University of Magdeburg

Faculty of Computer Science

Bachelor Thesis

Pipeline Preparing Artery Geometry for Simulation

Author:

Simon Sabri Schönhofen

April 2021

Advisors:

Prof. Dr.-Ing. Bernhard Preim

Department of Simulation and Graphics

Dr. Gabriel Mistelbauer

Department of Simulation and Graphics

Schönhofen, Simon Sabri:
Pipeline Preparing Artery Geometry for Simulation
Bachelor Thesis, University of Magdeburg, 2021.

Abstract

Computational Fluid Dynamics (CFD) Simulations of the coronary system is a common support for medical procedures and risk analysis. It is dependent on a well formed geometry. The creation of such a geometry is challenging and over the years multiple options for this have been proposed. For the usage with many datasets, a fully automatic pipeline is wanted.

The work described in this thesis is a implementation of a pipeline for creating a well-suited mesh for a CFD simulation. It is demonstrated how this technique can be applied to different datasets with minimal user input. Therefore a tree structure for the centerlines of the vessels is used. This allows modifications of the vessels to the needs of the CFD tools. An extrusion of the coronaries and cutting of the outlets is presented aswell, to counter reverse flow in simulations.

Kurzfassung

Strömungssimulationen von Koronarterien sind gängige Methoden zur Unterstützung von medizinischen Maßnahmen und bieten Unterstützung bei der Risikoanalyse. Diese Qualität dieser Simulationen hängen in erster Linie von der erstellten Geometrie ab. Die Erstellung solch einer Geometrie ist aufwendig und diverse Möglichkeiten hierzu wurden bereits vorgestellt.

In dieser Arbeit wird eine Implementierung einer Pipeline für die Erstellung eines simulations-bereiten Mesh vorgestellt. Es wird aufgezeigt, wie die zugrundeliegende Technik auf multiple Datensätze angewandt werden können, mit einer geringen Nutzerinteraktion. Für die Representation der Centerlines wird eine Baum Struktur verwendet. Diese erlaubt leicht nötige Veränderungen im Hinblick auf die Simulation durchzuführen. Zu diesen Veränderungen gehören die Extrusion und das Abschneiden der Enden der Adern, welche ebenfalls präsentiert sind.

Acknowledgments

I would like to thank my supervisor Gabriel Mistelbauer for the continuous support and discussions during the development of this thesis. Furthermore i learnt a lot from his experience, codebase and technical knowledge. I also would like to thank Sandy Engelhardt for her discussions and input throughout the development since they always led to further improvements.

Next, I would like to thank Prof. Dr. Görich of the University of Heidelberg for providing several CT data-sets. I also have to thank Simon Sauerzapf for extracting the provided data.
There exists a positive Ethik-votum of the University clinicum heidelberg under the abbrevations S-758/2018 and S-226/2016.

Special thanks goes to my girlfriend Rebekka, for spending plenty of time and effort to read and comment this thesis concerning the language grammar, furthermore I would thank her for supporting and motivating me throughout the development of this thesis and the situation under which this theses was written.

I would like to thank my brother for supporting me throughout my study at the university.

Finally i thank my father, who is not able see the end of my studies anymore, for his wholehearted support and believe in me during my life.

Contents

List of Figures

1. Introduction

1.1 Motivation

Coronary Heart Disease (CHD) are the leading cause of death in developed countries. An estimated 7.2 million people die each year from CHD [MN11]. CHD are defined as myocardial infarction, chronic consequences of myocardial infarction or angina pectoris.

There is a close connection between cardiovascular diseases and blood flow [MN11]. To understand the magnitude and progression of these diseases, an accurate description of the blood flow is indispensable. Currently there are no reports of fully in-vivo measurements of flow features like wall shear stress reported [DDD^{+}00]. There is currently research for ex vivo evaluations of atherosclerosis and bloodflow, but these are currently only possible with the usage of hard X-rays which have an high impact on the patients health and wellbeing [BHS^{+}19]. Therefore to get an accurate description of the blood flow, computational fluid dynamic simulations are used to simulate the flow in a given structure. These mathematical models, which represent physiological flow are only limited by computer power. With CFD it is possible to simulate diseased and healthy conditions on an individual basis.

Since CHD has a long asymptomatic latent period before causing any difficulties, it has an opportunity for early interventions. Therefore, we need to do a risk assessment to detect it early and have a higher chance of successful treatment.

Coronary Artery Disease (CAD) develops when blood vessels narrow. This is mostly caused by atherosclerosis. Atherosclerosis is the deposition of cholosterol-containing plaque inside arteries. Plaque can lead to clogging of the arteries, which limits the blood flow to and from the heart. Main cause are high blood pressure and high cholesterol.

CAD (also known as coronary heart disease) is the most common cardiovascular disease. CAD occurs, when arteries that are necessary for the blood supply to the heart become hardened and narrowed, due to the buildup of cholesterol (plaque) [CAD].

It is shown that Wall-shear-stress (WSS) promotes plaque development and high WSS promotes plaque destabilisation [SEM$^+$11].

1.1.1 Coronary Anatomy

Blood leaves the left heart ventricle and passes through the aortic valve into the ascending aorta. Upon leaving the ascending aorta, the blood deflects into three larger branching vessels which are supplying arms, the head and the abdomen.

The aorta branches off into two main coronary arteries. These are the main vessels of the coronary circulation, the left anterior descending and the right coronary arteries. These originate at the base of the aorta form openings (also called coronary ostia), which are located behind the aortic valve leaflets. Each of the vessels has multiple branches and splits up into subvessels, to guarantee blood flow to all regions of the heart muscle. The coronary arteries have different shape regarding branching, bending and tampering, and therefore are resulting in different blood flow fields [Ho09].

In a healthy vessel, walls deform due to change in pressure during the diastole and systole. Therefore, changing the bloodflow properties. The coronary walls are made out of four basic structures: endothelial cells, elastic fibers, collagen fibers, and smooth muscle cells [Zam06]. The elastic fibers are mostly consisting of elastin and provided longitude aswell as circumferential support, and are as the name suggests, responsible for the elastic properties of the vessel. Stiffening of the vessel is mostly associated with the loss of elastic fibers [SHF01]. Smooth muscle cells are muscle fibers, which can contract the wall to change the diameter of the vessel and change the blood pressure and restrict blood flow. Collagen fibers support load in circumferential direction. Arteries are thicker then veins due to the larger amount of smooth muscle cells in the wall.

1.1.2 Cardiovascular Diseases and Blood Flow

WSS, pressure and other blood flow characteristics are involved in the progression of cardiovascular diseases. Highest risk indicators are flows which are uncommon in healthy person, namely turbulent, disturbed or highly oscillating flows [CGV$^+$11]. Blood flow affects the structure of the vessel wall, which can lead to diseases such as atherosclerosis or aneurysms.

Coronary blood flow exerts a shear stress on vessel walls that alters cell physiology. Hereby arches and bifurcations of the artery are exposed to an disturbed flow, which determines a lower shear stress, whereas uniform geometry is exposed to constant flow and therefore physiologic shear stress. Atherosclerotic lesions are known to mainly deveolop in areas of low shear stress, so close to arches and bifurcations[CGV$^+$11].

1.1.2.1 Atherosclerosis

Atherosclerosis is a disease where the artery walls develop lesions due to the buildup of plaque, mostly consisting out of lipids and cholesterol and other cellular residues

Figure 1.1: *Schematic picture of the coronary vessels.[Lyn10]*

[CR15]. This buildup usually restricts the blood flow, and changes the effect of the flow on the vessel walls, regarding pressure and WSS.

This plaque can be distinguished between two different types: soft and hard plaque. Soft plaque is more likely to break apart and enter the blood flow, which can result in a blockage of bloodstream. Clinical sequelae of atherosclerosis are vessel narrowing with angina pectoris.

Atherosclerosis can be diagnosed invasively by catheter insertion into an artery to inspect a risk region, or non-invasively by medical imaging methods. Since the later approach is non-invasive, effort has been made to extend the possibilities of medical imaging using Computer Tomography Angiographie (CTA) or Magnetic Resonance Angiography (MRA) for data acquisition in addition with a contrast-agent to enhance vessels.

If atherosclerosis is detected, medication to adequately control lipoprotein levels needs to be initiated when risk reduction through lifestyle modifications such as dietary changes, stimulation of physical activity and smoking cessation is not sufficient. In secondary prevention, medical therapy is almost invariably needed in addition to lifestyle optimisation. Clinical treatment can be divided into two types, bypass which is a surgery to restore normal blood flow to the heart, by either inserting a piece of vein from the leg or by splitting the left anterior descending branch of the aorta. On the other hand is angioplasty, which is a method to enlarge narrowed arteries.

The deposit of cholesterol in coronary arteries leads to a change of the blood flow regarding speed and pressure. This leads to changes in wall shear stress and can therefore disrupt the wall. This can lead to different diseases.

Figure 1.2: Developement of plaque and atherosclerosis and the following clot [Pen16]

1.2 Data Acquisition

1.2.1 Computed Tomography Angiography

Computer Tomography (CT) imaging data is the composition of a series of X-ray images into one volume dataset. These X-ray images are acquired by a system of emitters and detectors, which are rotating around the patient, as illustrated in Figure 1.3. After a full rotation, the patient is moved in an orthogonal direction to acquire the next image slice. Meanwhile filtered back projection is based on the Radon transformation [KSW02].
Since the introduction of CT-Scanners, they evolved and the acquisition time was siginifcantly reduced. This advancement gives the option to inject a contrast agent intravenously and scan a region of interest. This is called CTA. Hereby, the contrast agent emphasizes the vessels, in order to make the vessels more distinguishable from other tissue. Further advances in multidetector CT Technology allows, to acquire a high temporal resolution permit contrast-enhanced image of coronary arteries and the corresponding plaque during a single breath hold, as stated by Hoffmann [HFCP06]. This technique has advanced to be a robust and accurate non-invasive imaging method for detecting arterial diseases and plaque, according to Straka [Str06].

Next to CTA imaging, there exist the possibility to perform the usual injection of a contrast agent into an artery with Digital Substraction Angiography (DSA). This

Figure 1.3: Left: schematic representation of the Aquisition of an CT-Image. [CMC+18] Right: Hounsfield Units for selects tissue types [KSW02].

would of course be way more invasive. The volume created by the CTA scan is very large, therefore the laborious task of issuing a radiological interpretation is very time-consuming because it is necessary to examine numerous slices in detail. Strong efforts have been made to automate or at least advance this task to a level that requires a minimum of user interaction.

The computed intensity values of a CT dataset are representing the densities of the material. These values are usually normalized in Hounsfield Units (HU). Hereby the intensity of water is mapped to 0, and air is mapped to -1000. Hounsfield values for selected tissue types are shown in Figure 1.3. As you can see in this figure, some tissue types have overlapping values. Therefore, it is hard to discriminate a specific type of tissue.

1.3 Visualisation Toolkit

The Visualisation Toolkit (VTK) is an open-source framework for image processing and 3D Graphics. VTK comes with a variety of, mostly standard, algorithms for their desired task. The big disadvantages with the usage of VTK is the non-optimized General-purpose computing on graphics processing units (GPGPU) (such as CUDA) support, which is used in the rest of the Code for parallel processing of tasks on the GPU. Without this support, the time performances of the presented algorithms are significantly worse compared to algorithms with GPGPU support.

Another disadvantage comes with the integration of the VTK into the codebase, since we rely mostly on the Visual Physician Framework, as described afterwards, it is desired to use VTK as little as possible, since the data has to be converted from the Visician Framework to VTK and back.

1.4 Goal of this Thesis

The goal of the research described in this theses is to present a novel pipeline for the creation of a „Simulation ready “ mesh for coronary arteries CFD simulations. The goal of this thesis is the development and evaluation of a pipeline to create and

prepare a mesh for a CFD simulation from the coronary arteries.
Specifically, the following aims were addressed:

- Implementation of a pipeline, with as little user-interaction as possible.
- Implementation in a way that only one tool is needed
- The mesh quality of the outcome mesh should be high

In the first section, a method for the creation of a vesseltree graph structure and volume transformation from the provided CTA-volume is presented. Afterwards the creation of a mesh out of the provided mask and the necessary processing steps for the creation of a simulation ready mesh are introduced.

Afterwards, all steps will be analyzed with focus on performance, robustness and computation effort. Each step is explained very briefly in this section and is discussed in detail in the corresponding methodolgy section. The presented pipeline is illustrated in Figure 1.4.

As input data, a CTA-dataset of the heart in a common format (.nrrd), centerlines of the coronary arteries as a pointcloud (.vtp) and the mask of the aorta according to the provided CTA-dataset are used.

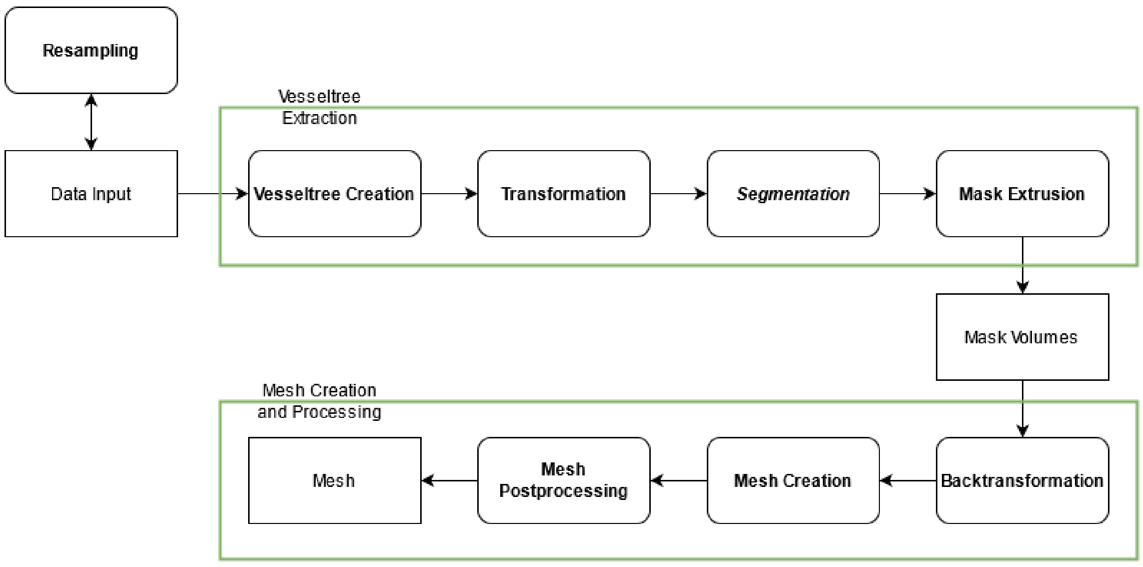

Figure 1.4: *Overview of the complete pipeline.*

This pipeline consists of the following steps:

1. Extraction of centerlines from the original volume and creation of a a vesseltree
2. Multi Planar Reconstruction (MPR) over every path of the tree
3. Segmentation and creation of 8-bit masks of the MPR

4. Inlet and outlet extrusion of the vesseltree and the MPR-volumes
5. Retransformation of the segmented 8-bit masks
6. Addition of the aortic mask to the previously created mesh.
7. Mesh post-processing

2. Related Work

2.1 Skeletonization

For visualizing vascular tree structures, like the coronary circulatory system, the vessel skeleton is often required. The skeleton of a vessel is usually a polyline, but in a structure with a number of branchings, a more complex skeleton arises and consists of the compulsion of the corresponding branches.

Wan et al [WLK+02] defines a skeleton in discrete 3D space as follows : A **discrete skeleton** is a one-voxel wide line representation which proceed in the center of a volumetric object. Based on this definition and on previous research, several properties of skeletons are described in [[WLK+02]; [KHH+04];[CSM07]]. Most of the important properties are:

1. **Connectivity** - The skeleton should be connected such it consists of directly connected voxels.
2. **Centeredness** - The skeleton should be as far away from a wall as possible, staying as close in the middle as possible
3. **Singularity** - The skeleton should have no manifolds and self-intersections
4. **Robustness** - Minor changes to the object shape should affect the skeleton only slightly.
5. **Invariant against affine transformations** - If a affine transformation, is applied to the skeleton of an object, the same skeleton should result as if the transformation would be applied before skeletanization.
6. **Smoothness** - The skeleton should not contain abrupt changes.

Another definition of a skeleton is that the skeleton is as the loci of centers of bi-tangent circles that fit entirely within the foreground region being considered.

There exists multiple skeletanization algorithms. Preim [PB13] classifies them into following categories:

- **Skeletanization by thinning** - In this method, the segmentation gets successively eroded, until a one-voxel wide line remains. To determine which points can be removed without disturbing the previous determined skeleton properties, various constraines are looked into [SBdB16].
- **Skeletanization by distance transformation** - In this method, a distance field is used to compute the skeletanization.
- **Removal of irrelevant side-branches** - Since very small side branches could not be the result of the anatomical situation, but the result of noise or partial volumen effect, filters with sensitive parameter are applied to control how much information is filtered out. Examples for such filters are given in [SPSP02].
- **Direct skeletanization** - In contrast to the other methods, direct skeletanization is used without prior segmentation of the volume. These methods are based on minimal path computation. Direct skeletanization is fast, but the results can strongly deviate from the correct centerline, especially at branchings.

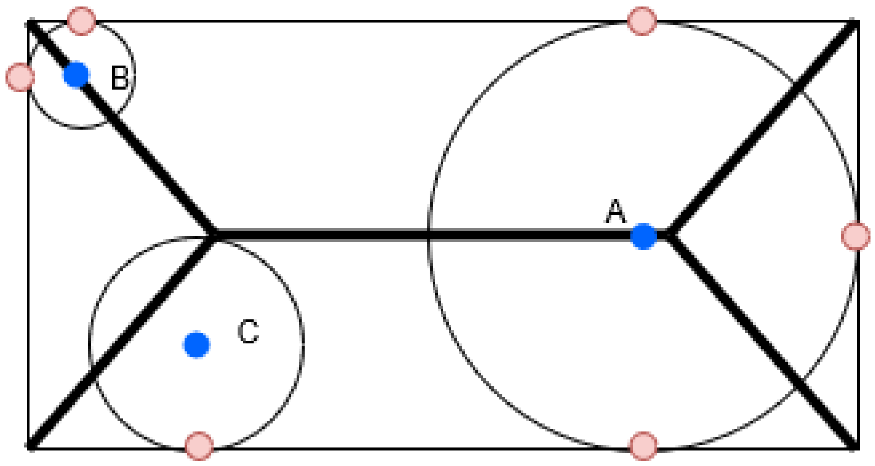

Figure 2.1: *Skeleton of a rectangle: Points **A** and **B** are part of the skeleton. **C** is not.*

2.2 Graph Conversion

The goal of a graph conversion is to transform a skeleton into a graph representation. This representation allows to visualize reformations with regards to the lumen of selected vessels. With $G = (V, E)$, V being the vertices which are representing branching points (e.g. bifurcations, or in rare cases trifurcations) or start/end points of the skeleton. The edges E are representing the connections between the vertices.

This graph can be stored with an vertex-edge incidence matrix, where each column represents the edges and each row represents a vertex.

To compute the graph, one has to calculate which voxels of the skeleton are a branching point and which are not. Since the skeleton is, as described in the previous section, only one voxel thick, one can check the following the number of neighbors of a voxel. If a voxel has only neighbor it is an end-point, more than two neighbors it is a branch point. All other voxels must have two neighbors and are called edge-points, end-points and branching-points together are called control-points (CP).
In a next step, segments between each CP are build by recursively tracing a path along edge-points until another CP is reached.

2.3 Multi-Planar Reformation

MPR gives the user the ability to define oblique slices through the volume data, to adapt a slice to the orientation of a relevant structure. This reformation can be defined interactively, with the help of a user interface, it can also be defined automatically, for example by definition of a segmentation or a relevant structure like a centerline.

MPR images can be reconstructed linearly, so the resulting images are coronal, sagital or axial representations, or curved. MPR images are in particular useful along the vessel for showing vascular detail in a cross sectional profile, facilitating characterization of stenoses or other intraluminal abnormalities [HR10]. The disadvantage is, that the definition of curved planes is often a time-consuming process that requires user interaction.

This practice is most commonly performed from a volumetric CT but is applicable for every image acquisition modality which is capable of cross-sectional-imaging (like MRI, PET and SPECT).

Since vessel are commonly not straight, joint alignment may not be readily apparent on axial selection, as seen in Figure 2.2. Therefore multi-planar, reformation will be used later in this thesis.

2.4 Computational Fluid Dynamics Simulation

The goal of CFD Simulations is to *approximate* the hemodynamics in vascular segments. This approximation process is based on Navier-Stokes equations. Those equations are partial differential equations, which are based on the laws of conversation of energy, impulse and mass. Explanations about them can be found in the work of Peter Constantin and Ciprian Foias [CF88]. The actual tools which are solving this equation system are called CFD solvers.

CFD processes contain three main components to provide useful information: preprocessing, solving mathematical equations, and post-processing.Preprocessing consists of the input of a fluid flow problem into a CFD program. This process includes the definition of the geometry, either through modelling or segmentation and

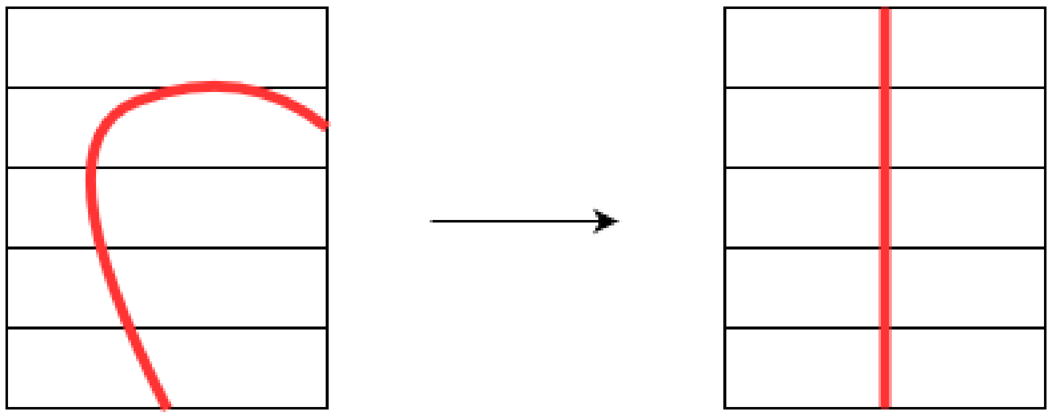

Figure 2.2: *Illustration of a MPR process*

the determination of correct boundary conditions. The solver solves mathematical equations with techniques such as finite element, finite volume and finite difference methods. Usually, a finite volume method is used for cardiovascular systems [Lee11]. The goal of the postprocessing process is the visualisation and analysis of the simulation results, so they can be understood easily. For example, changes in blood pressure and wall shear stress can be visualized by color rendering techniques.

Blood flow simulations are based on a number of parameters which influence realism and computational effort. Most of the important parameters are the following:

- **Viscosity**
- **Vessel Geometry**
- **Speed profile at the inflow**
- **Regional distribution of the speed at the inflow plane**
- **Elasticity of the vessel wall**
- **Thickness of the vessel wall**

As you can see, most of the influences parameters are connected with the mesh.

2.4.1 Geometry Definition

In [BKP+10] several properties for geometris used in CFD simulations are described. These properties are necessary to ensure convergence of the simulation.

Meshes are defined by a set of vertices and edges between those. There exist different types of edges:

- **Regular edge**. An edge is regular, if it is adjacent to exactly two faces.
- **Singular edge**. An edge is singular, if it is adjacent to more than two faces

- **Boundary edge**. An edge is a boundary edge, if it is adjacent to exactly one face

For a CFD Simulation, a mesh has to be watertight, meaning it has to have a closed boundary. From [BKP+10], [Ede03] and [Gib10] following definitions can be derived:

1. **Self-intersection**. A self intersection is an intersection of two faces of the same mesh.
2. **Star of a vertex**. The star of a vertex is the union of all its incident faces.
3. **Non-manifold vertex**. A non-manifold vertex, is a vertex where if removed, the corresponding star is not connected.
4. **Non-manifold edge**. A non-manifold edge, is an singular edge.
5. **Closed**. A closed mesh, is a mesh with no boundary edges.

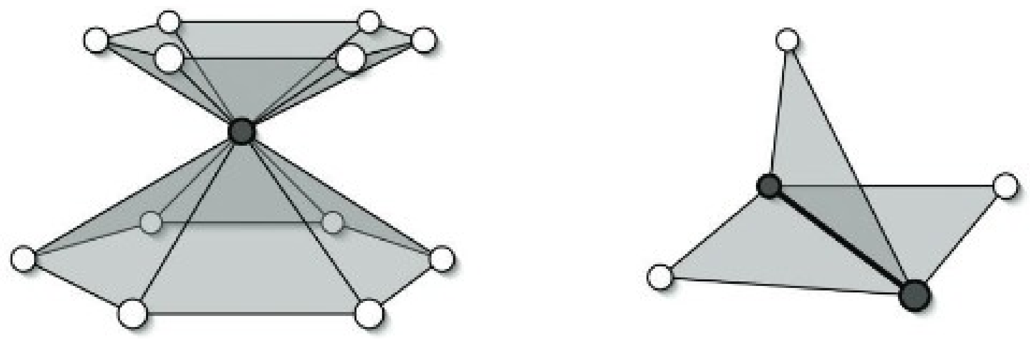

Figure 2.3: *Left: Non-manifold vertex. Right: Non-manifold edge. Image courtesy by [BKP+10]*

If a mesh has either a non-manifold edge or vertex, there is no geometry which can physically support the representation of it in the real world.

Furthermore every mesh relies on its grid. The grid designates the cells on which the flow is solved in the simulation. It has significant impact on the rate of convergence, solution accuracy and computation time required for solving [VM07]. A denser grid results in a more accurate solution, since the flow is calculated in more points, but this comes also with an significant increase of computing time. So it is important to find a sufficient grid density.
Types of grids can generally be differentiated into two types, structured and unstructured. In the former, grid points can be regarded as the points of intersection of curvilinear co-ordinate curves (in two dimensions) or surfaces (in three dimensions). The connectivity anywhere in the solution domain is thus dependent on the overall generation scheme used [FM03] . Unstructured meshes are meshes with (mostly) arbitrary structure and therefore the connectivity of elements must be stored. The

element types are non-orthogonal (such as tetrahedras). These types of Meshes provide greater flexibility for discretization of complex structures, such as a system of vessels. Furthermore they enable easier implementations of adaptive techniques where nodes can be added or deleted, while the mesh connectivity is updated locally. The drawback is, using unstructured meshes for a simulation requires larger computation effort than their structured counter parts [BA07]. But this increase may be justified through an increased solution accuracy [FM03].

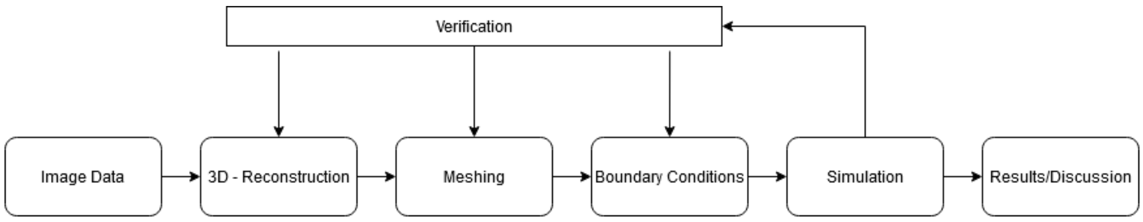

Figure 2.4: Simplified pipeline of a CFD process

2.4.2 Finite Volume Simulation

Blood flow simulations are a special instance of biomedical simulations. The goal is, to have a as phsyically correct simulation, as possible. The behaviour of tissue is represented by partial differntial equations (PDE). Finite volume methods are the standard tool for such simulations, because they are specialized on solving large system of coupled PDEs. These PDEs are representing the physicial behaviour at all cells of an appropriate volume grid. When the underlying phenomena are modeled with sufficient accuracy, and if the temporal resolution and spatial resolution is high, the simulation converges against the true solution. Since the number of such equations is often large, numerically efficient techniques are employed that take advantage of the sparse nature of the coefficient matrix (most elements are zero).

2.5 Segmentation

Segmentation is the task of decomposing image data into meaningful structures that are relevant for a specific task. It has two key aspects:

- Identification of relevant objects - meaning the objects should be recognized as particular (medical) structures.
- Border illustration - meaning they should be illustrated in a sense that their borders are precisely specified.

Moccia et al. [MDMMEH18] distinguishes between five state-of-the art vessel segmentation categories used for medical image processing. These will be briefly explained in the following:

1. **Unsupervised machine learning** - These methods are building segmentation models based on unlabeled image features, such as gradient or local intensity, working on the base of statistical distribution of the input data.

2. **Supervised machine learning** - Here the segmentation infers a rule from labeled training couple. For each pixel (or voxel) and an corresponding output value which states whether the weather the pixel (or voxel) belongs to a vessel or not, according to a ground truth segmentation.

3. **Edge-based deformable models** - Deformable models consider curves and surfaces that can move and deform under influence of internal and external forces. Here the external force is formulated according to edge-based-features like the gradient

4. **Region-based deformable models** - In the region based versions, the external force is formulated according to constraints defined by the image for and background.

5. **Tracking approaches** - Tracking algorithms usually build upon definition of seed points, followed by a growth process guided by image-derived constraints (such as the gradient). Those seedpoints can be placed manually or automatically.

Although every method has strengths, no single segmentation approach is suitable fo all anatomical regions or imaging modalities. The most suitably methods is highly depended on the task and noise of the input data.

3. Methodology

Everything is programmed in C++ with utilization of the *Visulisation Toolkit* and the *Visician Framework.*

As input data a CTA-volume, referred to as *Input Volume*, corresponding centerlines and a mask of the corresponding aorta, referred to as *Aortic Mask*, are used.

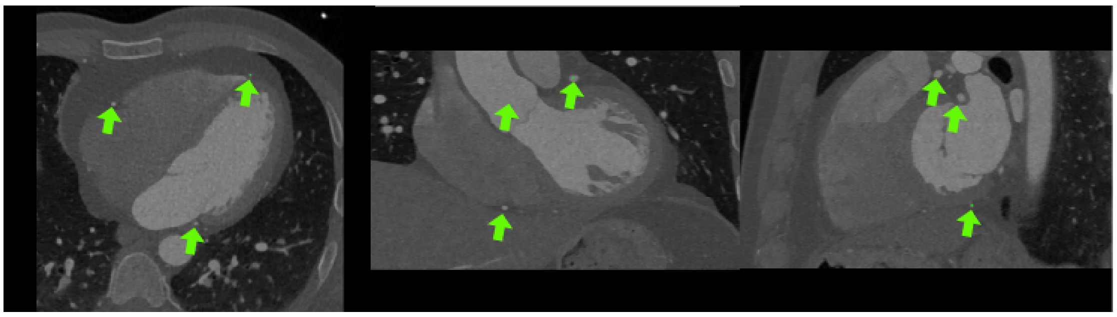

Figure 3.1: *left: axial view, middle: coronal view, right: sagital view of an input dataset. Yellow points are representing the centerlines*

3.1 Vesseltree Extraction and Preperation

In the following sections, each step for the extraction of the vesseltree and the processing will be discussed in detail. Furthermore, challenges and complications will be pointed out.

The main goal of this pipeline is the automation of every step. And the creation of a vessel tree which can be used for mesh generation in addition with the corresponding MPR transformations. The stability and robustness are also necessary and will be analyzed. At the end of each step, further possible optimizations will be discussed.

The goal of the vesseltree creation is to provide a vesseltree graph structure. For further processing the graph has to be directed, acyclic and has to have defined paths from the *root vertex* to each *leaf.*

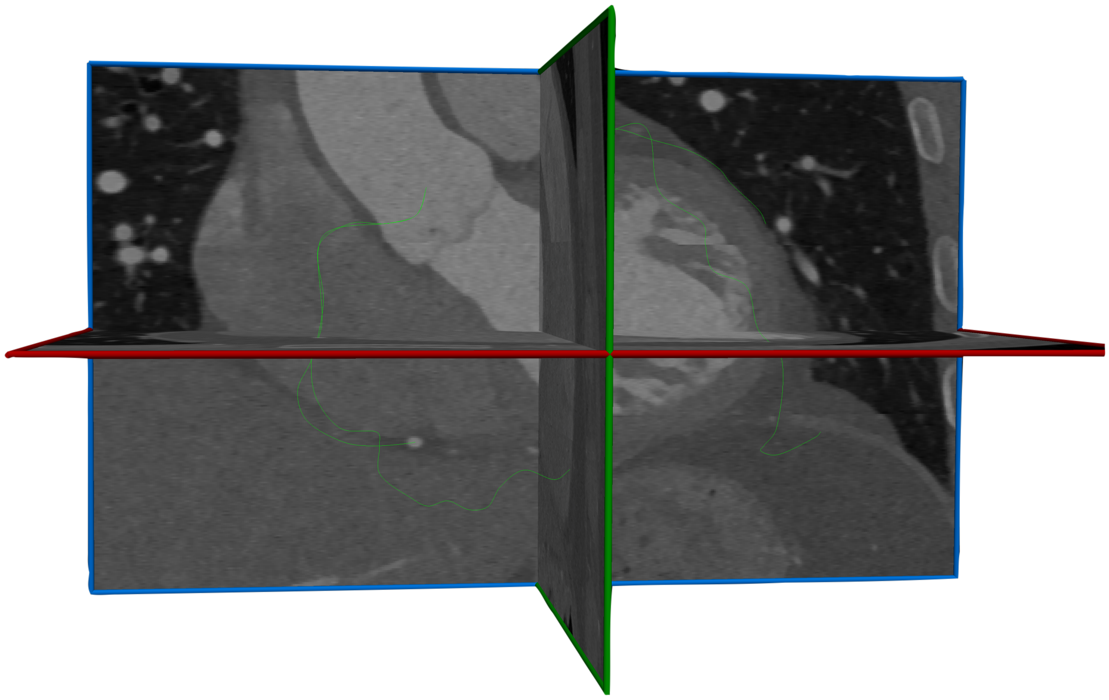
Figure 3.2: *Volume view of the input data*

An overview of the pipeline in the first part is illustrated in Figure 3.3. The first step consists of necessary preprocessing step for the centerlines, followed by voxelizing and reorientation. Afterwards the skeleton of the centerlines must be computed and a graph has to be created out of this. In a next step this graph will get oriented and paths are getting created from the root to each leaf.

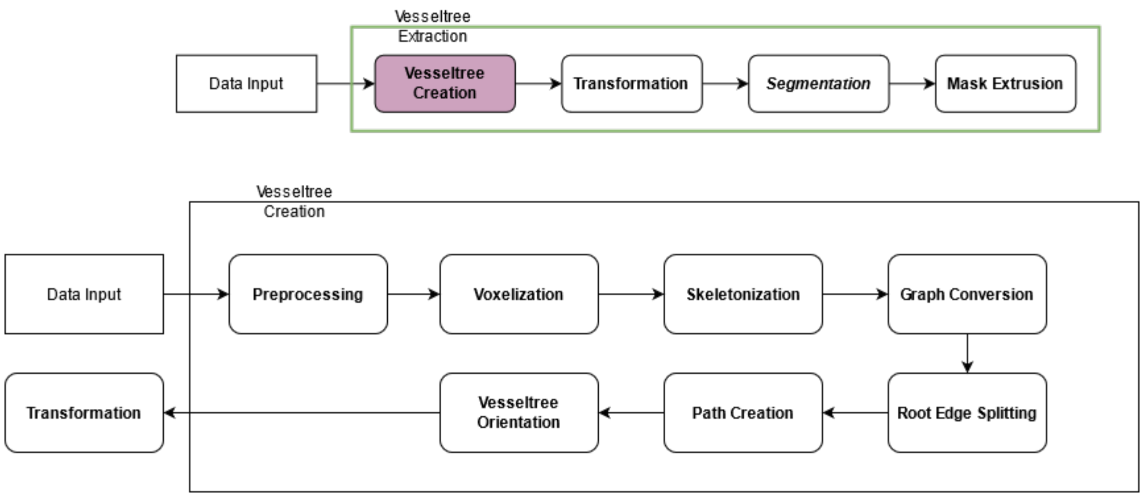

Figure 3.3: *Overview of the subtasks of the vesseltree creation process*

3.1.1 Centerline Preprocessing

The provided centerlines are representation of the vessel centerlines in the *inputVolume*. Each is originating from the aorta. A visualization of the centerlines, as in Figure 3.2, shows that some sub-branches are represented by multiple centerlines and are therefore overlapping. Since we do not have access to sublabels which label each part to the corresponding vessel, a merging of the centerlines is necessary.

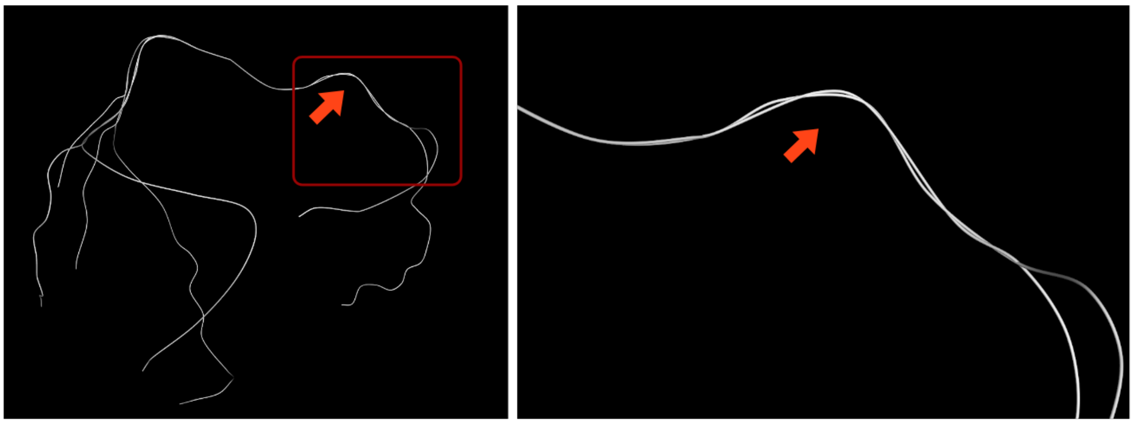

Figure 3.4: *Left: Visualisation of the unprocessed centerlines. Right: Zoom on the centerlines of the left coronary artery. The representation of the artery by multiple centerlines leads to the duplication.*

The used centerlines are provided as independent pointclouds, with no information about their origin and spacing. Since this information is needed to compute the vesseltree, it is necessary to voxelize the provided centerlines and convert all of them into one volume. Afterwards, the volume can be used for the skeletanization process. Furthermore it will be necessary to interpolate points in each centerline. Therefore each pointcloud is saved in a Bezier-spline format, so interpolation and approximation is applicable.

Before that, it has to be checked if all provided centerlines are oriented in the same way (so either from aorta to vessel end, or in reverse). If this would be not the case, the parametrization of the according curves would be reversed as well, which would be suboptimal for further work.

Because all centerlines share the same root point, we know that the root point has to be the first provided point from each centerline (or last, depending on the orientation of the centerlines). To determine if a centerline is reversed, we check if the head of the centerline and the tail of the next centerline are only ϵ apart. The same property for the tail of the centerline and the head of the following centerline in our list is checked. If either of that is the case, the centerlines gets reversed.

3.1.2 Voxelisation

These centerlines are corresponding to the centerlines of the vessels in the *inputVolume*. Since the voxelized representation of those centerlines should also correspond to those, it is necessary that the newly created *CenterlineVolume* has the same properties as the *inputVolume*. This properties are namely: **Origin**, **Size** and **Spacing**.

Since this *CenterlineVolume*, will only be used for the skeletanization an 8-bit representation is sufficient.

The voxelization works by interpolating the position of each centerline at equidistant sample points. Since these positions are given in physical coordinates, they have to be transferred to the index space by multiplication of the physical-to-index-matrix given by the *Input Volume*. Only the number of sampling points is variable. The number of sample points for each centerline is set to 1000.

Since one has to assume, that all input data is correct, the difference in the centerlines comes from small computational errors, which are non-significant and representing the same (or one neighboring) voxel in the index-space. As illustrated by Figure 3.3 the centerlines are leading to a connected voxelisation.

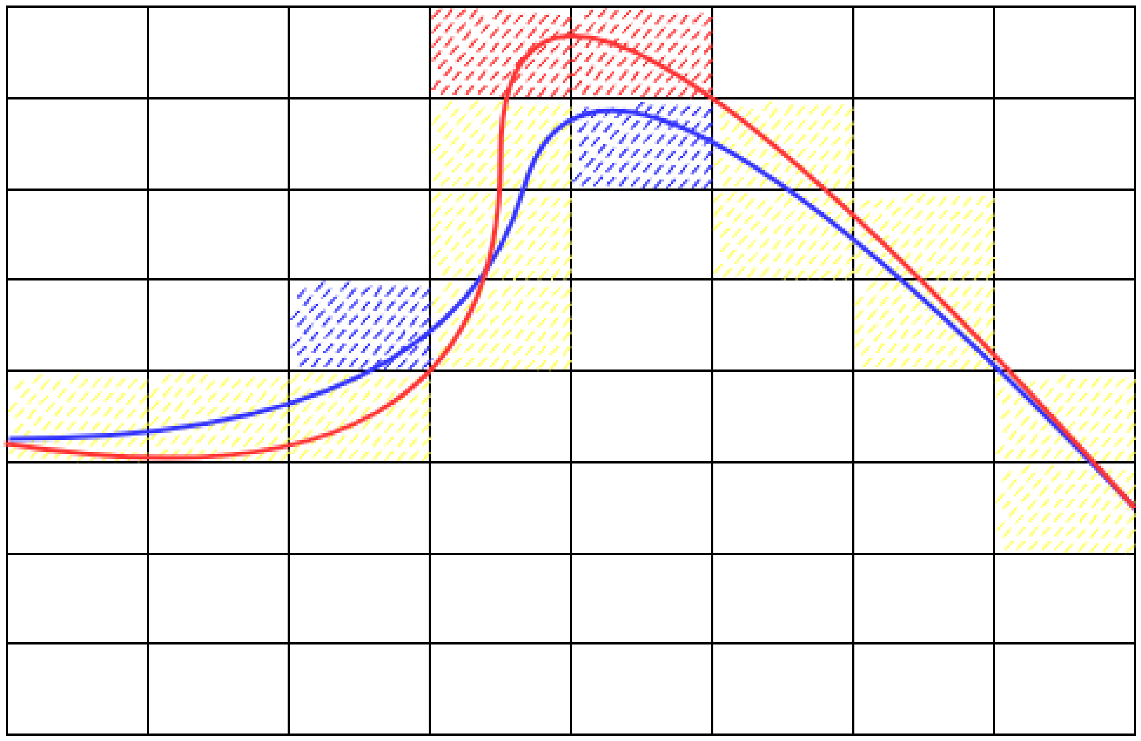

Figure 3.5: *Schematic overview of two centerlines (red and blue) of the same vessel. Yellow dots represented voxels which are represented by both centerlines, blue by the blue centerline and red by the red respectively.*

3.1.3 Dilation

As seen in Figure 3.6, the centerlines are not fully connected due to the holes. To create a vesseltree a skeletonization method will be applied, for that to work the volume must be continuous, e.g. the holes have to be closed. Therefore a morphological dilation is applied. The structuring element is spherical with radius of 2. A sphere with increased radius, could lead to better results in some cases, but can also lead to self-intersections of the centerlines. This self-intersection is obviously not anatomical correct and would lead to a misrepresentation of the aortic blood flow in

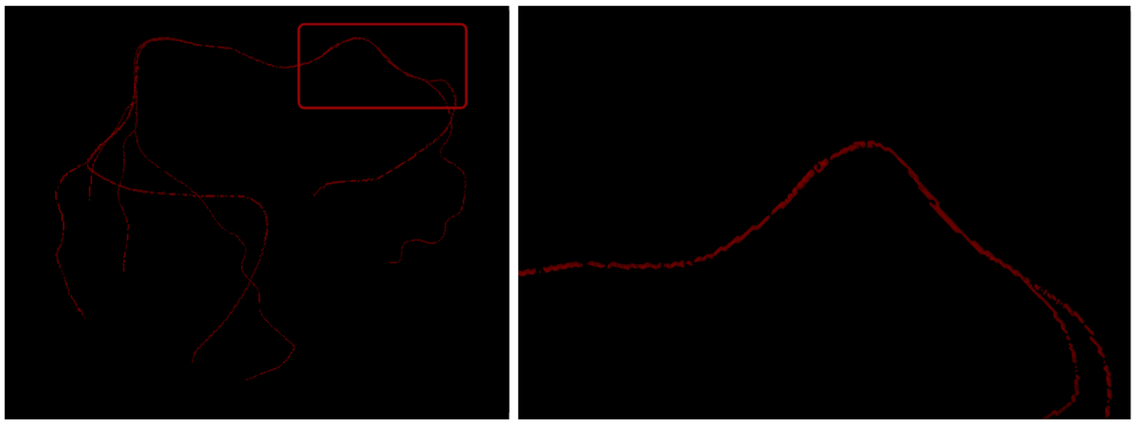

Figure 3.6: *Left: Visualisation of the voxelized centerlines. Right: Zoom on the centerlines of the left coronary artery. The centerlines are not continous since they got sampled. To fix this a dilation process can be done*

the simulation. Since the aim is a fully automatic pipeline with as little user-input as possible, this hard set value is chosen.
For further development, an implementation for automatic calculation of an appropriate radius which takes the grade of disturbance and the possible merging into account should be investigated.

3.1.4 Vesseltree Extraction

3.1.4.1 Skeletonization

The implemented method is based by on the description by Lee et al [LKC94].

This method classifies each point into borderpoint, endpoint and simplepoint. This implementation is based on parallel peeling of the border points of each voxel. A voxel can not be removed if it is not an end point, euler-invariant and not a simple point. For each voxel in 6-neighborhood of the current voxel, it is checked if the point is not an end-point. In a serial implementation, removal of the voxel would be sufficient. But in a parallel implementation, removal could lead to the disconnection of the volume. For the parallel implementation, the point is added in a list of points for sequential rechecking. This sequence of steps is repeated until no changes occur. The output is a one voxel thin centerline in 26-connectivity, which will be used in the Graph Conversion.

Since a dilation on the centerlines is applied, before the skeletanization, no noise reduction on the skeleton is necessary.

3.1.4.2 Graph Conversion

The goal hereby is to create a graph representation out of the skeleton. A graph representation is needed for assigning each part of the skeleton to the correct coronary vessel. This assignment is later on used in the Transformation, where the graph allows a reformation in regards to the lumen of each vessel. Furthermore a graph representation is useful for visual exploration.
The underlying algorithm was first introduced by Mistelbauer [Mis10] in 2010. This

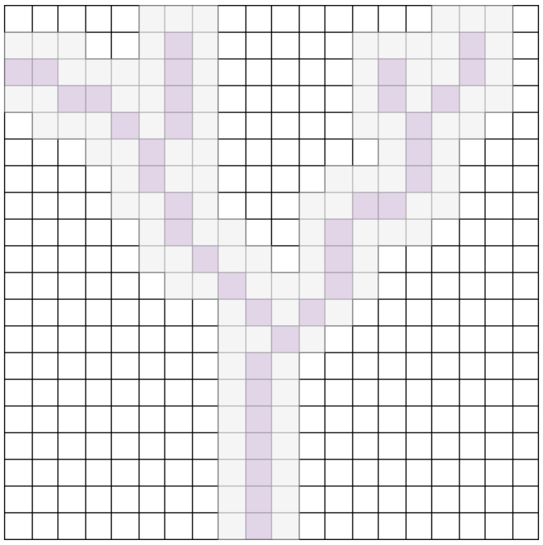

Figure 3.7: *Illustration of a skeletonization process of one vesselbranch. Darker pixels represent the discrete skeleton*

is implemented in a three-step process. In a first step the number of neighbours of each foreground voxel is determined. The number of neighbors are classifying the current voxel: A voxel with one neighbor is a end-point, a voxel with three neighbors is a branching point and all other voxels are edge-points. Edge points must have exactly two neighbors. Branchpoints and edgepoints together are called ControlPoints (CP). This classification works, since the skeleton is only one voxel thick.

In the second step, segments between the control points are build by recursively tracing a path from each control point, over edgepoints until another controlpoint is reached. Tracing from each part works under four underlying conditions for the neighbor determination. This algorithm terminates if another checkpoint is reached.

1. Background and traced voxels are excluded.
2. Only consider neighbors which are not the previous point.
3. Only consider neighbors with component-wise distances larger than 1.
4. If a controlpoint is a neighbour it is taken.

Figure 3.8 illustrates this process.

Each path between two CPs is representing an edge. Now in a last step, each edge is combined with the corresponding centerline-section. Every CP is stored with an index, all edge-points of each segment are stored too, but independently, such that one is able to define different edge points without changing the connection between two CPs.

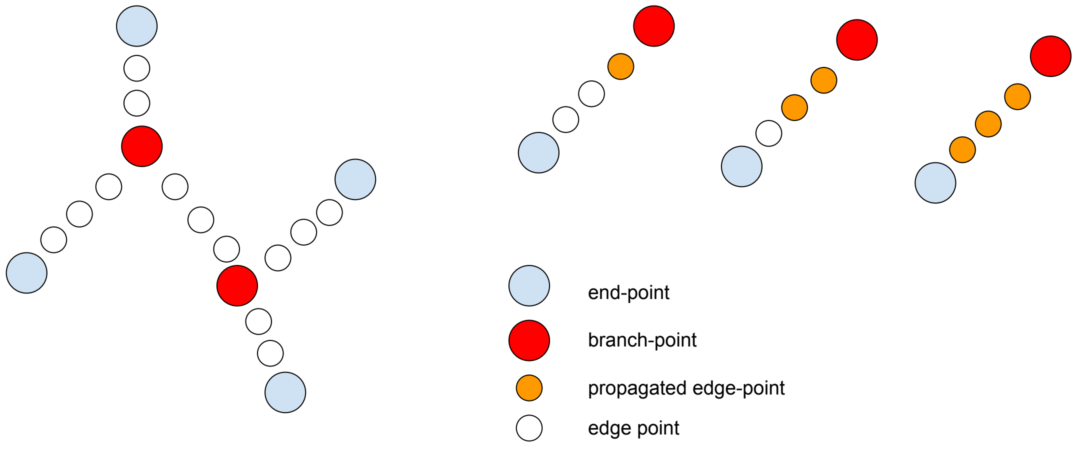

Figure 3.8: *Illustration of the propagation to determine each segment. Visited edge-points get marked. Each CP is used only once for propagation.*

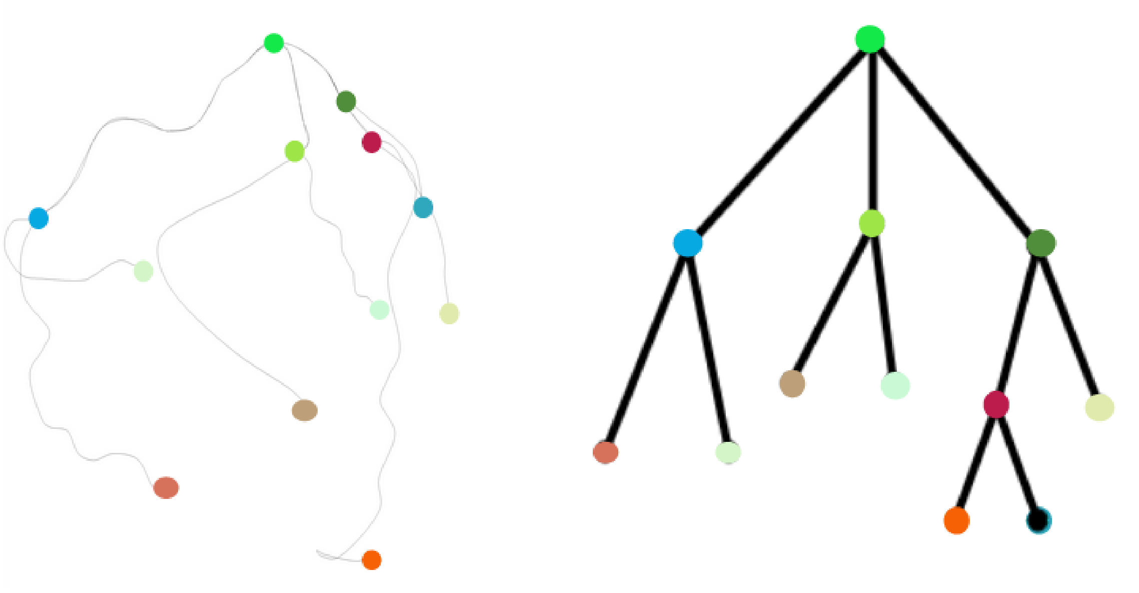

Figure 3.9: *left: Coronaries, right: Vesseltree as graph representation. The colors of each vertex represent each other.*

This leads to the creation of a graph, the *vesseltree*, which corresponds to the structure and bifurcations of the underlying arteries in the volume.

3.1.4.3 Root Edge Splitting

Since the blood flow splits from the aorta into the two descending arteries, the vesseltree should represent this by having the root vertex in the middle of the aorta and two descending edges from it.

If one takes a closer look at the collapsed centerlines, one notices that there exist none distinctive root point, rather there exist an root edge which goes through the aorta Figure 3.9. This also leads to the trifurcation from the root vertex in the graph. For a correct representation of a tree, there should be one vertex in the middle of the aorta, where two edges are descending from. But, due to the fact that the *root vertex* is a shared point of both edges, it has two neighbors and was therefore classified as an edge-point instead of an end-point during the Graph Conversion. This leads to the merge of the two edges which are representing the right and left coronary artery.

To have the vesseltree correctly represent the coronary arteries, the location on the root vertex has to be determined and the corresponding edge has to be split in two parts.

The coordinates of the rootpoint are known from the Centerlines processing.

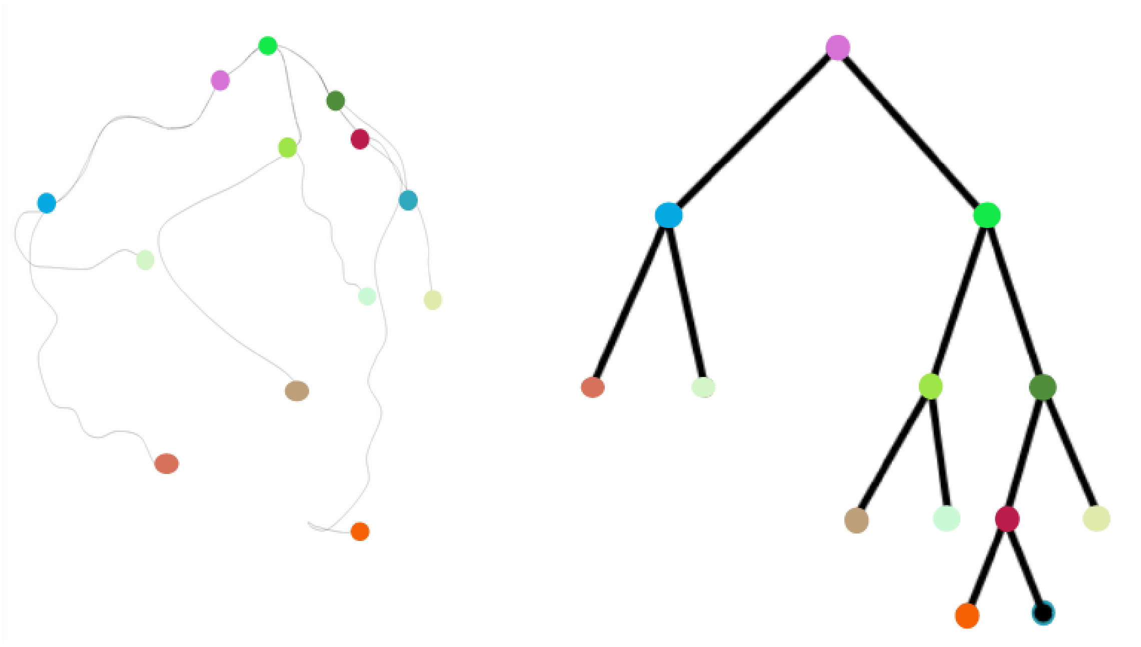
Figure 3.10: *Resulting vesseltree after the splitting. The pink dot represents the newly inserted root. Left: Coronaries, Right: corresponding vesseltree*

The edge which the root vertex part of, is the edge with the minimal distance to the rootpoint. To determine the edge which the rootpoint is part of, the distance from the *root point* to each edge is calculated and compared. Furthermore the timestep t of each centerline has to be determined as well.

In a last step, a new root vertex is created. Two new edges are created from *root vertex* to *e1* and from *root vertex* to *e2*. And finally the *root edge* gets removed from the vesseltree.

3.1.4.4 Vesseltree orientation

In the current state the vesseltree is an undirected graph. Meaning it is not defined if a vertex is source or end-point of an edge. In the next step paths are being created, for this the direction of each edge has to be given.

Each end point, is a destination vertex, where the edge comes to an end. The *root vertex* does only have outgoing edges, and is therefore a source point. All branch-points are source and destination points, since they have incoming edges (starting from another branch-point or the root vertex) and outgoing edges (going to another branch-point or end-point).
For computing the directions, a simple Depth-first-search is implemented, starting from the root vertex. As a reminder, the vessel tree is implemented as a graph structure with a unique root vertex. Mind, that by design the graph is strongly connected and a representation of the coronary system. Therefore, no self-intersecting loops can be present.

3.1.4.5 Path Creation

For applying a MPR, defined paths from the *root vertex* to each end-vertex are necessary. A path hereby is the order of edges one has iterate over to reach the end vertex from the root vertex. To compute the paths, one has to simply follow the root vertex to each end-point.

After the paths were computed, centerlines of the paths are created by concatenation of the centerlines of each edge, and stored with each path.

3.1.5 Multi-Planar Reformation

As stated in Related Work, multi-planar reformation of the volume over each path leads to the creation of volume which are transformed in such a way, that the vessel is centered. This reformation is useful for two key reasons. First of all, segmentation on MPR slices is easier, due to the good distinguishability of a vessel and non-vessel, furthermore segmentation can be made easier by zooming in on the vessel. Although this is not necessarily the case with a well trained deep-learning model or with advanced techniques like described by [MDMMEH18]. Second the necessary extrusion of the vessel is easier in the MPR-volume, since the orientation and structure of the extrusion is given. This will be closely explained in the corresponding section.

For the reformation an iterator volume, so a volume where the transformation is written to, has to be defined. Obviously this volume has to have the same datatype as the *Input Volume*. Since the *Input Volume* is resampled around the centerline and vessels usually have an elliptic shape, the volume size in x and y direction can be the same and adapted to the wanted resolution. The size in z-direction, so the number of slices needs to be chosen carefully. A to low amount leads to undersampling and therefore to a not correct backtransformation into the original volume, since too much information is missing. While a too high amount would not make any problems, it would take a toll on the memory an device space needed to save the volume. The size is choosen according to 2.5 times the length of the underlying centerline but with a minimum slice amount of 400. The spacing of iterator volume in x and y direction is depended on the zoomfactor used in the transformation and conclude to $1/zoomfactor$. The spacing in z-direction is depended on the number of slices and the length of the centerline. It accords to the length of the centerline divided by the number of slices. The origin of the iterator volume is set to the same as the origin of the *input volume*. The results of such a reformation are visualized in Figure 3.10.

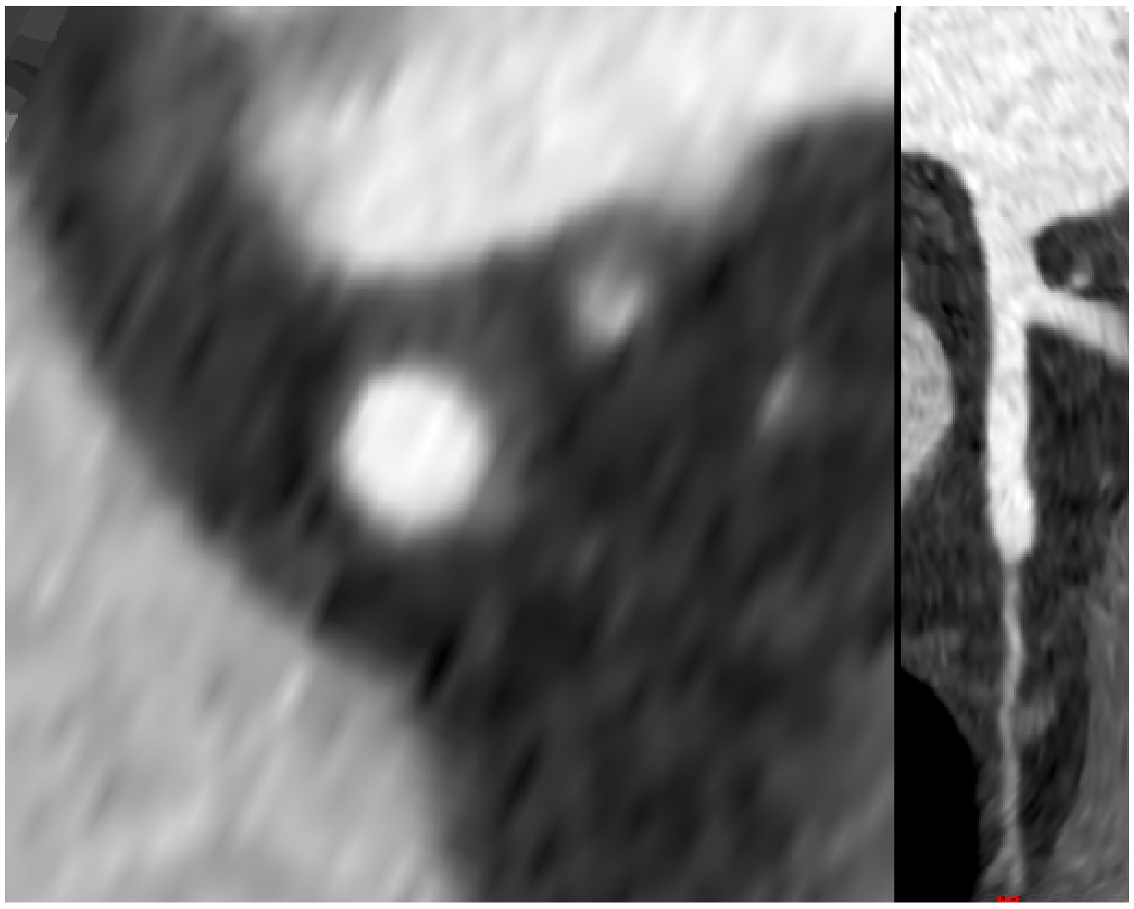

Figure 3.11: Left: Corresponding vessel highlighted in CTA Image and in the vesseltree; Right: Corresponding MPR Image

3.1.6 Segmentation

For the mesh generation, a *Vesselmask* for each of the previously created *MPRvolume* have to be created. These maska should have a value of 0 for the background and a constant value for the coronary arteries. They should have the same size, spacing and origin, as their corresponding MPRvolume. To create such a mask different segmentation methods are available and in Section 2.5 state of the art methods were presented and can be used. If it the masks are unprecise, the resulting mesh and therefore the simulation are unprecise aswell.

3.1.7 Extrusion

For the CFD-simulation the inlets and outlets require a minimum length of the adjacent vessel. The inflow and outflow areas may not be directly adjacent to bent vessel parts to achieve more stability during the simulation process and hinder the creation of reverse flow. The minimium length should be about seven to ten times the diameter of the vessel. In the backtransformation, each mask will be backtransformed into its original place according to the corresponding centerline information. To extrude each of the vessels, each centerline has to be extruded aswell. Since the volume is recreated out of the segemented masks via the centerlines. The masks should be extruded aswell to account for the extruded length of the vessel. Also the

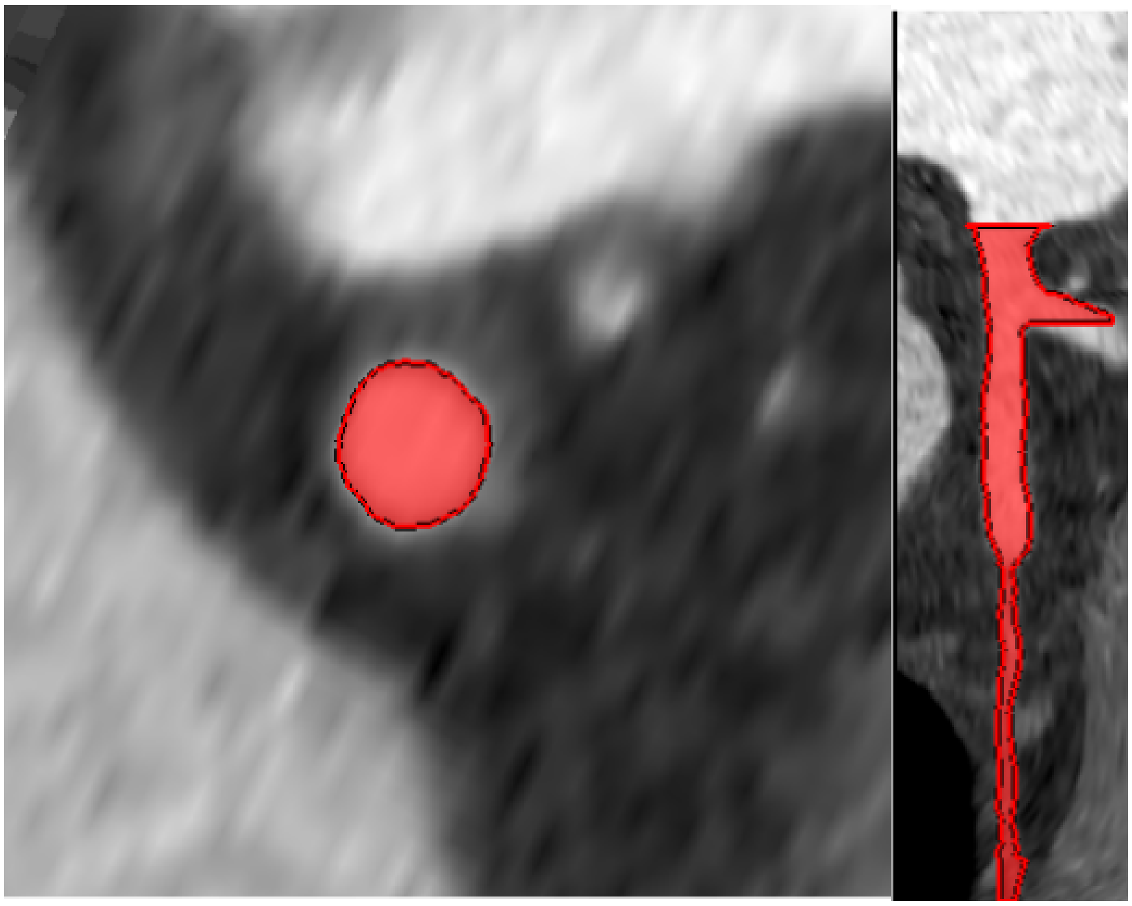

Figure 3.12: *Left: Axial view of the segmentation. Red is the segmented mask. Right: Coronal view of the segmentation*

inflow and outflow of the aorta has to be tackled. In the following subsections, each necessary extrusion will be explained in far greater detail. In general, to extrude a centerline or a mask three attributes must be known: The direction of the extrusion, the position of the extrusion and the shape of the extrusion.

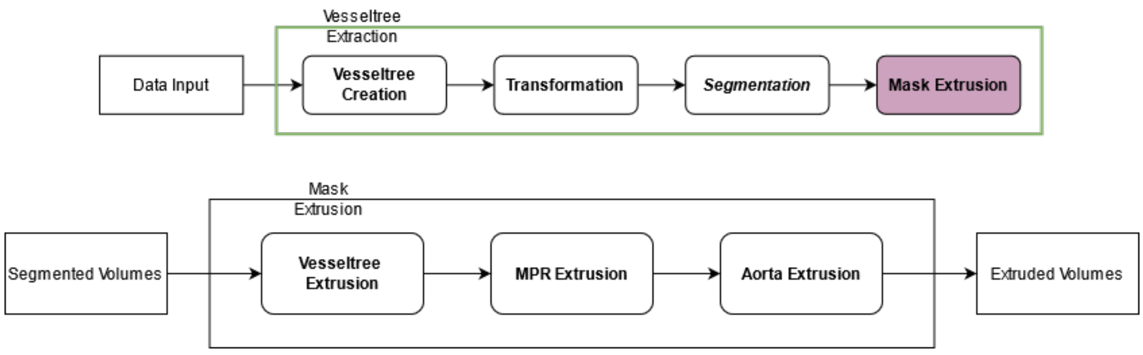

Figure 3.13: Overview of the steps for all extrusions

3.1.7.1 Vesseltree Extrusion

Each vessel, so each path in the vesseltree, has to be extruded. Since all paths consists of a ordered list of edges, this is implemented by creating an extra edge and add it to the path at the end. The creation of an additional edge in each part, as opposed to the extrusion of the last edge is useful for the labeling process, which is explained in detail in the corresponding part of the pipeline.

To create a new edge at least four control points have to be defined, since the underlying edge structure is based on a cubic bezier curve. The new edge has to have a watertight connection with the vesseltree, therefore the first point used should be the last point of the current path. One can derive the direction of the extrusion by the direction of the vector between the last two points of the edge. Furthermore, since all underlying CP's in an edge are equidistant sampled with 1mm distance to each other, this vector also has the length of 1mm. Since the extrusion needs to be 8mm long, the endvertex of the extruded edge, can be computed by addition of 8 times the vector to the edge end. The points between can be computed by creating equidistance points between the the edge end and the just computed endpoint.

For the further extrusion of the corresponding MPR, the *ratio* between the length of the new edge and the length of the whole path has to be saved.

One challenge hereby is, that the extrusion could be to large, so that the extrusion could lead to an intersection between the extruded edge and another edge. Although this was not the case in any of the datasets that were used, it is possible. To solve this, one edge has to be either shortened or change the direction. Keep in mind, that edge shortening could lead to back coupling in the simulation which can lead to non-convergence. On the other hand, edge redirection leads to a different flow pattern at the artificial curve created. And it is also of a high computational effort to compute a fitting edge direction.

3.1.7.2 MPR Extrusion

The MPRmasks created during the segmentation, do not reflect the extrusion of the centerline yet. Since we did not write the image information for the extruded edges. To keep the blood flow at the end of the original vessel unaltered, there should be as minimal disruption as possible on the transition of the vessel segment and the extruded vessel segment. Since the direction of the extruded edge does not change, the only way that a change in bloodflow could occur is a change of form and radius. To prevent this the last slice of the each MPRmask can be copied and appended to the of the MPRmask. This ensures that bloodflow will not be disrupted. This appending has to be repeated *ratio* times, to reflect to correct extrusion length.

3.1.7.3 Aorta Extrusion

The extrusion of the aorta is different for the inlet and outlet. If one takes a closer look at the aortic mask, one can see that the aorta outlet is touching the z-axis and is orientated vertically.

Therefore to extrude the outlet, a similiar technique is implemented, than in the MPR extrusion. The last slice on z = 0 is duplicated and appended to the mask. This ensures that the direction of the aorta does not change. This process has

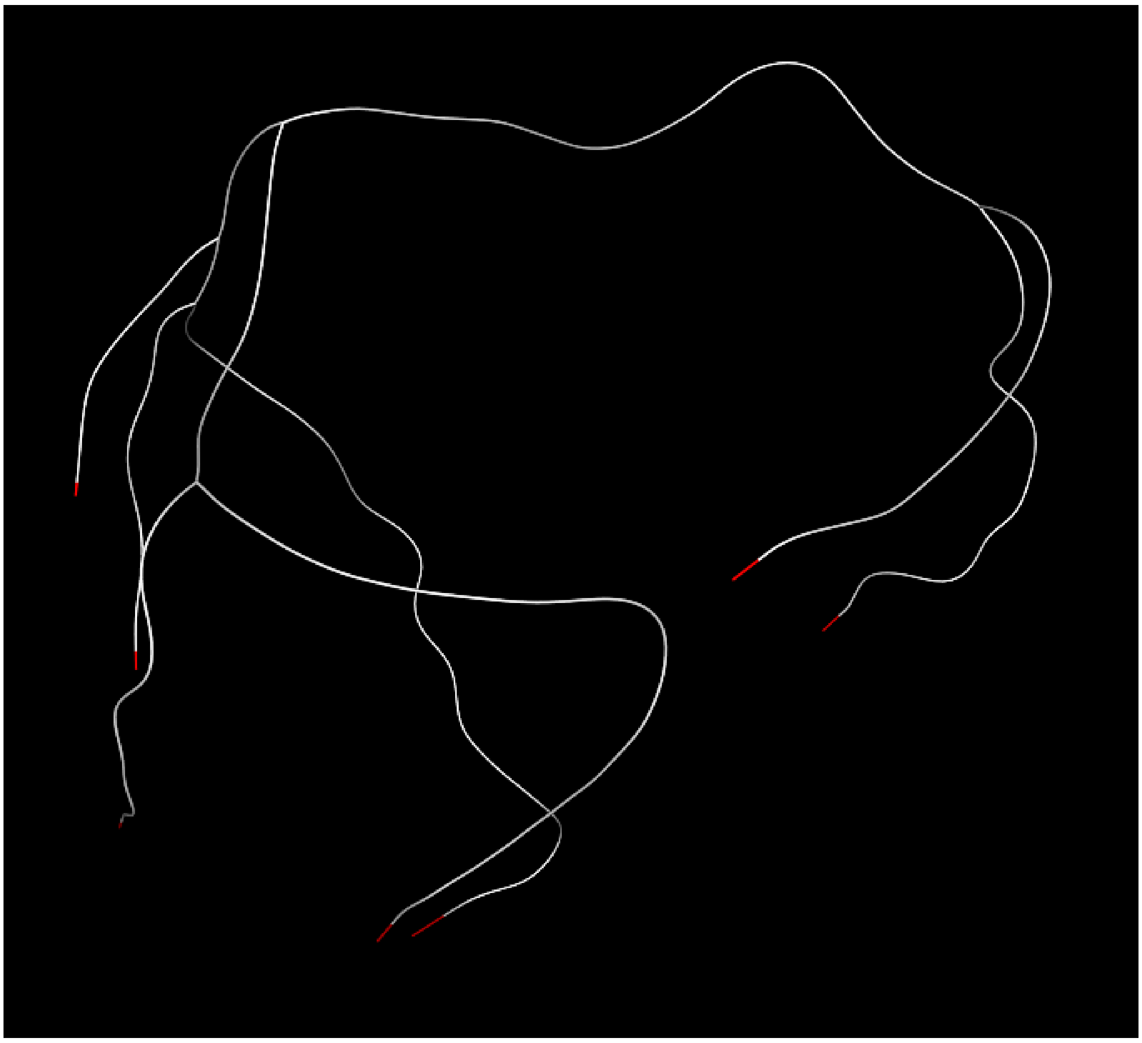

Figure 3.14: Visualisation of the extruded vesseltree. Added edges are colored in red.

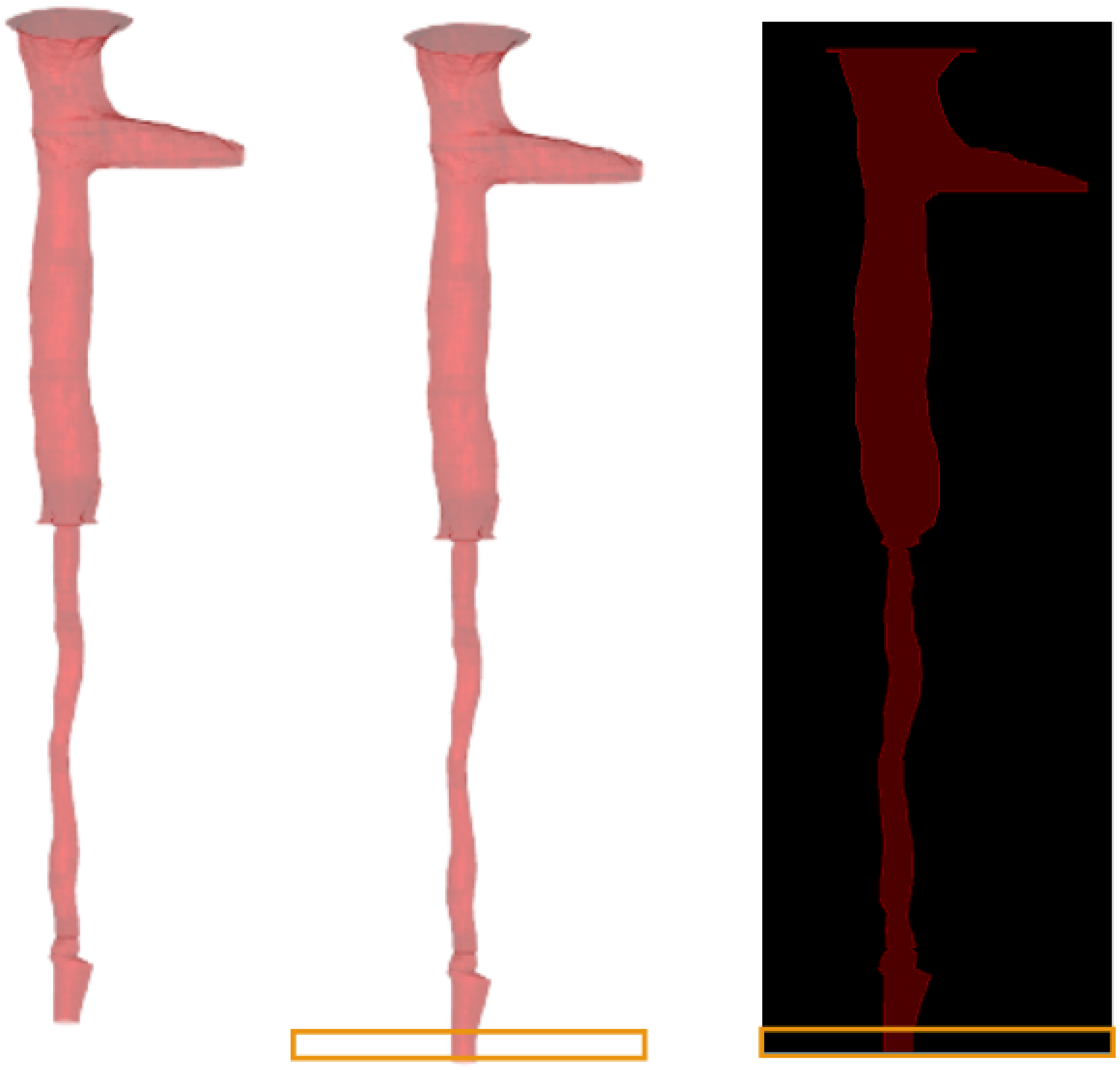

Figure 3.15: Left: Mesh of the original segmentation. Middle: Mesh of the extruded segmentation. Right: Extruded segmentation. The box shows highlights the extruded part.

to be repeated 8 times *AortaThickness* - times, to ensure stable flow during the simulation.

Since the aortic mask will be expanded, there is need for a new volume with bigger size in z-direction. It is also necessary to update the origin of the volume, since the original origin would not fit with the extrusion. The size of the new volume *AortaInflowExtrusion* stays the same in x and y direction, and is the size of the extrusion + 1 bigger in z-direction to reflect the Aorta extrusion. This +1 comes from the fact, that a marching cubes algorithm is used for the mesh generation and without an empty slice, the mesh would not be closed.
The change in size is followed by the change of origin with the addition of the size of the extrusion to the z-coordinate. One has to keep in mind that the origin is saved in physical coordinate space and the size is in index coordinate space, therefore the change of origin has to be done in regard with the spacing.

The extrusion of the inflow is more complicated, since neither position or direction of the extrusion can be easily computed.
This challenge was tackled in a similar approach as for the extrusion of the *MPR Masks*.
First the skeleton of the aorta was computed. Since the aorta does not have any branches, a simple skeletanization process is enough and no prunning is needed. The skeleton consists of one edge and two CPs. Afterwards a MPR volume of the aorta is created. Mind, that the aortic volume is only a mask volume and therefore only consists of foreground (aorta) and background. Since a segmentation of the aorta is not needed, the zoomfactor does not matter. Only the spacing and size of the iterator volume must be adapted the same way as for the cornonary MPR.
The centerline computed by the skeletanization and the MPR are extruded, in the same way, as described described for the MPRmasks and the vessel tree. Nevertheless the extrusion length can not be decided the same way as before. Anatomically, the aorta does not have a linear extrusion at this point, this extrusion can lead to collision with other vessels.

In further research one could create a segmentation of the ostia, together with the aorta to represent the anatomy correctly and stop the merge of the vessels.

3.2 Mesh Creation and Preperation

As mentioned, the main goal is to provide an simulation-ready mesh for a wall shear stress analysis. In the next section, the steps for creating a Mesh out of the *Extruded MPRs* are explained in detail. Afterwards the postprocessing needed, to make the mesh "simulation ready", is explained.

3.2.1 Multi Planar Reformation - Backtransformation

To get a combined volume out of all segmented Masks, the reversal of the MPR has to be executed. Since it is wanted to transform it into a volume with the same parameters as before the transformation, a new volume *CoronaryMaskVolume* with the same size, extent and spacing as the *InputVolume* is used as iterator volume. This process is similar to the MPR process, except instead of reading the subimage and writing the information to the MPRMask, the information in each of the MPRMask is written to the corresponding position of the *MaskVolume*.

This results in one mask for all coronaries, *coronary mask*. Furthermore the previously created MPR representation of the aorta has to be transformed back as well, *aortic mask*.

For future work a parallel implementation of the backtransformation should be investigated. A challenge with this, is that a destination voxel could be written to at the same time. But since the masks are in a binary format, a implementation of a mutex should be kept in mind.

3.2.2 Combining of volumes

In the current step of the pipeline there exist two volume files, the *AorticVolume* and the *CoronaryMaskVolume*. Since the endgoal is to get one mesh for the simulation, the volumes have to be merged at some point. In theory there exist two ways: Either the volumes are merged before the mesh generation, and one mesh is getting generated from one volume. Or a mesh is created for each volume, and the meshes are merged afterwards. One problem with latter approach is, that the connection between the aorta and the two descending arteries is closed. Therefore no blood flow can be simulated. This can be fixed by rebuilding the structure and collision detection, but that would be considerably harder and would lead to a higher computational effort needed.

During the aorta extrusion, a new volume was created, which has a different size and origin as the *CoronaryMaskVolume*. Since the *AorticVolume* has a greater extent is z-direction. The voxel information of the *CoronaryVolume* has to be copied into to the *AorticVolume*.

Since every voxel of the volume is only read once, this task is suited for an parallel implementation on the GPU.
For further aspects one could look into the usage of streaming for the combination. Both volumes have to be loaded into the system memory. Since on masks with a higher resolution, the extent of the volumes is far bigger. In this case, it could be not feasible to run this in a performant way.

3.2.3 Mesh Creation

To make sure that the mesh is watertight, it is necessary that the *merged Mask* does not have any holes. This holes could be created by unprecise segmentation or on the connections between the aorta and the coronaries. To close these holes a closing filter after implementation of VTK is used [Leh05]. The morphological closing of an image is defined as a dilation followed by a erosion of the image. Afterwards a morphological opening is performed to remove border voxels which have been created by the closing. In the final step, the mesh is build by the usage of a marching-cubes algorithm, implemented in VTK.

Although marching cubes is a very popular and wide used algorithm, it can lead to some problems. The marching cubes algorithm neither guarantees correctness nor topological consistency [NY06]. Frühauf noted that marching cubes is unlikely to yield a correct result for CFD data. Since CFD data is generated by a non-linear interpolation function instead in contrast to the linear interpolation used by the marching cubes algorithm.

Furthermore a problem with marching cubes, is the merging of close vessels. Since marching cubes does not detect boundaries and only checks for neighboring voxels. An example of this can be seen in figure Figure 3.17.

Their are multiple options to solve this problem. A modification of the marching cubes algorithm which allows to specify boundaries over which the algorithm is not allowed to work over. An other option would be to use an alternative to marching cubes.
Different meshing algorithms should be investigated regardless.

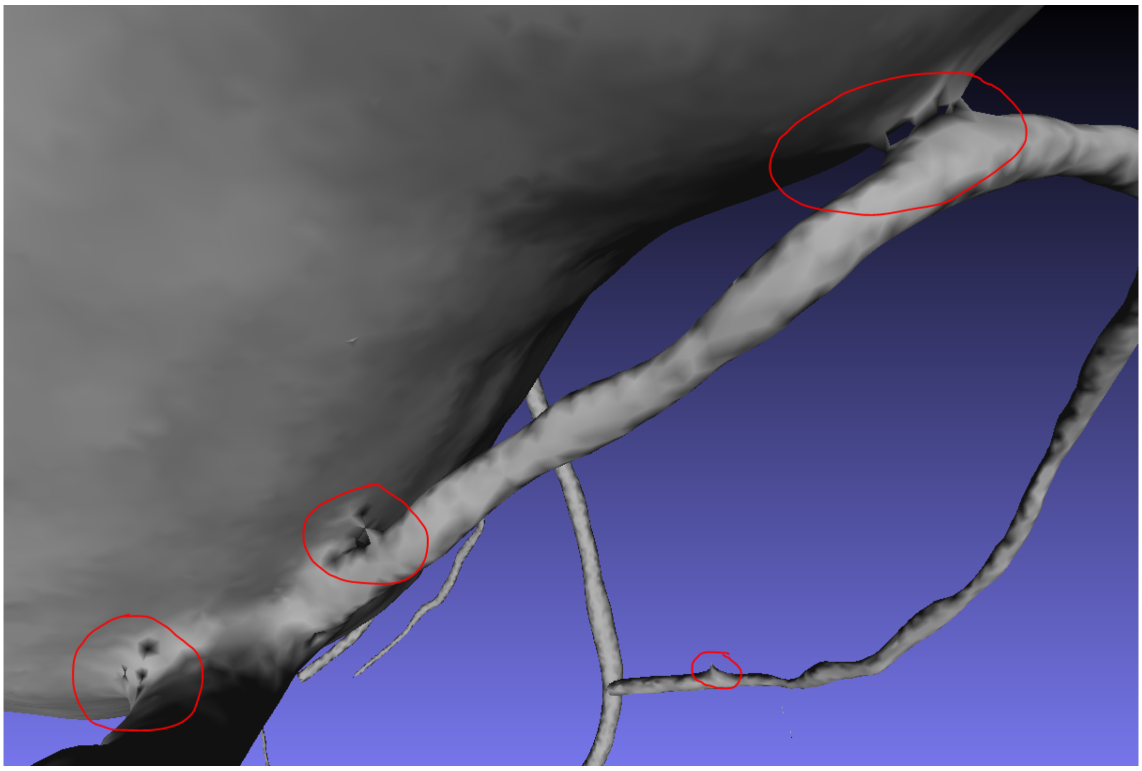

Figure 3.16: *Merging of an vessel with the aorta during the marching cubes process*

3.2.4 Mesh Postprocessing

3.2.4.1 Smoothing

Marching-cubes based algorithm create coarse meshes, as seen in Figure 3.22. For anatomical correctness in the CFD siimulation, the mesh has to be smooth. Smoothing is done by adjusting vertex positions using a windowed-sinc interpolation kernel. The implementation is based on the work of [TZG96]. Since normal smoothing leads to a shrinkage of the volume. This algorithm consists of four steps. First, for each

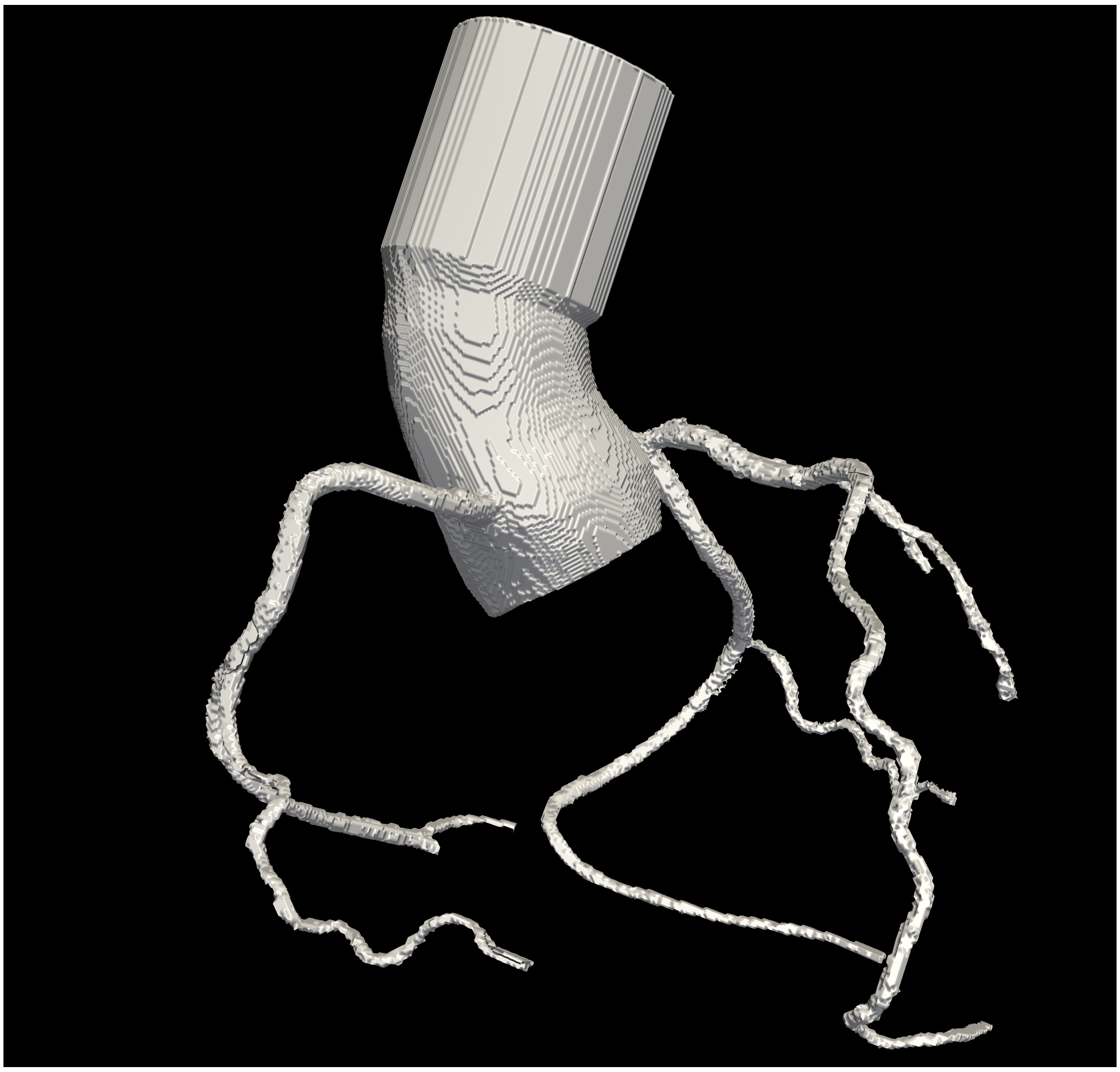
Figure 3.17: *Visualisation of the coarse mesh*

vertex the connected vertices and cells are computed. Afterwards a connectivity array for each vertex is computed. This array consists of a list of vertices which are connected to the vertex.

Since a Chebyshev approximation with too few terms is a relatively poor approximation. The first few smoothing iterations represent a severe scaling and translation of the data. Subsequent iterations cause the smoothed polyhedron to converge to the true location and scale of the object.

The total smoothing is controlled by using two parameters. The number of iterations determines the maximum number of smoothing passes. These corresponds to the degree of the polynomial that is used to approximate the windowed sinc function. The second parameter is the specification of the PassBand for the windowed sinc filter. Where lower PassBand values produce more smoothing. A PassBand value of 0.001 is used wiht 15 iterations.

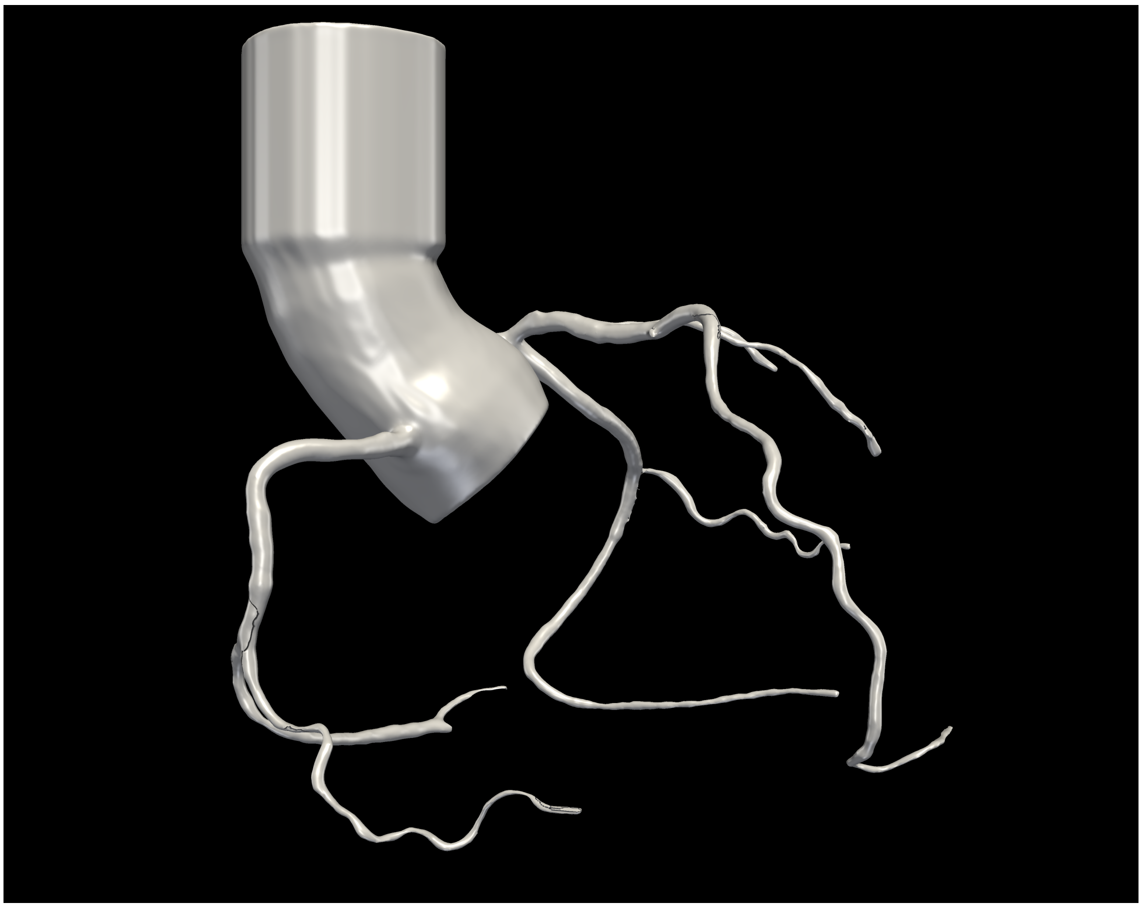

Figure 3.18: *Smoothed Mesh*

3.2.4.2 Outlet Cutting

For a CFD simulation, inlets and outlets have to be specified to define inflow and outflow behavior. The inlet in this case is the aorta (coming from the bottom) and the outlets are the endpoints of the vessels. But the defined inlet and outlet have to be flat, manifold surfaces, which have distinctive edges between other surfaces. The smoothing process removes those sharp surfaces as seen in Figure 3.22. To recreate these surfaces, a simple boolean subtraction can be done between each vessel end and a disc shaped mesh. The position and orientation of this mesh can be directly determined by the vesseltree/the corresponding centerline. Where the last points of each path, gives the position. The orientation has to be perpendicular to the path. The only missing information is the radius of the disk. If the radius is set too small, the subtraction will not flatten the edges. Since the pathological structure of the coronary arteries is well known, it is known that the end of the coronary arteries are having a radius of [DJBBD92]. Our radius is, for robustness reasons, 2 times larger.

This process is applied on every end of a path, and leads to a simulation-ready mesh.

3.2.4.3 Mesh Labeling

As of now the mesh can be used for a satisfying CFD simulation. If the simulation of the mesh is finished, we get the resulting information in every cell/face of the mesh.

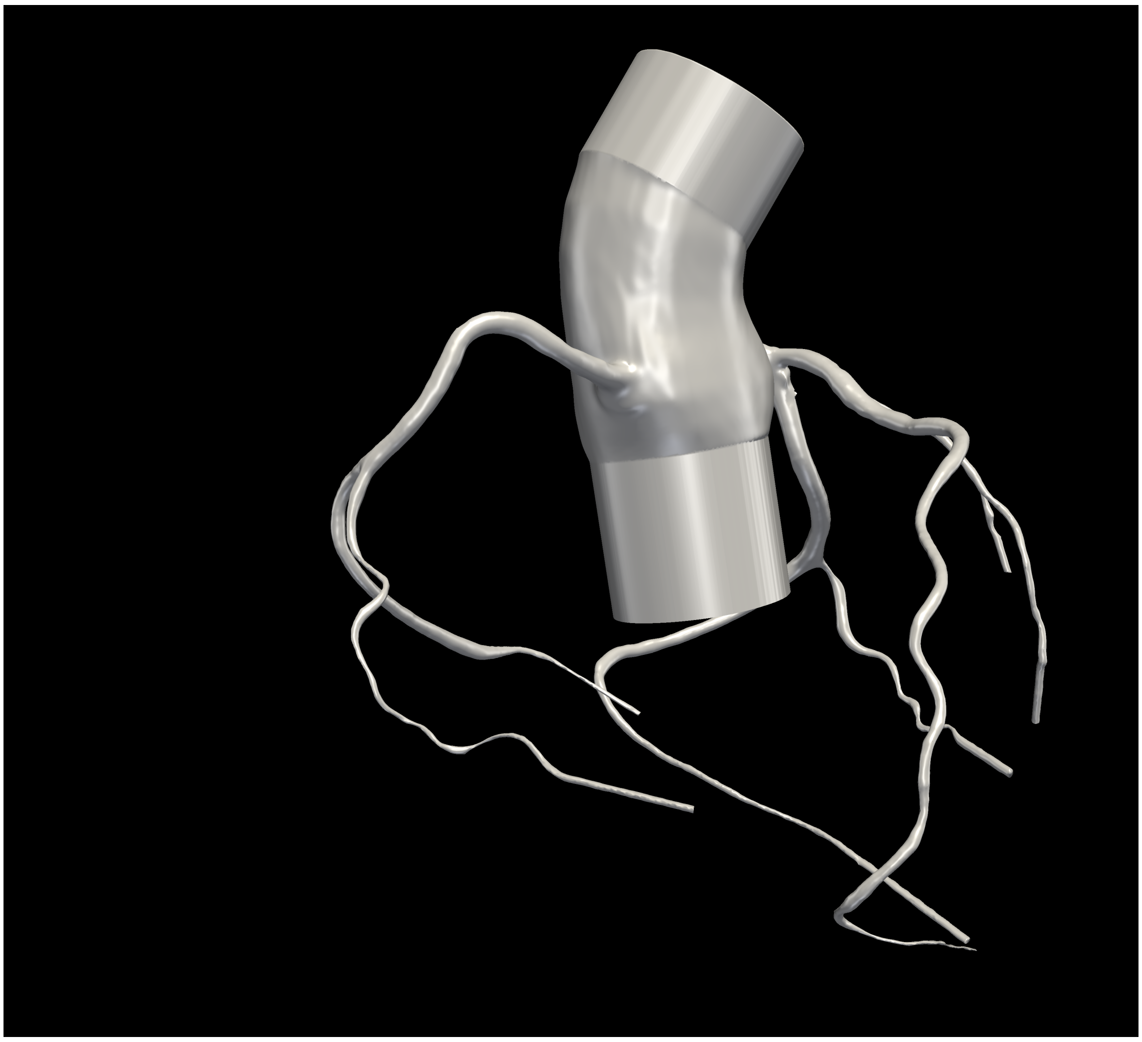

Figure 3.19: *Visualisation of the mesh with cut in/outlets*

However, without labels it is not easily possible to know which cell corresponds to which edge of the vessel.

The goal is to label the mesh and export it to a standard format. As export format STL was choosen, since it is supported by all mayor CFD tools and is easy to write. Each face needs to be labeled to the corresponding edge of the coronaries or to the aorta. This could be implemented with a distance calculation from each face to every centerline as used during the Section 3.1.4.3. However, this would need a high amount of computation. To minimize this effort the bounding box for each edge is calculated beforehand. Now a simple check up is performed in which bounding box the face is in. However, on branching regions multiple bounding boxes could overlap. If a face is member of multiple bounding boxes a distance comparison between each box and the face has to be performed.

To optimize the calculation effort, an octree representation of the mesh could be implemented, with this implementation not every face has to be checked. To optimize the file size, a conversion from the ASCII to a binary STL representation can be implemented.

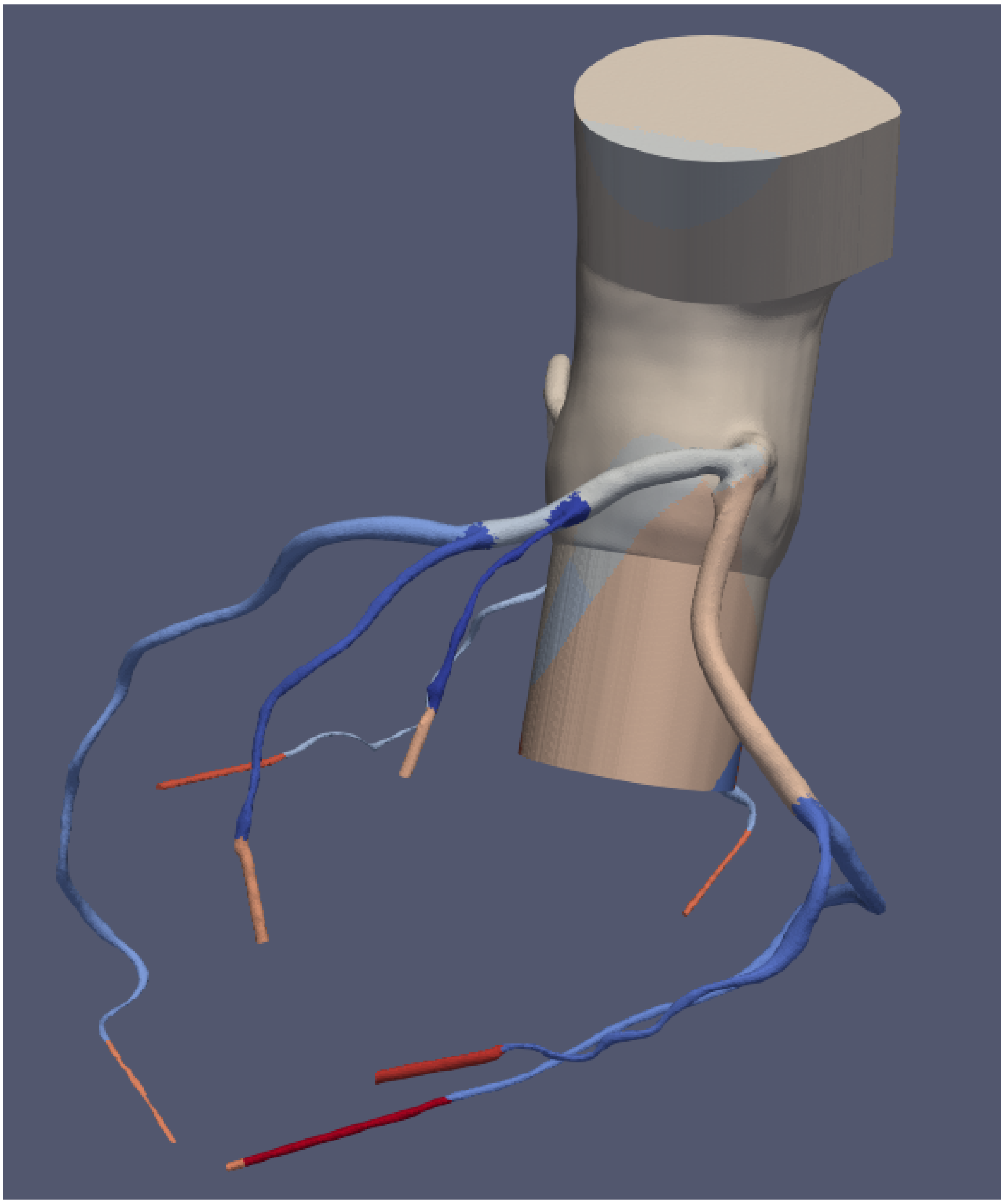

Figure 3.20: *Visualisation of the Colored Mesh*

4. Results and Discussion

To evaluate the results of the created Mesh, we have to either compare the created Mesh to a goldstandard or use some measureable quality criteria. Since we do not have access to a ground truth mesh or simulation, we are using some metrics to define the quality of the mesh.

4.1 Mesh Quality Criteria

To determine if a mesh is sufficient enough and results in a good simulation, we have to define some quality criterias for Meshes uses in a CFD Simulation. Mesh quality affects the quality of the CFD solutions drastically. Therefore we are using industy-standard mesh quality criterias to determine our Mesh quality.

Skewness

Skewness is defined as the difference between the shape of the cell and the shape of an equilateral cell of equivalent volume. Cells with high skewness can destabilize the solution and can even lead to non-convergence. As example, an optimal triangular mesh should have angles close to 60 degree, but with none higher than 90 degree. It is widely accepted that the maximum skewness for a triangular mesh should be kept below 0.95. An higher value might result in non-convergence and may require changes to the properties of the solver which would result in a less accurate simulation.
The skewness for a triangle mesh is defined as

$$Skewness = \frac{optimalCellSize - cellSize}{optimalCellSize} \tag{4.1}$$

Cell Aspect Ratio

The aspect ratio of a triangle is the ratio of the longest edge length to the shortest edge length. It should be qual to 1 for an ideal mesh. Local Variations should also be minimal, adjacent cell sizes should not vary by more than 20%. It can also be defined as the ratio of the circumradius to twice the inradius. The aspect ratio of a triangle lies between 0 and 1. The larger aspect ratio implies the better quality of the triangle.

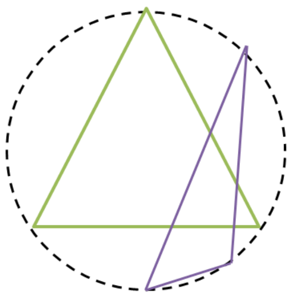

Figure 4.1: Skewness based on quilateral volume. The red triangle is the optimal equilateral cell, the blue triangle is the actual cell and the red circle is the circumcirle. [Com12]

4.1.1 Results

In this section, results to five different datasets are presented. Each dataset is unique and has own challenges which will be explained in detail. Furthermore at the end is a comparison and an overview of the datasets regarding the previous presented mesh quality criteria. Each dataset consists of an CTA-volume of the coronary, an aortic mask and centerlines of the coronaries.

4.1.1.1 Dataset 1

This dataset has seven centerlines, and therefore seven coronary arteries where segmented. All these have a high distance to each other on every point and therefore are not problematic for the creation of a mesh. If one looks at the top of the extruded aorta, it can be discovered that the shape of the aorta outlet is a bit off and does not seem to be correct. This is caused by the provided mask of the aorta. The last slice, which is copied for the extrusion, is not correctly segmented. This shows, furthermore that the results are heavily input dependent. Although the shape is not anatomically correct, it is not clear if this has a significant impact on the simulation, since it does not necessarily alter the flow of the coronaries.

The resulting mesh consists of 194808 triangular polygons and 97366 vertices.

4.1.1.2 Dataset 2

This dataset has 7 centerlines, and therefore seven coronary arteries where segmented. All these have a high distance to each other on every point and therefore are not problematic for the creation of a mesh. Here, the aorta extrusion of the aorta outlet is problematic. As seen, the inlet of the aorta is not extruded to a high extend. This is because a further extension would collide with an vessel. This is problematic and can lead to simulation anomalies. To counter the problem, a shorter extrusion was manually set so that this collision does not happen.
The resulting mesh consists of 236850 triangular polygons and 118321 vertices.

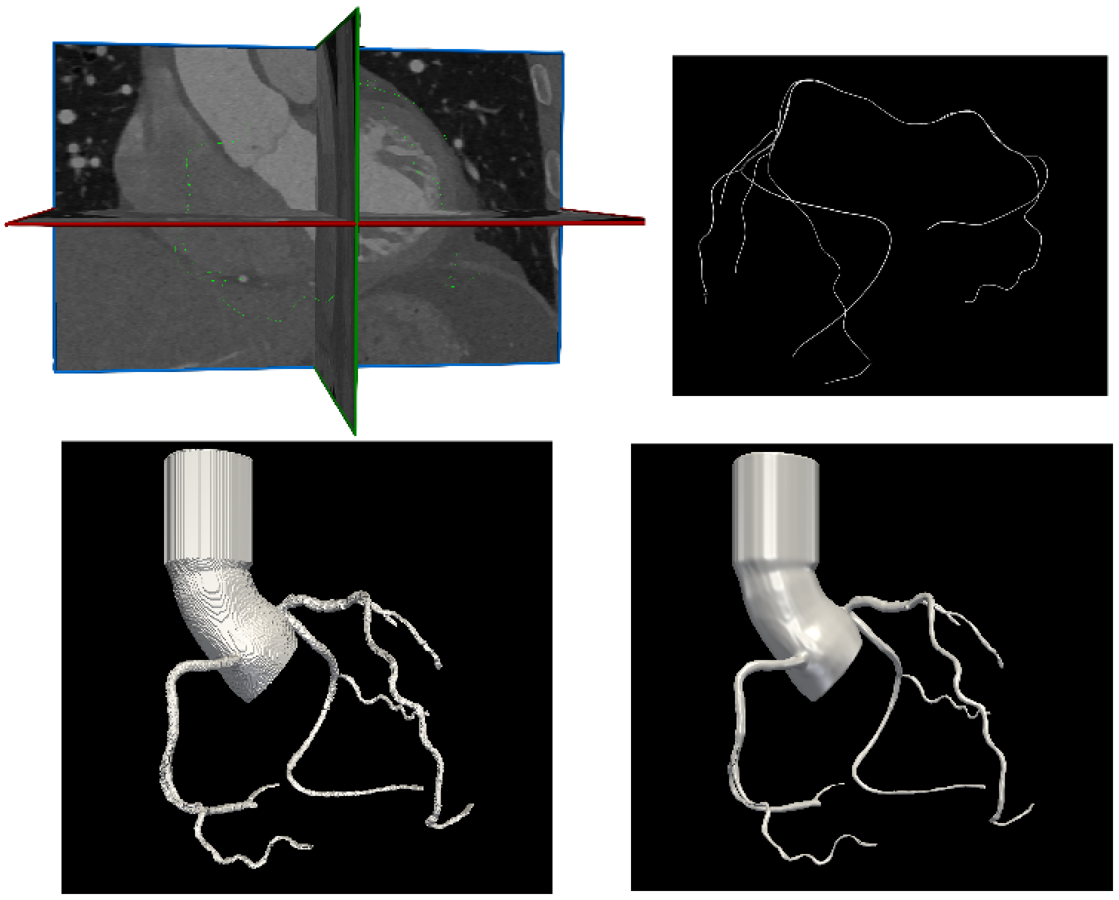

Figure 4.2: *Top left: Unprocessed input data. Top right: vascular strucutre. Bottom left: coarse mesh. Bottom right: smoothed mesh*

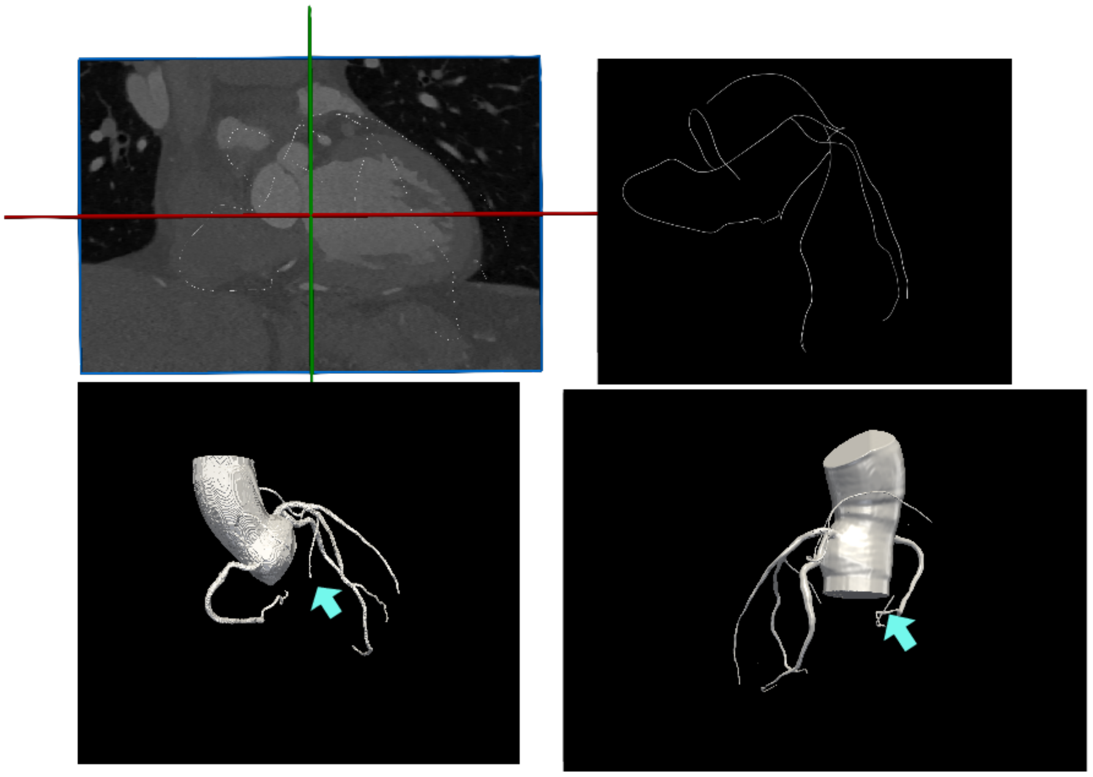

Figure 4.3: *Top left: Unprocessed input data. Top right: vascular strucutre. Bottom left: coarse mesh. Bottom right: smoothed mesh. The arrows indicates the vessel which would be merge by further extrusion of the aorta*

5. Conclusion and Future Work

5.0.1 Conclusion

In this thesis a novel pipeline for the creation of a simulation-ready mesh has been presented. The first part of the pipeline, describes the build up of a tree structure for coronary vessels and the necessary preprocessing steps for the data. Next required changes to this representation are described. Splitting of the root edge, path creation and extruding the tree.

In the second part, a pipeline for the creation and preparation of mesh has been presented requiring minimal user input. Like described this process starts with the creation of one mask consisting of several individual MPR masks. Afterwards this mask is combined with the aortic mask and a mesh is created. This mesh is smoothed and the outlets are cut so these can be recognized as inlet and outlet from CFD tools. Finally, every vertex of the mesh is labeled, so that it is possible to connect each part of the simulation to a part of the mesh.
A user has to define only a minimal amount parameters in order to run the whole pipeline.
Furthermore the whole pipeline can be adapted easily, since every step is independent of each other and can be run standalone.

As shown in the result section, the presented version is not robust enough to be used on a variety of datasets and further improvements have to be made.

5.0.2 Future Work

The most prevalent part of improvement is the extrusion of the aorta. Therefore a variety of options should be investigated. The mesh reconstruction relies on the aortic data aswell, segmentation of the aorta and the ostia could lead to an significant improvement. Nevertheless it has a higher computation effort and the provided aortic mask would not be used. An other option would be to not necessarily extrude the aorta straight, rather building a curved version so it does not lead to an intersection with any vessel.

For supporting physicians an easier matching of detected deformations and problem zones to the correct coronary artery, a labeling of the artery could be implemented. Of course, manually tagging each path or edge of the vesseltree is possible, but require a huge amount of time. Therefore reasearch should be made into automatic labeling on bases of previous works from Yang et. al [YBP+11].

This pipeline was tested on a limited amount of datasets. Therefore, it would be necessary to do a comprehensive test, especially in regards to simulation results with this pipeline. Investigation regarding alternatives to the marching cubes algorithm could be done, to improve time, stability and potentially make the smoothing step obsolete.

Techniques for further performance research regarding the usage of the GPU should be done. Frameworks like CUDA are easily accessible and look promising for huge performance boosts.

Bibliography

[BA07] G Bono and AM Awruch. Numerical study between structured and unstructured meshes for euler and navier-stokes equations. *Mecanica Computacional*, 26:3134–3146, 2007. (cited on Page 14)

[BHS+19] Marzia Buscema, Simone Hieber, Georg Schulz, Hans Deyhle, Alexander Hipp, Felix Beckmann, Johannes Lobrinus, and Bert Müller. Ex vivo evaluation of an atherosclerotic human coronary artery via histology and high- resolution hard x-ray tomography. *Scientific Reports*, 10 2019. (cited on Page 1)

[BKP+10] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy. *Polygon mesh processing*. CRC press, 2010. (cited on Page 12 and 13)

[CAD] Coronary artery disease (cad). https://www.cdc.gov/heartdisease/coronary_ad.htm. Accessed: 2021-01-09. (cited on Page 1)

[CF88] Peter Constantin and Ciprian Foias. *Navier-stokes equations*. University of Chicago Press, 1988. (cited on Page 11)

[CGV+11] Emanuele Cecchi, Cristina Giglioli, Serafina Valente, Chiara Lazzeri, Gian Franco Gensini, Rosanna Abbate, and Lucia Mannini. Role of hemodynamic shear stress in cardiovascular disease. *Atherosclerosis*, 214(2):249–256, 2011. (cited on Page 2)

[CMC+18] Nguyen Cuong, Nguyen Minh, Hoang Cuong, Phan Quoc, Ngo Anh, and Nguyen Hieu. Porosity estimation from high resolution ct san images of rock samples by using housfield unit. *Open Journal of Geology*, 08:1019–1026, 01 2018. (cited on Page 5)

[Com12] Wikimedia Commons. Skewness based on equilateral volume, 2012. (cited on Page 40)

[CR15] Lee Ann Campbell and Michael E Rosenfeld. Infection and atherosclerosis development. *Archives of medical research*, 46(5):339–350, 2015. (cited on Page 3)

[CSM07] Nicu D Cornea, Deborah Silver, and Patrick Min. Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on visualization and computer graphics*, 13(3):530, 2007. (cited on Page 9)

[DDD$^+$00] Pierre-André Doriot, Pierre-André Dorsaz, Lidia Dorsaz, Edoardo Benedetti, Pascal Chatelain, and Patrice Delafontaine. In-vivo measurements of wall shear stress in human coronary arteries. *Coronary artery disease*, 11:495–502, 10 2000. (cited on Page 1)

[DJBBD92] J Theodore Dodge Jr, B Greg Brown, Edward L Bolson, and Harold T Dodge. Lumen diameter of normal human coronary arteries. influence of age, sex, anatomic variation, and left ventricular hypertrophy or dilation. *Circulation*, 86(1):232–246, 1992. (cited on Page 35)

[Ede03] Herbert Edelsbrunner. Surface reconstruction by wrapping finite sets in space. In *Discrete and computational geometry*, pages 379–404. Springer, 2003. (cited on Page 13)

[FM03] M Farrashkhalvat and JP Miles. *Basic Structured Grid Generation: With an introduction to unstructured grid generation*. Elsevier, 2003. (cited on Page 13 and 14)

[Gib10] Peter Giblin. *Graphs, Surfaces and Homology*. Cambridge University Press, 3 edition, 2010. (cited on Page 13)

[HFCP06] Udo Hoffmann, Maros Ferencik, Ricardo C Cury, and Antonio J Pena. Coronary ct angiography. *Journal of nuclear medicine*, 47(5):797–806, 2006. (cited on Page 4)

[Ho09] Siew Yen Ho. Structure and anatomy of the aortic root. *European journal of echocardiography*, 10(1):i3–i10, 2009. (cited on Page 2)

[HR10] Vincent Ho and Gautham P Reddy. *Cardiovascular Imaging E-Book*. Elsevier Health Sciences, 2010. (cited on Page 11)

[KHH$^+$04] A. Kiraly, J. P. Helferty, E. Hoffman, G. McLennan, and W. Higgins. Three-dimensional path planning for virtual bronchoscopy. *IEEE Transactions on Medical Imaging*, 23:1365–1379, 2004. (cited on Page 9)

[KSW02] Avinash C Kak, Malcolm Slaney, and Ge Wang. Principles of computerized tomographic imaging, 2002. (cited on Page 4 and 5)

[Lee11] Byoung-Kwon Lee. Computational fluid dynamics in cardiovascular disease. *Korean circulation journal*, 41(8):423, 2011. (cited on Page 12)

[Leh05] Gaetan Lehmann. Binary morphological closing and opening image filters. *Africa Insight*, (141):5–p, 2005. (cited on Page 32)

[LKC94] Ta-Chih Lee, Rangasami L Kashyap, and Chong-Nam Chu. Building skeleton models via 3-d medial surface axis thinning algorithms. *CVGIP: Graphical Models and Image Processing*, 56(6):462–478, 1994. (cited on Page 21)

[Lyn10] Patrick J Lynch. Coronary arteries.svg, 2010. (cited on Page 3)

[MDMMEH18] Sara Moccia, Elena De Momi, Leonardo Mattos, and Sara El Hadji. Blood vessel segmentation algorithms — review of methods, datasets and evaluation metrics, 02 2018. (cited on Page 14 and 25)

[Mis10] Gabriel Mistelbauer. *Automated processing and visualization of vessel trees*. Citeseer, 2010. (cited on Page 21)

[MN11] S. Mendis and Bo Norrving. Global atlas on cardiovascular disease prevention and control: World health organization:. 924:156–4373, 01 2011. (cited on Page 1)

[NY06] Timothy S Newman and Hong Yi. A survey of the marching cubes algorithm. *Computers & Graphics*, 30(5):854–879, 2006. (cited on Page 33)

[PB13] Bernhard Preim and Charl P Botha. *Visual computing for medicine: theory, algorithms, and applications*. Newnes, 2013. (cited on Page 10)

[Pen16] Matt Penfold. *Creation of atherosclerosis*. Feb 2016. (cited on Page 4)

[SBdB16] Punam K Saha, Gunilla Borgefors, and Gabriella Sanniti di Baja. A survey on skeletonization algorithms and their applications. *Pattern recognition letters*, 76:3–12, 2016. (cited on Page 10)

[SEM$^+$11] Habib Samady, Parham Eshtehardi, Michael C McDaniel, Jin Suo, Saurabh S Dhawan, Charles Maynard, Lucas H Timmins, Arshed A Quyyumi, and Don P Giddens. Coronary artery wall shear stress is associated with progression and transformation of atherosclerotic plaque and arterial remodeling in patients with coronary artery disease. *Circulation*, 124(7):779–788, 2011. (cited on Page 2)

[SHF01] Frederick H Silver, Istvan Horvath, and David J Foran. Viscoelasticity of the vessel wall: the role of collagen and elastic fibers. *Critical Reviews™ in Biomedical Engineering*, 29(3), 2001. (cited on Page 2)

[SPSP02] Dirk Selle, Bernhard Preim, Andrea Schenk, and Heinz-Otto Peitgen. Analysis of vasculature for liver surgical planning. *IEEE transactions on medical imaging*, 21:1344–57, 12 2002. (cited on Page 10)

[Str06] Matúš Straka. *Processing and Visualization of Peripheral CT-Angiography Datasets*. PhD thesis, 2006. (cited on Page 4)

[TZG96] Gabriel Taubin, Tong Zhang, and Gene Golub. Optimal surface smoothing as filter design. In *European Conference on Computer Vision*, pages 283–292. Springer, 1996. (cited on Page 33)

[VM07] Henk Kaarle Versteeg and Weeratunge Malalasekera. *An introduction to computational fluid dynamics: the finite volume method.* Pearson education, 2007. (cited on Page 13)

[WLK$^+$02] Ming Wan, Zhengrong Liang, Qi Ke, Lichan Hong, Ingmar Bitter, and Arie Kaufman. Automatic centerline extraction for virtual colonoscopy. *IEEE transactions on medical imaging*, 21(12):1450–1460, 2002. (cited on Page 9)

[YBP$^+$11] Guanyu Yang, Alexander Broersen, Robert Petr, Pieter Kitslaar, Michiel A de Graaf, Jeroen J Bax, Johan HC Reiber, and Jouke Dijkstra. Automatic coronary artery tree labeling in coronary computed tomographic angiography datasets. In *2011 Computing in Cardiology*, pages 109–112. IEEE, 2011. (cited on Page 44)

[Zam06] Mair Zamir. *The physics of coronary blood flow.* Springer Science & Business Media, 2006. (cited on Page 2)

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Saarbrücken, den 24. April 2021