} \tag{6.1}$$

The length of an artery is a linear approximation of its propagation throughout the volume. The points for this calculation are drawn out of a table where the global coordinates of the breaking points of this approximation are stored. The points are picked manually by the user.

7. Results: Artery Analysis

To demonstrate the outcome of the GUI, an artery analysis is provided. The measurements of lengths are computed as linear approximation of the artery. The table 7 presents the lengths of the main chosen arteries to be measured for demonstartion in this work. The exact theoretical location of those arteries can be seen in figure in 10.1. The lenghts are calculated for 3 patients and compared to the results provided from a doctor. Those results were manually computed and the paths were derived as an approximation of the position of the edge points of each artery.

Table 7.1: Presented results of the measurements of the arteries in mm

Artery	Patient 1	Patient 2	Patient 3	Expected
arteria femoralis dextra	402.6	400.3	395.2	416
aorta abdominalis	97.8	123.1	100	112
arteria carotis communis dextra	103.4	75.6	81	99
aorta thoracia	205.7	200.1	185	216

The picked paths are demonstrated in the following figures. Results are mostly visual based on the variety of the patients. The subjects are of different ages and heights. A comparison with theoretical data is practically impossible due to the lack of the information for such diverse group of people.

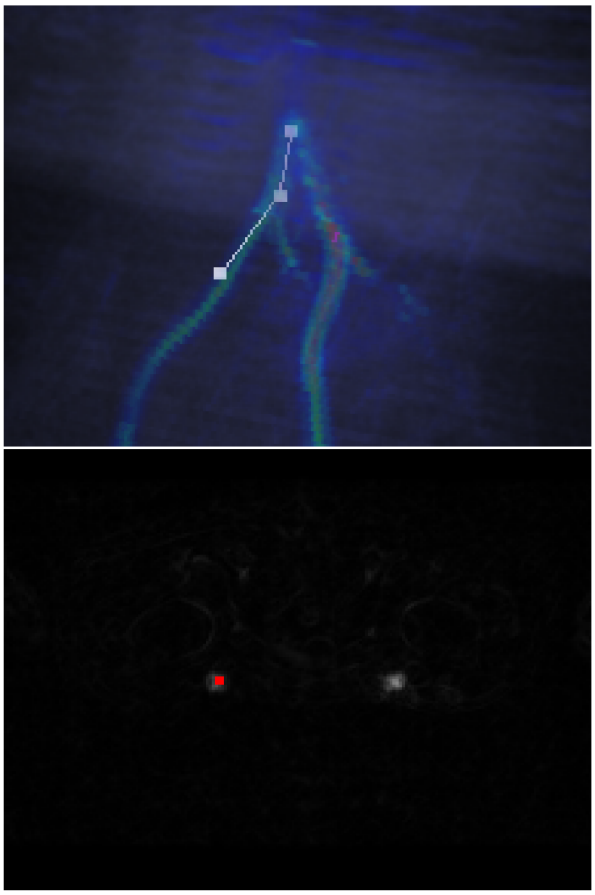


Figure 7.1: Images show possible picking in different windows

8. Limitations of Implemented Segmentation and Visualization

The intensity of the voxels in the output volume is assigned based on the function described in 5.3.1. The acquired data present some weaknesses. Those defects come from the acquisition process as well as from the limitations of the GE MRI. The problematic regions are discussed below and some modifications are suggested to allow a better and more precise results.

8.1. Data Input

The possibilty to make this method automatic is limited by the data that is provided. The file that represents the patient consists of around 1500 DICOM images. The sorting process organizes the file and draws out the uniform z-slice sequences. The whole-body acquistion is also composed of a prescan sequence and other non-angiographic data. The impossibility of the extraction of the right uniform volumes that were acquired as an agiographic sequence is not possible due to the lack of information on the acquisition mode in the DICOM information. The user should therefore have some information on the desired segmented volume and should have a basic knowledge of the arterial system in the human body.

Furthermore, there are several weaknesses some of the patients' folders show.

- *Loss of Data*
  In the case where a partial volume is lost, an approximation for the measurement can be done by the user. This approximation is linear according to the calculation of the distances. This estimation resembles an interpolation of the missing segment inbetween two volumes, therefore can be sufficient. In cases where a greater part of the volume is absent, the propagation of the artery is probably non-linear and cannot be estimated by the user.

- *Shift in Volume*
  A shift in volume can be caused by the movement of the patient. This problematic requires a separate processing where global coordinates of the shifted segment are redefined. This demands a manual revision of the *VTK* file. The corrected global coordinate could be matched by the user or a matching algorithm could align those separate volumes.

8.2. Acquisition Process

One of the weaknesses of some of the acquired data is a very high intensity of the outline. This leads to a processed image with highlighted edges that are of a similar intensity as the desired segmented volume. The opacity of these regions cannot be set separately resulting in an obstacle for the visualization of the inner part of the body. The possible solution is the option for the user to display separate slices. That helps to distinguish the arteries from the faulty segmented outer structure. The disadvantage of this problematic volume is the inability of its improvement due to errors made during acquisition.

Figure 8.2 shows the faulty area as well as a correctly acquired volume. To avoid the incorporation of this volume into the dataset, the user must exclude the VTK file from the global visualization.

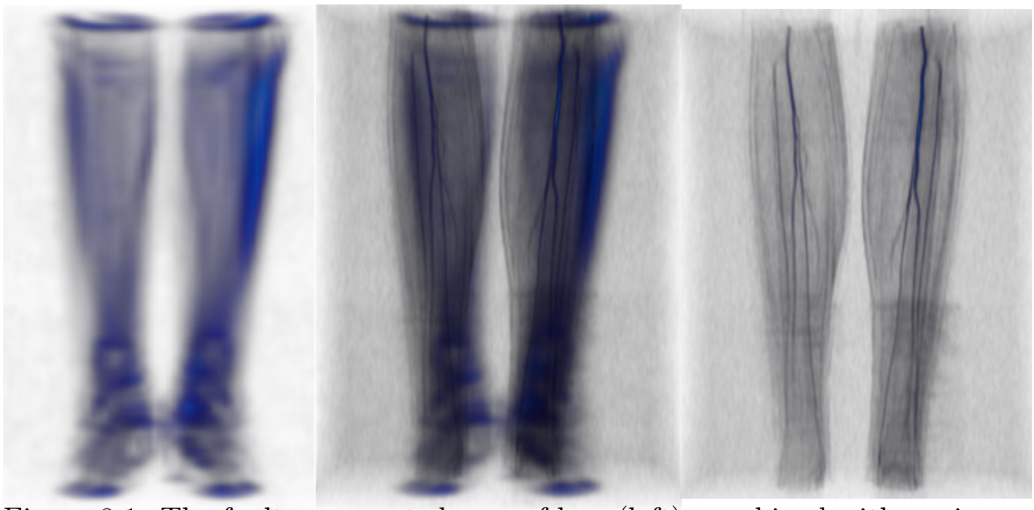


Figure 8.1: The faulty segmented area of legs (left); combined with angiographic volume (center); angiographic volume (right)

8.3. Field of View Problematics

The whole-body acquisition sequence can be problematic for people with bigger body statures. In this case, the field of view of the GE MRI Discovery is $48 \times 48 \times 48$ cm. This can lead to problems in the upper body area. Two of the occuring limitations are desribed in following sections.

8.3.1. Arms

The arms can be a problematic region. The arteries are poorly visible due to the MRI field of view. Patients whose data is provided are of a different age and height. The body stature differs and can reach the MRI limitations considering the $x-y$ plane. The arms of such patients can be strapped to the body to ensure the inclusion in the acquired volume. This policy must be performed during the acquisition process, otherwise the data cannot be obtained. The arms region is shown in figure 8.3.1.

8.3.2. Aliasing

The aliasing in MRI data occurs when the FOV is smaller than the data. An artefact appears outside the body outline and can cause nontransparent voxels to block the desired volume structure. This region is present very often around the chest area due to its size. The solution can be the extraction of a volume of interest or the 2D visualisation of a slice. Figure 8.3.2 shows a region with lower visibility as a result of aliasing.

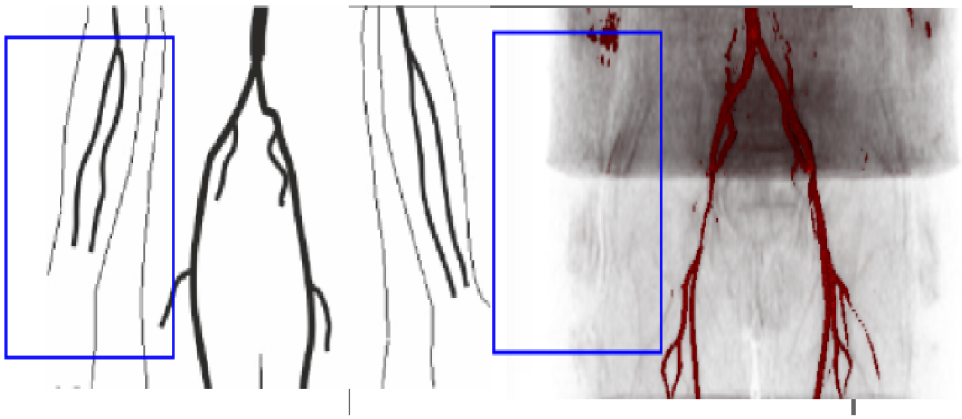


Figure 8.2: The comparison of desired segmentation to the segmented volume; difference highlighted in blue

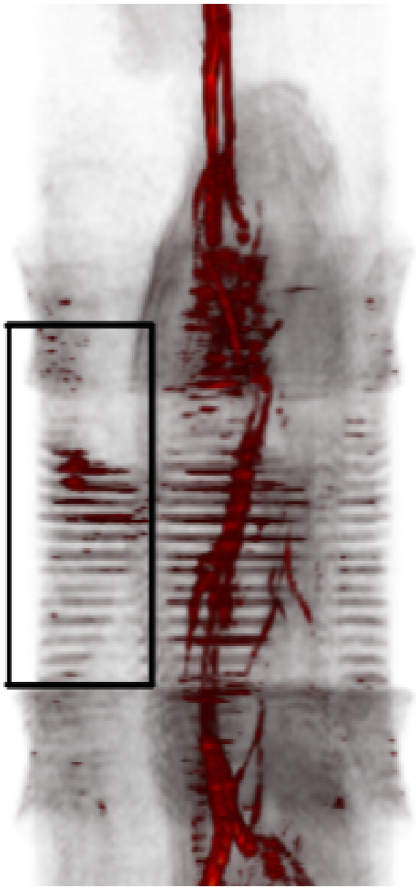


Figure 8.3: The result of aliasing highlighted in a rectangle

8.4. Aortic Region

The aortic region is problematic due to its anatomy. The angiographic acquisition deals with the separation of vessels (arteries, veins) based on the direction of the flow as described in chapter 5. The arteries' intensity is enhanced due to the blood flow direction defined for the z-axis. The aortic arch propagates differently (as seen in figure 8.4) and

therefore cannot be enhanced and segmented clearly. The suggested solution is to visualize the parametric volume based on the λ_1 eigenvalue. The highlighted edges of the artery are clearer and show the propagation in the body more precisely. The comparison of the volumes can be seen in figure 8.4.

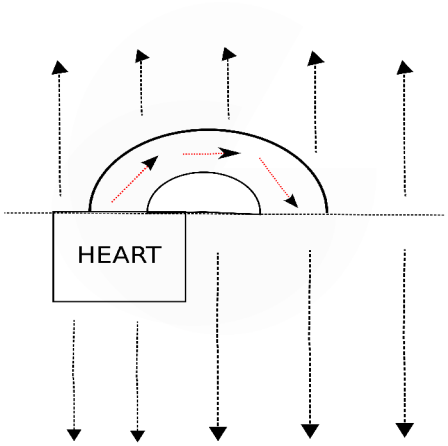


Figure 8.4: The blood flow direction in aorta in comparison to the set direction expected during angiographic acquisition

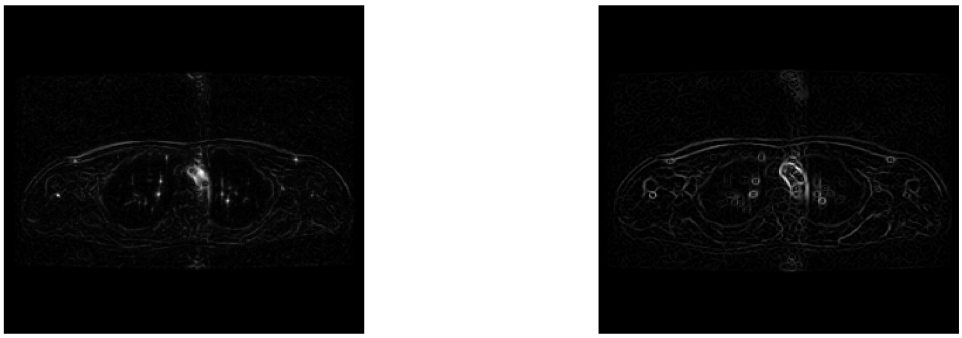


Figure 8.5: Aortic region visualized as a λ_1 parametric image in comparison to the λ_p image

9. Summary and Outlook

The method of segmentation achieved the desired image. Best results are acquired for the main arteries. There are also very precise requirements on the data. The acquisition process is essential for the successfull outcome of the processing. There are possible modifications of the Hessian matrix eigenvalue method. Those could most likely elaborate the function of the output intensity to acquire results of higher quality. Another improvement of the outcome could be the advancement of the visualization. The user could in the future choose from a wider set of functions or define his/her own. The focus of this chapter is the outlook on the measurements of arteries.

The linear approximation of the lengths of the arteries is done manually. An approach of automatic tracing of a structure could be implemented for further study. Tracing is difficult in parts of artery's bifurcations. It is therefore beneficial to implement a semiautomatic measurement method where the user specifies the region of interest (eg. the beginning and the end of a vessel). User's interaction could correct wrong paths or already specify several points of the artery for a more precise connection between artery's edges.

Tracing of the artery is based on its intensity in comparison to the background's intensity. The artery is traced in the direction of the next user's selected point. Neighbouring voxels are taken into consideration to search for a resembling intensity value. The volume of interest that is studied for the path is based on the assumption that big arteries are represented in the cross-section by circa 3×3 selection with the most common spacing $1.875mm$. Furthermore, VTK translates the information of voxel spacing onto the global coordinates. Therefore a different resolution of volumes is not a limiting factor. VTK enables the computation of the volume of interest from global coordinates and presents the possibilitz of acquiring the artery's cross-section in mm.

This method would enable the user to pick only necessary amount of points for the correct approximation. Those points do not have to be the edge points of a linear region, but rather significant points of the propagation. The benefit of the semiautomatic length measuring is the reduction of user's picking. The lengths should not differ greatly without the semiautomatic option in comparison to the manual picking if user picks the linear segments accurately. Additionally, with the semiautomatic function, the correction and review of the traced region by the user is recommended.

10. Conclusion

The goal of this thesis was the segmentation of the arterial system. Several methods were briefly described with the focus on the second derivatives. Moreover, the basic anatomy of the circulatory system was described to explain the basis for the segmentation, such as the tubular shape of arteries or the blood flow direction.

The method of segmentation was based on the Hessian matrix eigenvalues and their properties. The segmentation's limitations were discussed and some further suggestions were made. The possibility of improvement can also include a more complex second derivative processing that involves the eigenvector computation. The implementation involved three main focusses. Firstly, the appropriate loading of the input data and the sorting of separate acquisition sets. The assumption of the correct angiography acquisition is expected. Furthermore, the implementation of the Hessian matrix eigenvalue method. Prior to the convolution for the computation of the second derivatives, preprocessing in the form of smoothening was adjusted for better outcome. The segmentation method was implemented and brought sufficient results. Those could be demonstrated in the 2D slices. Last focus of the implementation was the visualization in 3D. The software VTK enabled the display of the segmented volume. Voxels were stored as structured points datasets representing volume with possible interaction. 3D visualization showed some imperfections of the method, such as artefacts or lost data. Mostly problems as consequence of the acquisition process or of the MRI limitations.

The last part of this thesis deals with the interaction with the volume set. The segmented arterial system is measured. The lengths are obtained as a linear approximation of the straight paths of the vessels. This application was demonstrated on few patients. The suggestions for further advancement were made for more time eficient and more precise picking.

Bibliography

[1] MOHLENKAMP, M. and YOUNG, T.: Introduction to Numerical Methods and Matlab Programming for Engineers. In *Numerical Differentiation.* Ohio University, Department of Mathematics. 2014.

[2] ROKYTA R., et al.: Fyziologie pro bakalářská studia. In MUDr. PAUL, T.: *Fyziologie srdce a krevního oběhu.*Praha: ISV Nakladatelství, 2008, p. 109-130. ISBN 80-86642-47-X.

[3] BEHRENS, T., ROHR, K., and STIEHL, S. H.,: *Using an Extended Hough Transform Combined with a Kalman Filter to Segment Tubular Structures in 3D Medical Images.* VMV. 2001.

[4] FRANGI, F. A., et al.: Model-Based Quantitation of 3-D Magnetic Resonance Angiographic Images. *IEEE Transactions of Medical Imaging.*, October 1999, vol. 18, no. 10, p. 945-948.

[5] LEI, T., et al.: Artery-Vein Separation via MRA - An Image Processing Approach. *IEEE Transactions of Medical Imaging.*, August 2004, vol. 20, no. 8, p. 689-692.

[6] OH, KWANG-MAN, et al.: A Segmentation and Abstraction of Blood Vessels from Volume Data for Surgical Simulation. *Int. Conf. on Artificial Reality and Telexitence*, ICAT. Vol. 99. 1999.

[7] QINFENG, L.: *Enhancement, Extraction, and Visualization of 3D Volume Data.* [Dissertations. No. 824] Linköping, Sweden: Institute of Technology, Linköpings Universitet, Department of Electrical Engineering, 2003. 213 s.

[8] GE Healthcare.*Discovery MR750 3.0T.* Waukesha, WI, USA: GE Healthcare, 2008. Print.

11. Appendix

11.1. Arterial System

The labeled arterial system is displayed as an aditional information for the expected outcome. Moreover, the labeled arteries are to be measured as a part of a future study. In this thesis only some of the main vessels' lenghts are presented to demonstrate possibility of the volume picking. To process a whole-body scan the suggested method for semiautomatic measurement should be implemented due to time efficiency and better approximation.

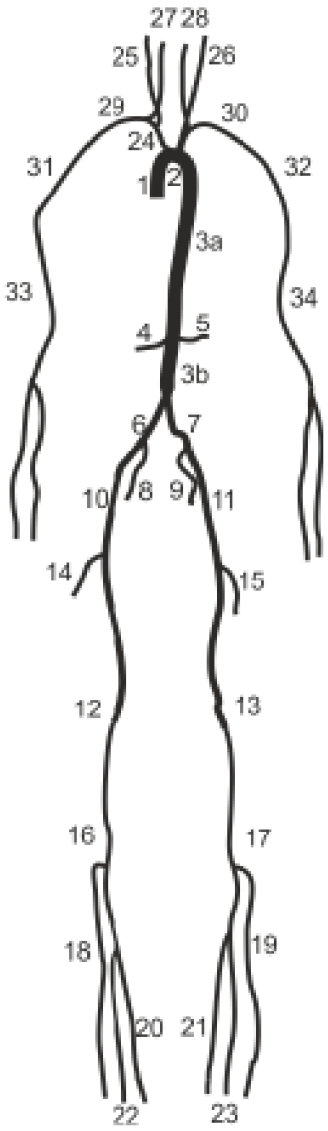


Figure 11.1: The desired arterial system to be segmented as a result of this work with labeled arteries

Table 11.1: Table with the arteries where numbers represent labels in figure 10.1

Nr.	Latin name
1	aorta ascendens
2	arcus aortae
3a	aorta thoracia
3b	aorta abdominalis
3	aorta descendens
4	arteria renalis dextra
5	arteria renalis sinistra
6	arteria iliaca communis dextra
7	arteria iliaca communis sinistra
8	arteria iliaca interna dextra
9	arteria iliaca interna sinistra
10	arteria iliaca externa dextra
11	arteria iliaca externa sinistra
12	arteria femoralis dextra
13	arteria femoralis sinistra
14	arteria profunda femoris
15	arteria profunda femoris
16	arteria poplitea
17	arteria poplitea
18	arteria tibialis anterior
19	arteria tibialis anterior
20	arteria tibialis posterior
21	arteria tibialis posterior
22	arteria fibularis (a. Peronealis)
23	arteria fibularis (a. Peronealis)
24	truncus brachiocephalicus
25	arteria carotis communis dextra
26	arteria carotis communis sinistra
27	arteria vertebralis dextra
28	arteria vertebralis sinistra
29	arteria subclavia dextra
30	arteria subclavia sinistra
31	arteria axillaris dextra
32	arteria axillaris sinistra
33	arteria brachialis dextra
34	arteria brachialis sinistra height

11.2. MATLAB code

The implemented algorithm in MATLAB consists of several programs that are briefly described in this section for the user's guidance. This explanation presents the expected inputs and outputs and needed user's interaction. For the elaboration of the method implementation see 5 or the MATLAB code attached on a CD. On the CD are also files provided to test the program. The DICOM file can be requested and is not included because of the confidentiality. File *processedData.mat* containes data that can be stored into a VTK file. Data with the *.vtk* are present to demonstrate visualization.

loadFile.m

- *input: filename*
  The input for the function *loadFile.m* is expected to be a *folder name* that contains DICOM images. Those DICOM images should have the ending *.dcm* and should also contain needed DICOM information. The expected names are 8-digit numbers starting with 00000000.*dcm*.
- *output: fileSet*
  The output of *loadFile.m* function is the processed volume set. For each volume set, two results are implicitly set: the parametric image defined with the function described in this thesis in 4 and a volume based on the omitted eigenvalue, see(5). Each of these cells represents a direct input for the function *toVtk.m* that stores those volumes as a VTK-file. The output cell array also contains the spacing information that is needed for the VTK-file.

volumes.m This function is a recursive function that sorts the folder containing the DICOM images. This function is used in the *loadFile.m* function and its output is the unprocesses cell-array of sorted volume. For more detailed description see the MATLAB code.

- *input: (slices,num,sorted)*
  The input of this function is the inicialization with an empty cell (*sorted*) and the position in the output cell where the sorted set is saved (*num*). At inicialization set to 1. *Slices* is a cell structure containing path to a DICOM slice, its voxel spacing and its global coordinate, e.g.: *'path/to/image.dcm'*, 1.51.54.5, 230.60.
- *output: sorted*
  The output is a cell array with sorted uniform volume sets.

toVtk.m This function creates a VTK-file that is saved onto the user specified place.

- *input: (image, spacing, origin, filename)*
  The input of this is the uniform processed volume in the form of a 3D matrix as *image*. The *spacing* is also additional information needed provided by the user as well as the origin of the volume. The *filename* is destination of where the file is stored, e.g: *'path/to/filename.vtk'* .

11.3. GUI

The program requires following free softwares: VTK, Python. The program is ran from a command line as *pythonprogram.py*. Following figure elaborates on the use of the GUI. It is a simple demonstration of the volume picking. The recommended use is to pick slices within the global volume and to select a specific point in the z-slice provided.

The picking is done with the right click mouse button and interaction is possible with the left mouse button. Picked points are highlighted for the information for the user.

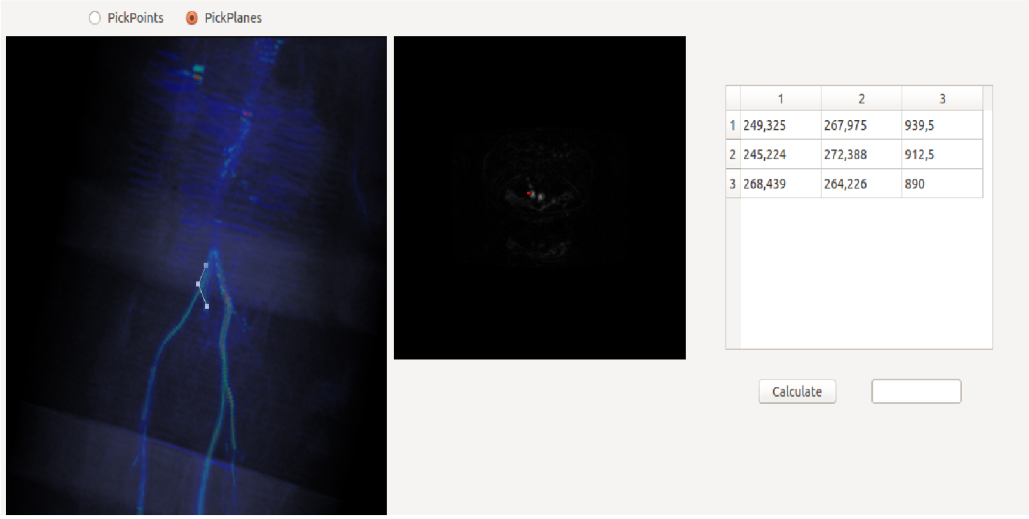


Figure 11.2: The GUI screenshot showing a picked path within the volume