Automatic Vascular Model Construction from Medical Imaging Using Deep Learning

By

Númi Sveinsson Cepero

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Shawn C. Shadden, Chair
Professor Grace D. O'Connell
Professor Jon Wilkening

Summer 2025

Automatic Vascular Model Construction from Medical Imaging Using Deep Learning

Abstract

Automatic Vascular Model Construction from Medical Imaging Using Deep Learning

by

Númi Sveinsson Cepero

Doctor of Philosophy in Mechanical Engineering

University of California, Berkeley

Professor Shawn C. Shadden, Chair

Computational modeling of the cardiovascular system plays a vital role in understanding, diagnosing, and treating cardiovascular disease. However, traditional workflows for generating simulation-ready, patient-specific models are time-consuming, requiring extensive manual labor for geometric reconstruction and simulation setup. This dissertation introduces deep learning based methods designed to automate and accelerate the construction of image-based models to support hemodynamics simulation.

First, we present SeqSeg (Sequential Segmentation), a novel deep learning method for automatic vascular segmentation. SeqSeg leverages a local U-Net-based architecture to iteratively track and segment vascular structures from medical imaging data. Compared to standard 2D and 3D global models such as nnU-Net, SeqSeg generates more complete vascular models and generalizes better to unannotated anatomy, enabling efficient geometric modeling from computed tomography (CT) and magnetic resonance (MR) data.

Building upon this, we introduce MeshGrow, an integrated framework that combines automatic vascular and cardiac modeling to generate combined cardiovascular anatomies. MeshGrow can reconstruct both the heart and great vessels by employing a template deformation approach for the cardiac chambers and a step-wise growth-based method for vascular structures. The result is a simulation-ready mesh, including valve boundaries, constructed directly from medical images with minimal human intervention.

In the third part of this work, we present MIROS (Medical Image to Reduced Order Simulation), a fully automated pipeline for performing reduced-order cardiovascular simulations. MIROS integrates SeqSeg-based geometry generation with reduced order modeling of blood flow and semi-automatic boundary condition assignment to produce hemodynamic simulations within minutes. This approach significantly reduces the computational and manual burden traditionally required, enabling rapid, patient-specific analyses and facilitating large-scale studies.

Finally, building on SeqSeg and inspired by advances in human trajectory forecasting, we propose VesselTrajNet, a novel method for vasculature tracking in medical images. By adapting a U-Net-based Gaussian heat map encoder-decoder architecture for multiple goal-driven path prediction, VesselTrajNet accurately models complex vascular branching without requiring explicit bifurcation detection. We demonstrate its utility on coronary artery CT data, underscoring its potential for diagnostic and interventional imaging.

Together, these contributions advance the state of the art in automated cardiovascular modeling and simulation. By harnessing deep learning for the modeling pipeline, this work aims to make high-fidelity cardiovascular simulations more accessible, scalable, and clinically relevant.

To the people in my life.

I am deeply grateful to my parents, my sister, my extended family, and all my friends in Iceland, the Bay Area, and around the world.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to express my deepest gratitude to my advisor, Prof. Shawn C. Shadden, for his mentorship, encouragement, and support throughout my PhD. His guidance shaped not only the direction of my research but also how I think as a scientist and engineer.

I am also sincerely grateful to Prof. Alison Marsden at Stanford University and Adjunct Assistant Prof. Nathan Wilson at UCLA for their invaluable feedback and guidance throughout my work with SimVascular. Along with David Parker, Senior Research Software Engineer, they have each played a crucial role in helping me grow as a researcher and in advancing my work in cardiovascular modeling and simulation.

Special thanks to Dr. Martin Pfaller from the Marsden Lab for his insight and collaboration early on in my PhD, as well as to Dr. Fanwei Kong and Dr. Arjun Narayanan from the Shadden Lab. Arjun and I had the pleasure of co-authoring a paper together, and I'm thankful for his thoughtful contributions and collegial spirit. I also deeply appreciated the opportunity to work with Boyang Gan (Columbia University) and Neerja Thakkar (Berkeley AI Research), both of whom brought energy and sharp thinking to our joint work.

I am grateful to all the members of the Shadden Lab and the Marsden Lab for their ideas, feedback, and camaraderie over the years.

I would also like to thank the members of my qualifying and dissertation committees—Prof. Grace O'Connell, Prof. Jon Wilkening, and Prof. Ömer Savaş—for their time, support, and thoughtful questions that helped shape and refine my work.

I was supported by NSF Awards 1663747 and 2310910, NIH Award 5R01LM013120, as well as fellowships from the UC Berkeley Department of Mechanical Engineering, the Leifur Eiríksson Foundation, the H2H8 Fellowship, and Landsbankinn. I am truly thankful for this generous support, which made my research and doctoral studies possible.

To my friends in the Berkeley graduate community: thank you for the countless coffees, hikes, and weekend trips that helped me stay grounded, energized, and connected. Your friendship was just as vital to this journey as the research itself.

Lastly, I am deeply thankful to my family and friends outside of Berkeley for their steady encouragement and love throughout this long and winding journey. Your belief in me, even from afar, has meant more than words can express.

Chapter 1

Introduction

1.1 Background and Motivation

Cardiovascular disease remains a leading global cause of morbidity and mortality, resulting in significant suffering through disability, pain, and premature death. Computational modeling of the cardiovascular system provides a patient-specific framework for understanding disease mechanisms, predicting progression, and informing treatment design. Specifically, image-based cardiovascular modeling is used for a variety of purposes including diagnosing, personalized treatment planning and fundamental understanding of disease progression [86, 79, 6, 109]. In this modeling paradigm, a medical image is used for patient-specific reconstruction of anatomy that is subsequently used to model fluid dynamics, tissue mechanics, electromechanics or mass transport of the organ of interest (e.g. cardiac chambers, vasculature, valves).

Figure 1.1 shows typical patient-specific computational modeling workflow. The inputs to the patient-specific model is processed personalized data (e.g. mesh of the aorta) and non-personalized data (e.g. average viscosity of blood). The patient-specific model includes the governing equations of interest, for example the Navier-Stokes equations for fluid dynamics, as well as necessary initial and boundary conditions. Then a numerical solver is used to solve the governing equations on the patient-specific domain given the initial and boundary conditions. The output solutions, simulations, can then be used to make personalized clinical predictions.

Specialized software has been developed for such image-based modeling of vasculature, including SimVascular [123, 53], CRIMSON [5] and VMTK [45]. This modeling paradigm uses medical imaging, such as computed tomography (CT) or magnetic resonance (MR) angiography, to construct a patient-specific anatomical model of vessels of interest. This geometric model is subsequently converted into a 3D computational mesh to support detailed blood flow and/or tissue mechanics simulation and analysis. The construction of an anatomical model from medical image data remains largely a manual process [123]. This process, noted as ‘data processing‘ in Figure 1.1, usually requires hours or days of manual

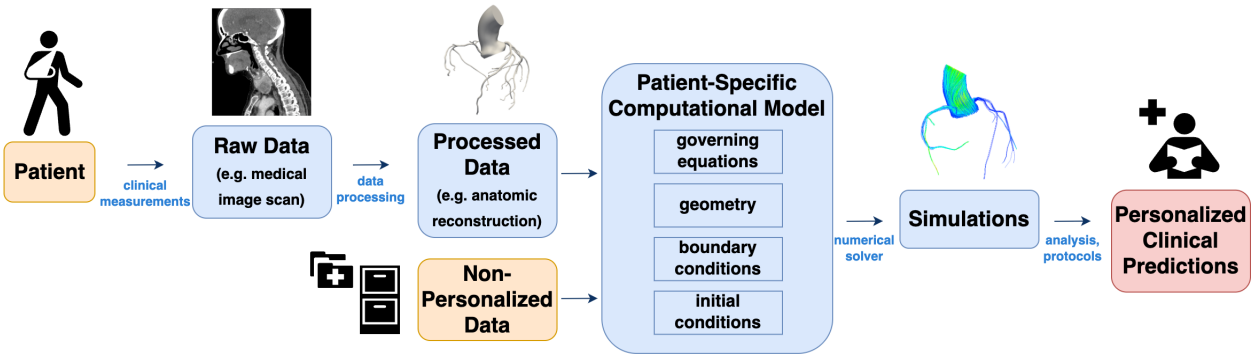


Figure 1.1: Patient-specific computational modeling workflow. Inputs to the model may be non-personalized data or personalized data which is derived from raw clinical measurements (e.g. medical imaging) and requires data processing. The patient-specific model consists of the patient-specific geometry, the governing equations of interest, boundary conditions and initial conditions. Simulations for the computational model are solved for using numerical solvers and their results are analyzed for personalized clinical predictions.

input from an expert, making it a heavy bottleneck in many research workflows.

Figure 1.2 shows a typical workflow for vascular model construction, starting with the creation of centerlines along the vessels of interest, 2D segmentation of the vessel lumen along the centerlines, and lofting of the 2D segmentations to generate a unified 3D model of the vascular geometry. Alternative segmentation approaches exist, including region-growing or level-set methods [53]; however, these methods generally struggle in the segmentation of highly-branched structures such as blood vessels, particularly in the context of limited image resolution, unclear boundaries and image artifacts [83]. Additionally, when the model is constructed manually, substantial user bias may result. Ultimately, despite the popularity and maturity of image-based cardiovascular modeling over the past 20 years, the process of deriving a simulation-suitable anatomical model from medical image data has remained a primary bottleneck for large-cohort studies or translational applications where timely results are needed.

Recently, deep learning has been applied to automate and speed up medical image segmentation[90, 42, 66, 104, 14, 77, 4, 102, 61]. Note that while medical image segmentation is performed for a variety of healthcare applications, we focus here on the purpose of generating a *simulation-suitable* model that can be utilized to generate a computational domain for physics-based simulation. Simulation suitable models have certain criteria that must be met such as, being connected, sufficiently "smooth", and able to be meshed (discretized) with quality elements.

Deep learning, a subfield of machine learning, involves the use of *deep* neural networks (DNNs) that learn to make predictions based on training data. During training, these

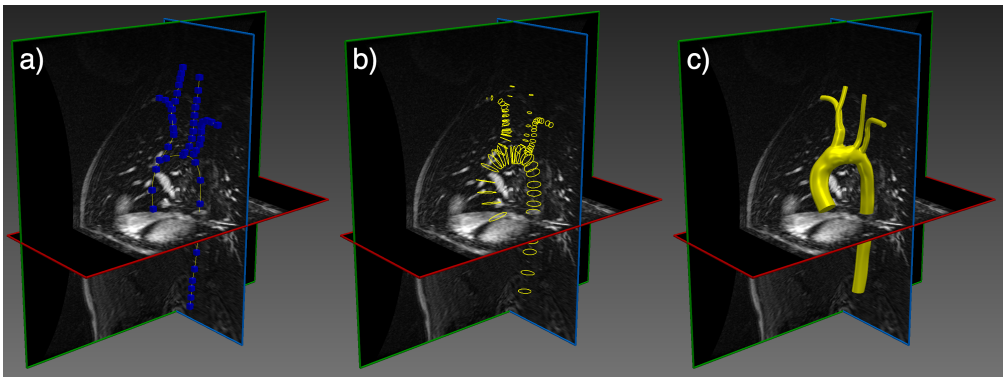


Figure 1.2: A typical vascular model construction workflow involves (a) creating vessel paths by manual selection of point (b) sequential segmentation of the vessel lumen boundary at discrete cross-sections along the paths and (c) lofting these segmentation rings into a unified model. This process is described in more detail in [123].

networks iteratively adjust their internal parameters to extract and utilize salient features relevant to the prediction task. Once trained, the network can generalize to previously unseen data. The widespread adoption of deep learning began in the early 2010s, driven by the development of efficient training algorithms such as backpropagation and the use of graphical processing units (GPUs), specialized hardware originally designed for rendering computer graphics, which significantly accelerated training times. As of 2025, deep learning underpins most state-of-the-art artificial intelligence applications, including those in medical imaging.

Most deep learning methods for medical image vascular segmentation focus on pixel classification[57, 112, 47, 38, 14, 77, 119, 55, 83]. The popular nnU-Net is a convolutional neural network (CNNs) with a U-like model architecture and can be viewed as the current state-of-the-art[42]. However, deep learning models trained for pixel classification often predict segmentations that are disconnected or have substantial artifacts that complicate, or prevent, generation of a mesh suitable to support numerical simulation. Pixel classification task does not require topological knowledge for prediction, but instead leads to models trying to maximize the number of pixels correctly classified. That task is inherently different from the goal of constructing simulation-suitable vascular models.

Some progress has been made when deep learning has been applied to isolated anatomic vascular regions [89, 13] including for cardiac models [50, 52]. In the work of Maher, et al. [67, 68] segmentation of branched vascular domains was achieved by assuming the existence of vessel centerlines (cf. Fig. 1.2a). Under such assumptions, these centerlines are traversed and local 2D cross-sectional segmentations of the lumen boundary are generated using a trained network. This framework essentially automated step (b) shown in Fig. 1.2. However, for

many vascular models, the generation of vessel centerlines is the most labor intensive step. Moreover, with this approach, segmentation is only performed at discrete 2D slices along the vessel, which provides incomplete sampling and can be problematic when the cross-section is not connected or the centerline is not sufficiently aligned with the vessel. And more importantly, discrete cross-sectional segmentation performs poorly at vessel bifurcations, which are present in almost all applications.

The goal of this thesis is to develop methods for automatic vascular model construction for simulation purposes with the aim of enabling clinical impact and research of patient-specific vascular hemodynamic simulations.

Firstly, a novel automatic blood vessel segmentation and tracking method, SeqSeg (short for Sequential Segmentation), is presented in Chapter 2. Afterwards, in Chapters 3, 4 and 5, this thesis presents extensions to this work, specifically for integrated cardiovascular modeling, reduced order modeling of hemodynamics and vascular tracking. In Chapter 3, SeqSeg is combined with automatic cardiac modeling to introduce a novel integrated framework for comprehensive cardiovascular modeling given the name MeshGrow. Then in Chapter 4, the rapid automatic geometric modeling of vasculature using SeqSeg is combined with lower computational cost of reduced order models of hemodynamics to obtain patient-specific hemodynamic simulations on the order of minutes. Finally in Chapter 5, this thesis presents novel approach inspired by human motion tracking methods in computer vision to automatically track blood vessels in medical images. This approach is designed to overcome a key limitation of SeqSeg's segmentation-dependent strategy; its reduced ability to reliably detect bifurcations in low-contrast or artifact-laden image data.

1.2 Prior Publication and Software Availability

Parts of this thesis have been published in journals and at conferences. All code was developed in Python and is open-source and available for use on GitHub[1].

- Chapter 2; the code is fully available on GitHub[2] and PyPi[3], compiled for all major operating systems: macOS, Windows, and Ubuntu. The work was previously published: Numi Sveinsson Cepero and Shawn C Shadden. "SeqSeg: Learning Local Segments for Automatic Vascular Model Construction". *Annals of Biomedical Engineering* 53.1 (2025), pp. 158–179. doi: 10.1007/s10439-024-03611-z.

- Chapter 3 ; in addition to SeqSeg code, code is available on GitHub[4]. This work was done in conjunction with the following publication: Numi Sveinsson Cepero, Arjun Narayanan, and Shawn C. Shadden. "Integrated Framework for Unified Cardiac and Vascular Mesh Construction from Medical Images". *Functional Imaging and Modeling*

[1]https://github.com/numisveinsson
[2]https://github.com/numisveinsson/SeqSeg
[3]https://pypi.org/project/seqseg/
[4]https://github.com/ArjunNarayanan/LinFlo-Net

of the Heart: 13th International Conference, FIMH 2025, Dallas, TX, USA, June 1–5, 2025, Proceedings, Part II. Lecture Notes in Computer Science. Vol. 1. Springer Cham, 2025, pp. 98–109.

- Chapter 4; This work was done in collaboration with Boyang Gan. In addition to SeqSeg code, code is available on GitHub[5]
- Chapter 5; This work was done in collaboration with Neerja Thakkar. Code is expected to be released on Github[6] once paper is published.

[5]https://github.com/BryannGan/MIROS
[6]https://github.com/numisveinsson

Chapter 2

SeqSeg: Learning Local Segments for Automatic Vascular Model Construction

2.1 Abstract

Computational modeling of cardiovascular function has become a critical part of diagnosing, treating and understanding cardiovascular disease. Most strategies involve constructing anatomically accurate computer models of cardiovascular structures, which is a multistep, time-consuming process. To improve the model generation process, we herein present SeqSeg (sequential segmentation): a novel deep learning based automatic tracking and segmentation algorithm for constructing image-based vascular models. SeqSeg leverages local U-Net-based inference to sequentially segment vascular structures from medical image volumes. We tested SeqSeg on CT and MR images of aortic and aortofemoral models and compared the predictions to those of benchmark 2D and 3D global nnU-Net models, which have previously shown excellent accuracy for medical image segmentation. We demonstrate that SeqSeg is able to segment more complete vasculature and is able to generalize to vascular structures not annotated in the training data.

2.2 Introduction

Image-based vascular modeling is used for a variety of purposes including diagnosis, personalized treatment planning and fundamental understanding of disease progression [86, 79, 6, 109]. Specialized software has been developed for such modeling, including SimVascular [123, 53], CRIMSON [5] and VMTK [45]. This modeling paradigm uses medical imaging, such as computed tomography (CT) or magnetic resonance (MR) angiography, to construct a patient-specific anatomical model of vessels of interest. This geometric model is subsequently converted into a 3D computational mesh to support detailed blood flow and/or tissue me-

chanics simulation and analysis. The construction of an anatomical model from medical image data remains largely a manual process [123]. Figure 2.1 shows a typical workflow for vascular model construction, starting with the creation of centerlines along the vessels of interest, 2D segmentation of the vessel lumen along the centerlines, and lofting of the 2D segmentations to generate a unified 3D model of the vascular geometry. Alternative segmentation approaches exist, including region-growing or level-set methods [53]; however, these methods generally struggle in the segmentation of highly-branched structures such as blood vessels, particularly in the context of limited image resolution, unclear boundaries and image artifacts [83]. Additionally, when the model is constructed manually, substantial user bias may result. Ultimately, despite the popularity and maturity of image-based cardiovascular modeling over the past 20 years, the process of deriving a simulation-suitable anatomical model from medical image data has remained a primary bottleneck for large-cohort studies or translational applications where timely results are needed.

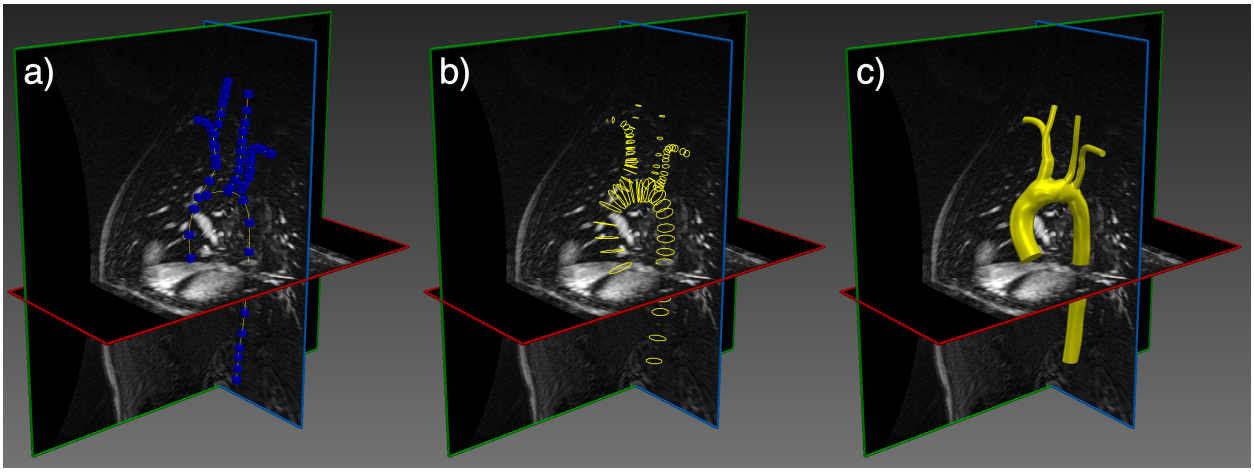


Figure 2.1: A typical vascular model construction workflow involves (a) creating vessel paths by manual selection of point (b) sequential segmentation of the vessel lumen boundary at discrete cross-sections along the paths and (c) lofting these segmentation rings into a unified model. This process is described in more detail in [123].

Recently, machine learning has been applied to automate and speed up image segmentation. Note that while medical image segmentation is performed for a variety of healthcare applications, we focus here on the purpose of generating a *simulation-suitable* model that can be utilized to generate a computational domain for physics-based simulation. Simulation suitable models have certain criteria that must be met such as, being connected, sufficiently "smooth", and able to be meshed (discretized) with quality elements. Most learning methods focus on pixel classification, which often results in segmentations that are disconnected

or have substantial artifacts that complicate, or prevent, generation of a mesh suitable to support numerical simulation.

Most progress has been made when machine learning has been applied to isolated anatomic vascular regions [89, 13] including for cardiac models [50, 52]. In the work of Maher, et al. [67, 68] segmentation of branched vascular domains was achieved by assuming the existence of vessel centerlines (cf. Fig. 2.1a). Under such assumptions, these centerlines are traversed and local 2D cross-sectional segmentations of the lumen boundary are generated using a trained network. This framework essentially automated step (b) shown in Fig. 2.1. However, for many vascular models, the generation of vessel centerlines is the most labor intensive step. Moreover, with this approach, segmentation is only performed at discrete 2D slices along the vessel, which provides incomplete sampling and can be problematic when the cross-section is not connected or the centerline is not sufficiently aligned with the vessel. And more importantly, discrete cross-sectional segmentation performs poorly at vessel bifurcations, which are present in almost all applications.

Herein, we present a novel method for segmenting branched vascular geometries from medical image data utilizing local deep learning-based segmentation that does not require *a priori* centerline information. This approach starts from a seed point and generates a local 3D segmentation of the vessel(s) containing the seed point over a local subvolume. Based on this local segmentation, we determine the orientation of the vessel and any locally connected branches. We then step the subvolume along the determined vessel direction (and new subvolumes along the identified local branch directions) to generate a 3D segmentation of the neighboring segment(s). This approach is motivated by the following considerations: when viewed locally by a subvolume that is centered on a vessel and slightly larger than the vessel diameter, vessels of different sizes and from different regions exhibit substantial geometric similarity (Figure 2.2), and consequently learning to locally segment a portion of a vessel should be easier than learning to segment an entire vascular network. While cropping of medical image volumes has been performed previously, for example, for coronary tracking [127] [58], to the best of our knowledge such approach has not been used to generate 3D segmentation or for segmentation of general vascular geometries.

By processing local subvolumes, we simplify the deep learning task and introduce beneficial inductive bias to the machine learning model, allowing it to generalize to vasculature not present in training data. We test this method on a dataset of publicly-accessible aortic and aortofemoral models, and the results are compared to benchmark global 2D/3D nnU-Net neural network models that have previously shown excellent results for medical image segmentation. The main contribution of this work is a new method capable of:

- Tracking vasculature after initialization with a single point and vessel radius estimate.
- Segmenting vasculature while ensuring global connectivity to maintain physiologic topology.
- Detecting bifurcations, storing them and tracking them sequentially.

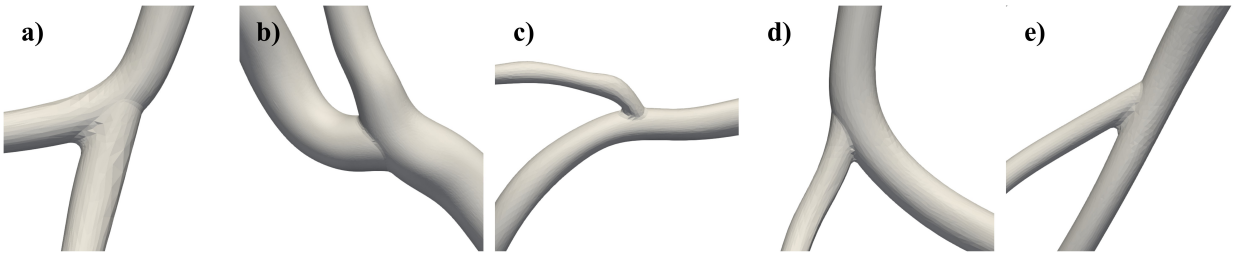


Figure 2.2: When viewed locally, vasculature of different sizes and anatomical regions exhibit substantial geometric similarity. A) the pulmonary artery ($r = 1.5mm$), b) the brachiocephalic artery ($r = 9mm$), c) the coronary artery ($r = 1mm$), d) the cerebral artery ($r = 2mm$) and e) the femoral artery ($r = 3mm$) are presented

- Delivering a global surface mesh of segmented vasculature.
- Generalizing to segment parts of vasculature not annotated in training data.

2.3 Method

Algorithm

Figure 2.3 shows a schematic of the algorithm. Breifly, a "seed point", (specifying a location and direction) and a rough diameter "size estimate" of the vessel containing the seed point are supplied by the user. A local subvolume surrounding the seed point is extracted from the global image volume. The vessel portion contained in the subvolume is segmented using a neural network. The segmentation is postprocessed and converted to a surface mesh, after which a centerline is extracted. The resulting centerline is subsequently used to choose the next subvolume location and size. These steps are explained in further detail below.

Segmentation

Dataset, Sampling and Augmentations

To train the U-Net and test the algorithm, we utilized a dataset of 41 CT and 44 contrast enhanced MRI aortic and aortofemoral cases, which is commensurate with the amount of data typically provided in segmentation challenges. The breakdown of how many cases are used for training, validation and testing is specified in Table 2.1. The VMR datasets used for training are accessible from the open access Vascular Model Repository at `https://vascularmodel.com`. For further testing on CT images, we also use a subset of the AVT dataset [65], specifically the dataset obtained from Dongyang Hospital. Table 2.1

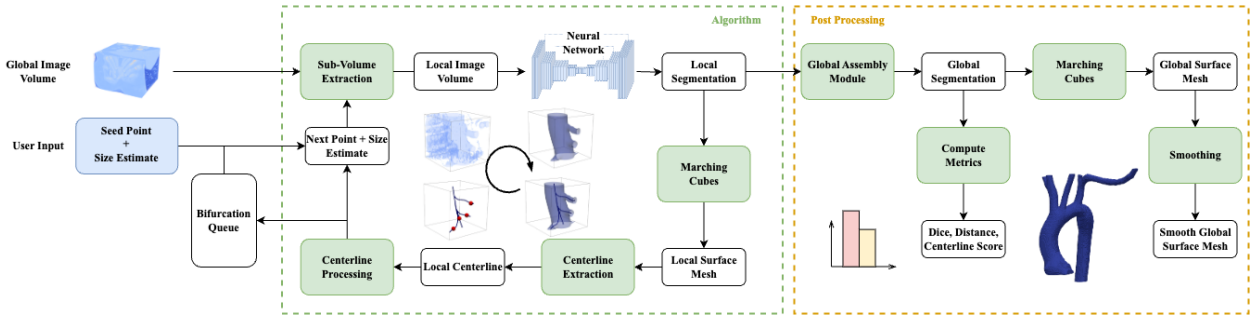


Figure 2.3: Overview of the tracking and segmentation algorithm with inputs of the global raw image and seed points for initialization. The algorithm takes steps, stores bifurcations in the queue during tracking, and outputs a global segmentation map for post processing

Table 2.1: The datasets of patients used for model training and method testing. Abbreviation are as follows: Datasets; VMR: Vascular Model Repository, AVT-D: Aortic Vessel Tree dataset, subset from Dongyang Hospital. Anatomy; AO:Aorta, AF:Aortofemoral. Disease; H:Healthy, AAA:Abdominal Aortic Aneurysm, MA: Marfan Syndrome, CA:Coarctation of Aorta, AOD:Aortoiliac Occlusive Disease, SVD:Single Ventricle Defect. Sex; M:Male, F:Female, U:Unknown. Sex and age information was not available for the AVT dataset.

Dataset	Modality	Purpose	Train/Test	Anatomy	Disease	Sex	Age(yr)
VMR	CT	Train/Test	33/8	25 AO, 16 AF	23 H, 15 AAA, 3 MA	23M 6F 12U	6 - 80 ave: 58
VMR	MR	Train/Test	37/7	38 AO, 6 AF	19 H, 14 CA, 5 SVD, 2 MA 4 AOD	30M 14F	0.6 - 67 ave: 17
AVT -D[65]	CT	Test	0/18	18 AO	18 H	-	-

shows details on the datasets; modalities, purpose, training/test split, anatomies, diseases (if present), sex ratio and age ranges. The datasets contain a 3D image volume and a respective "ground truth" vascular segmentation map (converted from segmentation surface models for the VMR data) and corresponding centerlines that served as ground truth labels for training and testing.

To generate training data for the local segmentation U-Net, the global 3D medical image

volumes in the VMR training datasets were sampled along the centerlines and these subvolumes (i.e. "Patches") were stored. Namely, two volumes were extracted at each Patch: 1) the original medical image data and 2) a binary segmentation of the subvolume based on the model representing the ground truth label.

To improve the learning process, we varied the samples in terms of centering and size. Briefly, some samples were centered along the centerline while others were shifted from the centerline, and the subvolume sizes varied from just capturing the lumen of the vessel to including more surrounding tissue. More specifically, each sample volume s_i is a cube dependent on its side length and center, i.e., $s_i(L_i, \mathbf{c}_i)$ where L_i is its side length and $\mathbf{c}_i$ is the center point of sample i. The side length and center are sampled as follows:

$$\begin{aligned} L_i &= R_i * \alpha_i \\ \mathbf{c}_i &= \mathbf{C}_i + \beta_i * R_i * \mathbf{w}_i \\ \alpha &\sim \mathcal{N}(\mu_r,\, \sigma_r^2) \\ \beta &\sim \mathcal{N}(\mu_s,\, \sigma_s^2) \end{aligned} \tag{2.1}$$

where R_i is the local radius of the vessel, $\mathbf{C}_i$ is the point on the centerline, $\mathbf{w}$ is a unit vector perpendicular to the centerline and α, β represent the radius ratios used to enlarge or shift the sample. $\mathbf{w}$ was chosen by sampling a random linear combination of orthogonal unit vectors $\mathbf{u}, \mathbf{v}$ that defined a plane perpendicular to the centerline:

$$\mathbf{w}_i = \frac{a_i * \mathbf{u} + b_i * \mathbf{v}}{\|a_i * \mathbf{u} + b_i * \mathbf{v}\|}, \quad a, b \sim U[-1, 1] \tag{2.2}$$

where a, b are scalars sampled from a uniform distribution between $[-1, 1]$. We used $(\mu_r,\, \sigma_r^2) = (5, 1)$ and $(\mu_s,\, \sigma_s^2) = (0, 0.8)$ so that the mean sample was 5 times the size of the radius and centered on the centerline. This stochasticity was purposefully added to represent the variance that the tracking algorithm encounters during inference and is intended to increase the robustness of the neural network. This process resulted in a training dataset D consisting of N pairs of image subvolumes X_i and corresponding blood vessel segmentations $Y_{t,i}$:

$$D = \{(X_1, Y_{t,1}), (X_2, Y_{t,2}), ..., (X_N, Y_{t,N})\} \tag{2.3}$$

In total, we get $D = 36289$ patches for CT data and $D = 33603$ patches for MR data. For the VMR dataset 15 patient datasets were excluded for final testing, i.e., 8 CT and 7 MR cases were not sampled for training or validation. The generation of subvolume data for network training is shown schematically in Fig. 2.4.

Before training, MR image volumes were normalized via z-scoring, where each voxel value, x, is subtracted from the image mean μ and then divided by the image standard deviation σ. CT volumes were clipped and z scored according to foreground image values where μ and σ are calculated only from voxels labeled as vessel in the ground truth training data and held constant during inference [42]; see Table 2.2 for details. The preferred image spacing was chosen as the median spacing across all cases and all image volumes were resampled using a 3rd order spline. Segmentation maps were resampled differently, using linear splines on one-hot encoded maps, and then the argmax of the result.

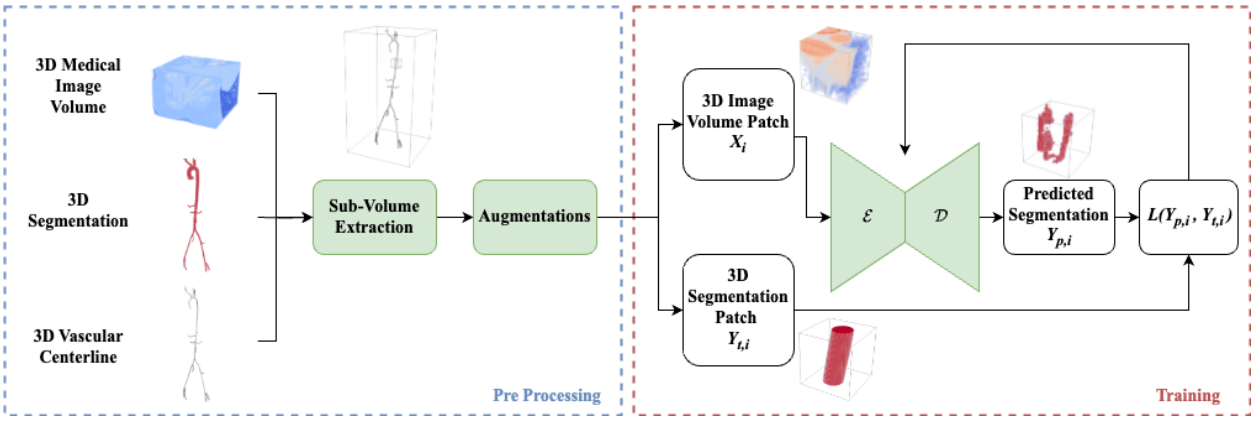


Figure 2.4: Preprocessing involves extracting subvolumes along ground truth centerlines and data augmentation prior to neural network training. Thousands of samples are acquired from only a few dozen models. The neural network consists of an encoder $\mathcal{E}$ followed by a decoder $\mathcal{D}$, which outputs the predicted segmentation map used to compute loss, L, during training

Neural Network Architecture and Training

The convolutional neural network (CNN) U-Net architecture was used for segmentation. The U-Net is tailored to the processing of medical images, going from the original resolution down to a low-resolution, high-dimensional space and then back up to the original resolution. Herein, a 3D version of U-Net was chosen for the SeqSeg segmentation step. Note, we also compare our end results to those of two benchmark U-Net models, i.e. a 2D U-Net and a 3D U-Net, trained on the global image volumes, see Section 2.3 for further details. The U-Net learns features primarily through two different mechanisms. First, by downsampling the original image data the model is forced to retain only important global information when squeezed through a lower-dimensional space. Second, by using skip connections across the neural network the model is able to retain features related to finer details from the higher resolution image in its final prediction. The skip connections are concatenations of blocks of the same resolution. The final output is a pixelwise probability map indicating the likelihood that a pixel corresponds to a target tissue.

Our U-Net was trained for binary classification: to predict whether voxel $y_{i,j,k}$ inside segmentation mask $Y_p \in \mathbf{R}^{W \times H \times D}$ belongs to a blood vessel:

$$
\begin{aligned}
Y_p &= \{y_{i,j,k} \in [0,1] \mid 0 \leq i < W; 0 \leq j < H; 0 \leq k < D\} \\
y_{i,j,k} &= \begin{cases} 1 & \text{if belongs to vessel} \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{2.4}
$$

where i, j, k refers to the index of a voxel in an image of width W, height H and depth D. In this section, lower case notation refers to individual nodes or voxels, e.g. y, whereas capital notation, e.g. Y, refers to a set of nodes or voxels such as composing an image, segmentation mask or output from neural network layers.

In mathematical terms, the neural network is a parameterized function f_θ that transforms a raw image input $X \in \mathbf{R}^{W \times H \times D}$ into a blood vessel segmentation map:

$$Y_p = f(X \mid \theta) \tag{2.5}$$

where θ are the parameters of the neural network, which are optimized using training data. The final output, Y_p, ranges between $[0, 1]$ and can thus be interpreted as a probability map of whether each voxel belongs to a blood vessel. This enables the volume to be binarized by thresholding to a particular probability value.

We utilized the nnU-Net framework for hyperparameter specification and training [42]. The framework automatically determines parameters such as image resampling spacing, patch size and batch size based on training data and GPU memory size. The underlying neural network architecture used is the U-Net, with additional constraints on specific parameters. Table 2.2 lists the specifications of our implemented U-Net model architectures and training parameters. Since the SeqSeg model is trained on smaller volumes compared to the benchmarks, its required batch size can be larger, see Table 2.2. The nnU-Net framework utilizes stochastic gradient descent with Nesterov momentum with an initial learning rate of 0.01 accompanied by a learning rate scheduler of $(1 - epoch/epoch_{max})^{0.9}$, where $epoch_{max} = 1000$ was chosen; see [42] for further details. Training was performed using an NVIDIA Geforce RTX 2080ti GPU (11 GB GPU memory) on the Savio High Performance Computing cluster at the University of California, Berkeley.

Loss Function

The loss function was a combination of Dice score ($\mathcal{D}$) and binary cross-entropy ($\mathcal{CE}$):

$$\mathcal{D}(Y_p, Y_t) = \frac{2 \cdot \|Y_p \cap Y_t\|}{\|Y_p\| + \|Y_t\|} \tag{2.6}$$

$$\mathcal{CE}(Y_p, Y_t) = \frac{1}{n} \sum_{y \in Y} \left(y_t \cdot \log y_p + (1 - y_t) \cdot \log \left(1 - y_p\right) \right) \tag{2.7}$$

where Y_p and Y_t are respective prediction and ground truth segmentation masks, respectively, and n is the total number of voxels. Y_t is defined similarly to Y_p in Eq. (2.4). Binary cross entropy is a common loss function for binary classification and we added Dice loss to regulate it for medical image segmentation. Namely, the Dice score helps counter the class imbalance that pixelwise classification problems face in medical image segmentation. This is critical when working with 3D images where the number of voxels belonging to a blood vessel is a

small percentage of the total voxels in the volume. It follows that our loss function is defined as

$$\mathcal{L} = \sum_{i}^{N_b} (1 - \mathcal{D}(Y_{p,i}, Y_{t,i}) - \mathcal{CE}(Y_{p,i}, Y_{t,i})) \tag{2.8}$$

for a batch size N_b, where each batch is a subset of the total dataset $N_b < N$ described in Eq. (2.3). The data is batched to fit into GPU memory as described in Table 2.2. Each image in the batch is processed in parallel on a GPU and the loss is accumulated before taking a gradient step to update the model parameters.

Surface, Centerline Calculations and Step Taking

As mentioned above, the output of the U-Net is a binarized image subvolume. The marching cubes algorithm [64] can be applied to this binarized image subvolume to generate a local surface mesh of the vessel segment. The resulting surface was cut using the image subvolume boundary planes, which results in truncation boundaries for the vessel(s), i.e., "inlets" or "outlets". One of these truncation boundaries is identified as the source (inlet) and others are identified as targets (outlets). This process was performed automatically using information from previous steps and from the direction of tracking. To do this, the centers of the truncation boundaries are calculated. The truncation boundary center closest to the previous stepping point is chosen as source and the rest as targets.

The surface mesh, with respective outlet labels, is used to automatically generate centerline(s) and radius estimates of the local vessel segment using a levelset based centerline extraction method. The method calculates centerline(s) as the path(s) that follow a wave propagation starting from a seed point [35]. The wave propagation is modeled by equation:

$$|\nabla T(x)| F(x) = 1 \tag{2.9}$$

where $T(x)$, the time it takes for wave to reach point x, is solved using a set "speed" function $F(x)$. $F(x)$ is set to have values proportional to distance from vessel boundary, leading to higher value towards the center and lower closer to vessel walls. When Eq. 2.9 is solved with $T(x_0) = 0$ at source point x_0, we obtain a solution with wave propagation faster in the center of vessels. Then, using that solution, we perform gradient descent starting from target point(s), where $T(x)$ is high, until we reach the source point, where $T(x)$ is low, and have therefore defined a centerline path(s). Since the "speed" function had higher values towards the center then so do the values of ∇T which forces the gradient descent towards the center of the vessel while tracing back, see [35] for details. Furthermore, we estimate the radius of the vessel at each point along the centerline by its distance to the surface.

The centerline extraction depends on well-defined outlet centers fed as seed points. Our method defines these outlet centers automatically, as described above. In the case of a bifurcation, a single outlet was labeled as the source based on the previous step and the direction of tracking. The point(s) to move to along the computed centerline(s) is chosen at

80% along each branch, see stepping point choice in Fig. 2.5. We found that this allowed for smaller step size, ensuring more accurate bifurcation detection.

Choice of Subvolume Size and Chances

The local centerlines calculated as described above contain 1D (lines) meshes in 3D space connecting all outlets with radius information along them, see Figure 2.5. This radius estimate is subsequently used to determine the size of the next subvolume to extract. The length of the next subvolume is chosen as five times the radius estimated, consistent with the size of the training samples as described in Section 2.3. Furthermore, to prevent underestimation of subvolume size, we let radius estimate carry on from one step to the next. The subvolume sidelength L is calculatated by an average of the current radius estimate, r_i, and the estimate from the previous step, r_{i-1}:

$$L = 5 * (r_i + r_{i-1})/2 \tag{2.10}$$

Additionally, we use the segmentation prediction itself as an indicator of subvolume size to vessel size ratio. If a high percentage of voxels within an image subvolume is predicted as belonging to vessel, that may indicate a small subvolume to vessel ratio, i.e., that the vessel occupies a large part of the volume. Utilizing this, we defined a cutoff percentage, γ^*, for which if the percentage exceeds it then we enlarge the subvolume size until it drops below, see Algorithm 1, where X is the subvolume, a function of sidelength L.

Algorithm 1 An algorithm to enlarge the subvolume based on the percentage of voxels predicted as vessel, γ.

$R \leftarrow (r_i + r_{i-1})/2$
$L_0 \leftarrow 5 * R$
$Y_p \leftarrow f(X(L_0))$
$\gamma \leftarrow \frac{1}{W*H*D} \sum_{y \in Y_p} y$
while $\gamma \geq \gamma^*$ **do**
  $L \leftarrow 5 * R * 1.1$ ▷ Increase sidelength by 10%
  $Y_p \leftarrow f(X(L))$
  $\gamma \leftarrow \frac{1}{W*H*D} \sum_{y \in Y_p} y$
  if $\frac{L}{L_0} > 1.3$ **then**
    break ▷ Maximum increase of 30%
  end if
end while

When SeqSeg encounters local subvolumes with image artifacts or unclear vessel boundaries, the neural network model sometimes produces inaccurate segmentations. However, in many cases, these inaccuracies are bound to those specific locations in the image volume whereas the following downstream vasculature may be clearer and easier to segment. To

handle these situations, we implemented a "chances" feature to SeqSeg. When SeqSeg fails to detect ≥ 2 outlets or fails to successfully compute a centerline, we give the step another chance and move further in the same direction and try again. Given a point $\mathbf{p}_i$ with a corresponding vessel tangent $\mathbf{t}_i$ and radius R_i, the next "chance" location $\mathbf{p}_{i+1}$ is calculated as:

$$\mathbf{p}_{i+1} = \mathbf{p}_i + R * \mathbf{t}_i \tag{2.11}$$

We set a maximum number of chances to three. This allows SeqSeg to better move past difficult regions of the image.

Bifurcations and Retracking Prevention

Bifurcations are detected by counting the branches of the centerline successfully computed. When bifurcations are detected, they are stored and returned to once other branches have been tracked. Namely, the largest radius outlet was chosen for continued tracking while the others were saved as bifurcation points and were revisited once the current vessel had been fully tracked. These bifurcation points were periodically sorted by radius to ensure prioritization of the largest vessels first, similar to how a human would interrogate the vasculature; see Figure 2.6.

Since the method detects outlets locally, it can sometimes detect the same bifurcations multiple times. This occurs especially if a small step size is used to advance the subvolume. To save computational time, we implemented a retracking prevention technique that periodically checks the global segmentation to determine whether the algorithm has segmented the current region before. We added buffers to the global assembly module to ensure that these checks only applied to segmentations involving past branches and not the current one.

Initialization

As mentioned above, the SegSeg method requires a seed point accompanied by a size estimate and direction for initialization. For evaluation purposes, this seed point is chosen at the 'start' of each vascular model, in the largest artery closest to the heart, similar to how a user would define it.

Stop Criteria

Since SeqSeg is an automated tracking method, stop criteria are required to terminate step-taking. The current framework has no explicit stop criteria; however, indirectly, it stops when either of two scenarios occur:

1. The method reaches the global image boundary, thereby requesting image data that does not exist.

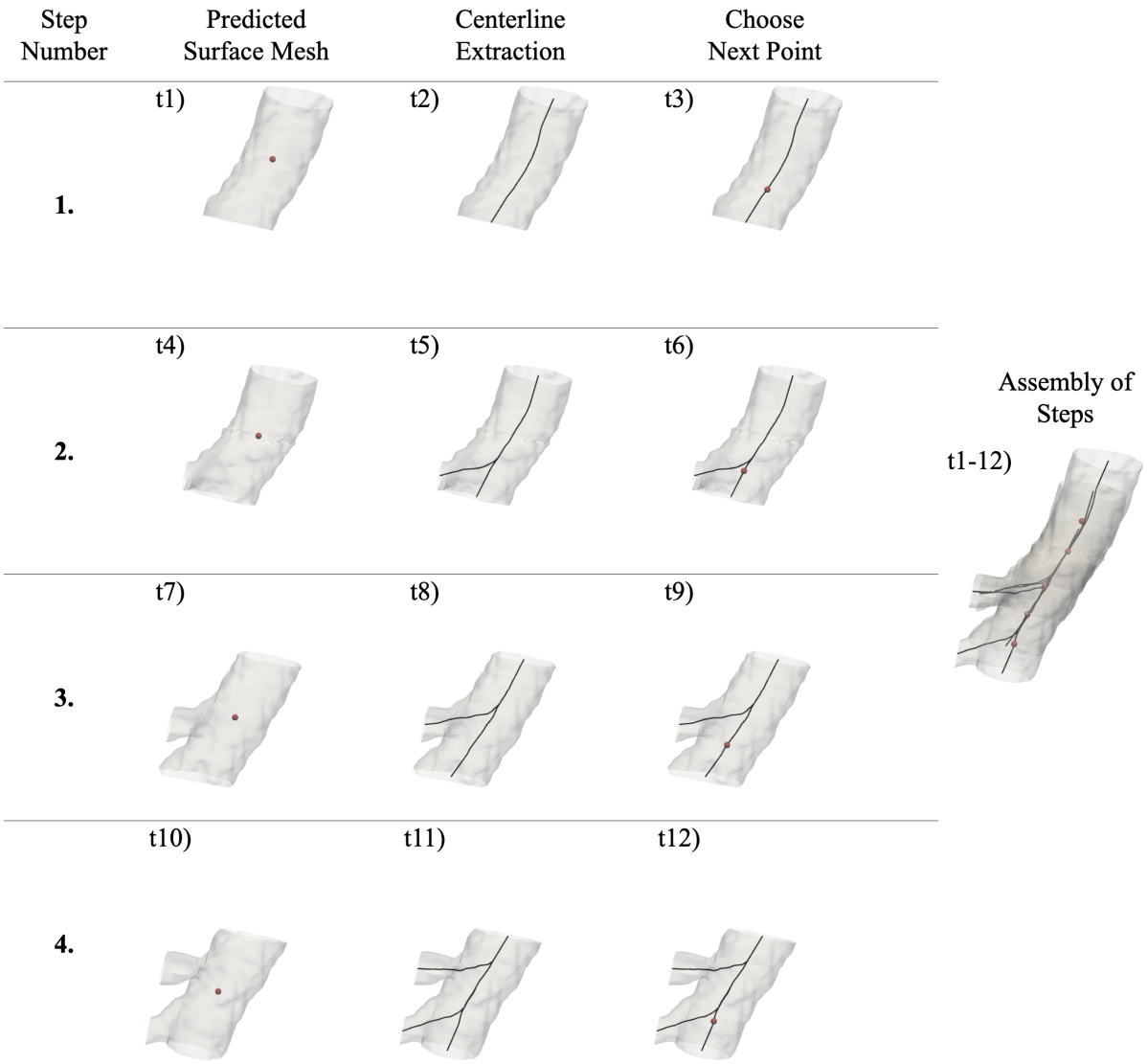


Figure 2.5: Automatic tracking using local surface mesh predictions for 3 steps, involving 12 calculation time steps. Centerlines are extracted and the next points are chosen to move to. These steps are subsequently assembled together to form the global vasculature model

2. The requested subvolume is of low resolution (e.g., resulting from tracking a small vessel) or is of too low quality (e.g., blurry or has image artifacts) that results in segmentation failure, or a centerline extraction failure from the resulting low-quality segmentation.

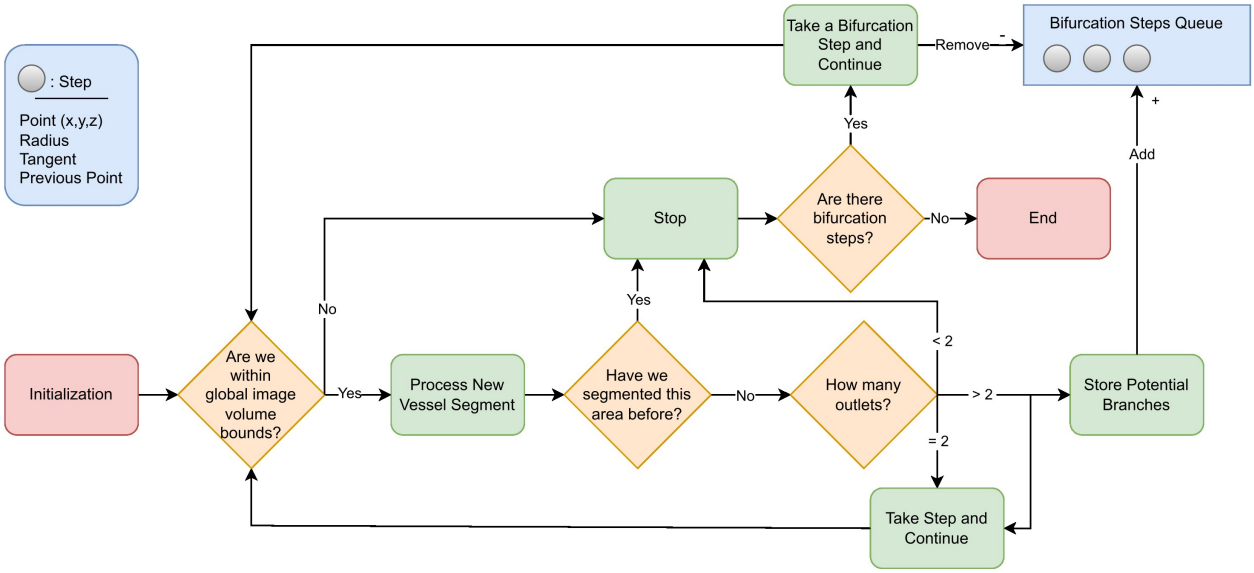


Figure 2.6: How the algorithm takes steps and handles bifurcations, starting from an initial seed point. The bifurcation points are stored in queue for subsequent tracking

In addition to the "forced" stop criteria mentioned above, we have implemented *optional* criteria as well, that can be toggled if premature stopping is desired. The optional stop criteria are as follows:

1. Define N_{max}, maximum number of steps taken and stop once number of steps taken has exceeded N_{max}. We set $N_{max} = 500$.

2. Define R_{min}, minimum allowed radius, and stop tracking down a branch once radius estimate is under R_{min}. We set $R_{min} = 0.5$mm.

3. Define NB_{max}, maximum number of branches to be tracked, and stop once number of branches exceed NB_{max}. NB_{max} can, for example, be set as 15.

Global Assembly

Sampling subvolumes may overlap a given voxel multiple times depending on the step size used to propagate the subvolume along the identified vessel (cf. superposition of subvolumes on the right side of Figure 2.5), or because of a new subvolume introduced to track a bifurcation. The end result is that several predictions may exist for a given voxel. Thus, all local segmentations are gathered globally by calculating a weighted mean prediction for each voxel. During development, we noticed that segmentations tend to be less accurate close

to the subvolume boundary, so we added Gaussian weighting giving more weight to voxel predictions closer to subvolume centers. This is also consistent with the benchmark nnU-Net method [42]. For each subvolume prediction of sidelength L and center point c, we define a weight map W, where each voxel with indices i, j, k and point location $p_{i,j,k}$ gets a weight $w_{i,j,k}$ according to:

$$w_{i,j,k} = \mathrm{e}^{-\frac{\|p_{i,j,k}-c\|_2}{2\sigma^2}}, \sigma = \frac{1}{4}L \tag{2.12}$$

Therefore, voxels close to c get a weight close to 1 while voxels close to the borders of the subvolume get a weight value of approximately two standard deviations from the mean, 0.02.

Given a set $\mathcal{S}$, of size $N_{\mathcal{S}}$, of segmentations in which local volumes included voxel $y_{i,j,k}$, the final global segmentation value was given by

$$y_{i,j,k} = \frac{1}{N_{\mathcal{S}} \sum_{s \in S} w^s_{i,j,k}} \sum_{s \in \mathcal{S}} w^s_{i,j,k} \cdot s_{i,j,k} \tag{2.13}$$

where i, j, k refer to global voxel indices and $w^s_{i,j,k}$ is the weight value for that voxel associated with segmentation s. This was performed prior to thresholding so that the resulting global segmentation retained voxel values ranging between $[0, 1]$ depending on the confidence. Finally, the segmentation was upsampled, and thresholded using a value of $t = 0.5$:

$$y_{i,j,k} = \begin{cases} 1 & \text{if} \quad y_{i,j,k} \geq t \\ 0 & \text{if} \quad y_{i,j,k} < t \end{cases} \tag{2.14}$$

where the largest connected body is retained, converted to a surface mesh using marching cubes and smoothed to remove voxel artifacts. For mesh smoothing, we use a windowed sinc function interpolation kernel to move mesh vertices[115]. More specifically, we perform ten iterations with a passband value of 0.01.

Experiments, Metrics and Statistical Analysis

We compared our results to those of two benchmark nnU-Net models, i.e. a 2D nnU-Net and a 3D nnU-Net, trained on the global image volumes. The 3D nnU-Net performs 3D convolutions whereas the 2D version performs 2D convolutions, and outputs 2D predictions, solely on the image z-plane, the patient's axial plane. A 3D segmentation map output from the 2D nnU-Net is assembled by a z-stack of 2D segmentations. The nnU-Net is arguably the most state-of-the-art method for medical image segmentation and thus is chosen as a benchmark for comparison.

The metrics for comparing SeqSeg with the global nnU-Net benchmarks were as follows:

$$\mathcal{D}(X, Y) == \frac{2 \cdot \|X \cap Y\|}{\|X\| + \|Y\|} \tag{2.15}$$

$$\begin{aligned} \mathcal{H}(X, Y) &= \max\{d(X, Y), d(Y, X)\}, \\ \text{where} \quad d(X, Y) &= \sup_{x \in X} \inf_{y \in Y} d(x, y) \end{aligned} \tag{2.16}$$

$$\mathcal{CO}(Y, C_t) = \frac{\int_{C_t} Y dx}{\int_{C_t} dx} \tag{2.17}$$

where $\mathcal{D}$ is the Dice score, $\mathcal{H}$ is the Hausdorff distance, and $\mathcal{CO}$ is the centerline overlap with C_t being the ground truth centerline and X,Y being segmentation maps. Here X represents the ground truth segmentation map and Y the predicted segmentation maps, either output from SeqSeg or the benchmark nnU-Net models. The Dice score measures the overlap between two segmentations and ranges between $[0, 1]$. The Dice score is common for medical image segmentation because of its ability to penalize imbalanced datasets accurately. The Hausdorff distance measures the maximum distance between two surfaces and has a minimum of 0 for identical surfaces. The centerline overlap is a score ranging between $[0, 1]$ and represents the percentage of the ground truth centerline captured in the predicted segmentation. Both the $\mathcal{H}$ and $\mathcal{CO}$ metrics give additional insight into a method's ability to capture bifurcations and specifically small blood vessels compared to the Dice score. The Dice score compares segmentations on a volumetric basis by comparing voxels, but since most voxels belong to larger blood vessels, it results in indirect bias.

Not all blood vessels were annotated in the test datasets, or they were present but truncated. We thus masked the outputs from all methods with the volume surrounding the ground truth annotated vessels. We define this mask volume by labelling all pixels within a six radius distance from the ground truth centerline.

For statistical analysis, we perform the Wilcoxon signed-rank test between the resulting metrics scores of SeqSeg and the benchmark. This is a non-parametric test similar to the paired t-test. But since the paired t-test has limitations when comparing machine learning model performances, we opted for the Wilcoxon test [99]. Specifically, we test the null hypothesis that the median of differences between the two sets of sample results (metric scores) is zero. A p-value less than 0.05 was considered to reject the null hypothesis, and therefore indicate a statistically significant difference between the two sets. We both perform Wilcoxon tests between SeqSeg and 2D nnU-Net predictions and between SeqSeg and 3D nnU-Net predictions.

2.4 Results

We tested SeqSeg on 15 held-out VMR image volumes (test set), 8 of which were CT volumes and 7 of which were MR volumes, as well as additional 18 AVT CT image volumes. Namely, SeqSeg and the 2D and 3D nnU-Net benchmark methods were used to segment the vasculature from these image volumes, and those resulting segmentations were compared to manually-generated "ground-truth" segmentations from the open data Vascular Model Repository and the AVT dataset[65].

A quantitative evaluation of the Dice score ($\mathcal{D}$), Hausdorff distance ($\mathcal{H}$) and centerline overlap ($\mathcal{CO}$) for segmentations generated from the VMR test set using SeqSeg and the nnU-Net benchmark methods is presented in Table 2.3. SeqSeg, on average, obtained higher Dice scores than the nnU-Net benchmarks in 11 test cases, lower Hausdorff distance scores

in 11 cases, and higher centerline overlap scores in 12 cases, all out of a total of 15 test cases. Specifically, SeqSeg on average obtained higher scores than the 2D and 3D nnU-Net benchmarks in terms of Dice score by 0.017 and 0.029, respectively for CT, and 0.036 and 0.029, respectively for MR. For the Hausdorff distance, SeqSeg on average obtained lower distance than the 2D and 3D nnU-Net benchmarks by 0.59 and 0.966 pixels, respectively for CT, and 0.565 and 0.872, respectively for MR data. For centerline overlap, SeqSeg obtained higher scores on average than the 2D and 3D nnU-Net benchmarks by capturing 3% and 10.9% more vessel segments, respectively for CT, and 9.4% and 10.4% more segments, respectively for MR. Improvements to metric averages that were statistically significant ($p < 0.05$) are indicated by an asterisk * in Table 2.3. Beyond mean improvements, SeqSeg appeared far more robust. This can be observed from the the box plots in Figure 2.9, which demonstrates greater consistency in the performance of SeqSeg for all quantitative metrics.

Since the objective of segmentation is a unified, high-quality geometric model, distilling the comparison down to any single metric is overly simplistic (if not deceptive). For broader perspective, Figure 2.7 provides a qualitative comparison, showing the CT and MR segmentations for which the 2D nnU-Net benchmark yielded the best, median and worst Dice scores, and includes comparison of these to the segmentations generated from SeqSeg. (Note, since the 2D nnU-Net was superior to the 3D nnU-Net, the 2D nnU-Net was considered the de facto benchmark.) This figure demonstrates that SeqSeg generally captures more of the vasculature, and particularly the connections to smaller branch arteries. The Appendix provides a visual comparison of the segmentations generated by SeqSeg and the nnU-Net benchmarks for all CT, and MR, VMR test cases in Figures 2.10, and 2.11, respectively.

For the AVT CT test set, the quantitative metric scores can be seen in Table 2.5 between SeqSeg and the 2D nnU-Net benchmark. On average, SeqSeg obtained higher Dice scores by 0.065 and centerline overlap by 10.8%. On average, the benchmark Hausdorff distance scores were lower by 0.401 pixels. In terms of statistical significance, both Dice and centerline overlap differences were found statistically significant whereas the difference in Hausdorff distance was not. Boxplots of the metric scores are shown in Figure 2.8, again showing a smaller spread (better robustness) for SeqSeg compared to the benchmark, particularly for Dice score and centerline overlap. For qualitative comparison, all resulting meshes are shown in Figure 2.12. As observed, SeqSeg captures more, and smaller, branches as compared to the benchmark, even branches not included in the ground truth (cf. cases 2, 4, 5, 6, 8, 9, 11, 15, 16, 17 and 18).

Inference time also differed between SeqSeg and the benchmarks. If both are run on the same CPU, measured SeqSeg inference time ranged 20-80min, depending on the number of branches, whereas the nnU-Net benchmarks ranged 2-3hr.

Comparison with nnU-Net's largest connected region

The ultimate goal in image-based modeling is to use a segmentation as the computational domain for numerical simulation. Simulations require domains to be unified and well defined. Since the nnU-Net segmentations are often disjoint, filtering and keeping only the largest

connected body would be necessary to use the segmentation for simulation purposes. Thus, for a more practical comparison, in this subsection we compare SeqSeg segmentations with nnU-Net segmentations that have been filtered to retain the largest connected region.

We present the quantitative metric values for the largest connected region results in Table 2.4. From this table we can observe that SeqSeg on average obtained higher metric scores than the 2D/3D nnU-Net benchmark, respectively, as follows: the Dice coefficient improved by 0.062/0.032 for CT and 0.064/0.029 for MR; the Hausdorff distance improved by 1.812/2.002 for CT and 2.153/0.839 pixels for MR; and the global centerline overlap increased by 10.2/16.8% for CT and 18.7/13.8% for MR. Improvements to metric averages that were statistically significant ($p < 0.05$) are indicated by an asterisk * in Table 2.4. Differences in centerline overlap scores were found statistically significant between SeqSeg and both benchmark methods. The box plots of these metrics for all cases are shown in Figure 2.9 and again reveal far less spread in the metrics for SeqSeg compared to both benchmark models, indicating greater robustness in segmentation results for SeqSeg.

Figure 2.7 displays segmentation results for the best, median and worst case results for the 2D nnU-Net benchmark largest connected region, and compares to the segmentation predicted by SeqSeg. (Again, we excluded the 3D nnU-Net in this comparison as it generally performed worse than its 2D counterpart.) For all cases shown, SeqSeg generally captures a greater number of branches and greater extent of the vessels, even when compared to nnU-Net's best results. This is further demonstrated in Figures 2.10 and 2.11 in the Appendix for all test cases.

For the AVT CT test data, Table 2.5 and Figure 2.12 show results for the benchmarks after largest connected component filtering, quantitatively and qualitatively respectfully. We obtain statistically significant difference between SeqSeg and the benchmark for all metrics; Dice, Hausdorff distance and centerline overlap. As shown in Fig. 2.12, SeqSeg produces better unified vascular trees in more instances than the benchmark.

2.5 Discussion

U-Net learning models, and particularly the more recent nnU-Net, have shown excellent potential for automating image segmentation tasks. However, segmentation of branched vascular structures from medical image data is fraught with challenges since vessels typically compose relatively few pixels, vascular geometry varies considerably between patient and location, and maintaining connectivity of highly branched structures by pixel classification is tricky. We herein propose a sequential segmentation technique (SeqSeg) that leverages U-Net learning to locally build vascular models. We observed that SeqSeg generally outperformed current state-of-the-art global nnU-Net models when tested on typical vascular images used for image-based modeling, particularly when comparing overall extent of connected vasculature predicted. Namely, SeqSeg was superior in extending the segmentations into smaller branch arteries or distal segments when compared to the global nnU-Net benchmarks (or, in

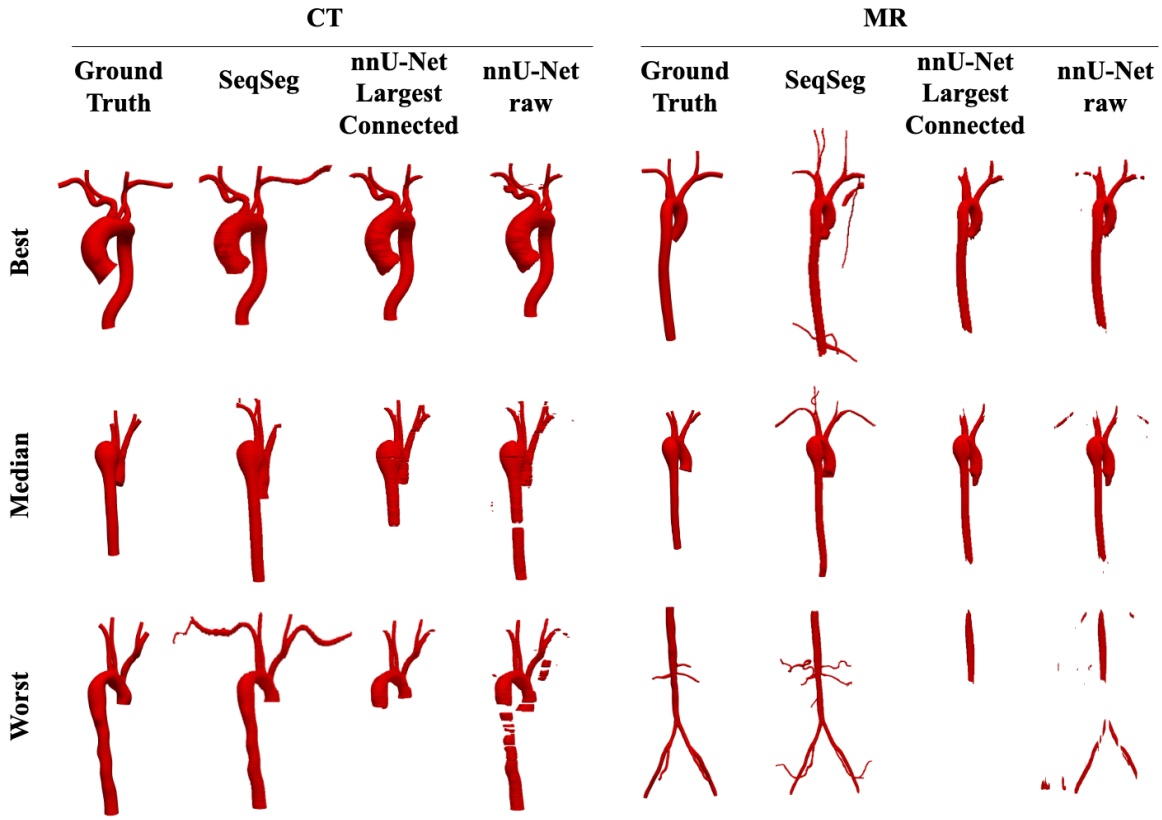


Figure 2.7: Qualitative comparison of the resulting meshes on the VRM test dataset, comparing the best, median and worst cases of the nnU-Net benchmark to those of SeqSeg. From Table 2.3 these are cases 7, 4 and 5 for CT and 4, 5, and 1 for MR data, respectively

fact, "ground truth" segmentations). In addition, SeqSeg performance was generally more robust, as indicated by less variance in the quantitative results.

We note that when comparing the difference in the quantitative metrics across the VMR test cases, the superiority of SeqSeg did not necessarily reach statistical significance as measured by $p < 0.05$. This is perhaps due to the smaller number of test cases we had access to. In addition, the VMR ground truth used for evaluating these metrics were not as extensive as they could have been, which likely handicapped the comparison since SeqSeg generally excelled at extending the segmentations further down the vascular tree compared to the nnU-Net benchmarks. Namely, the ground truth data tended to be limited to the aorta and proximal segments of branch arteries, which are generally easier to segment, and both SeqSeg

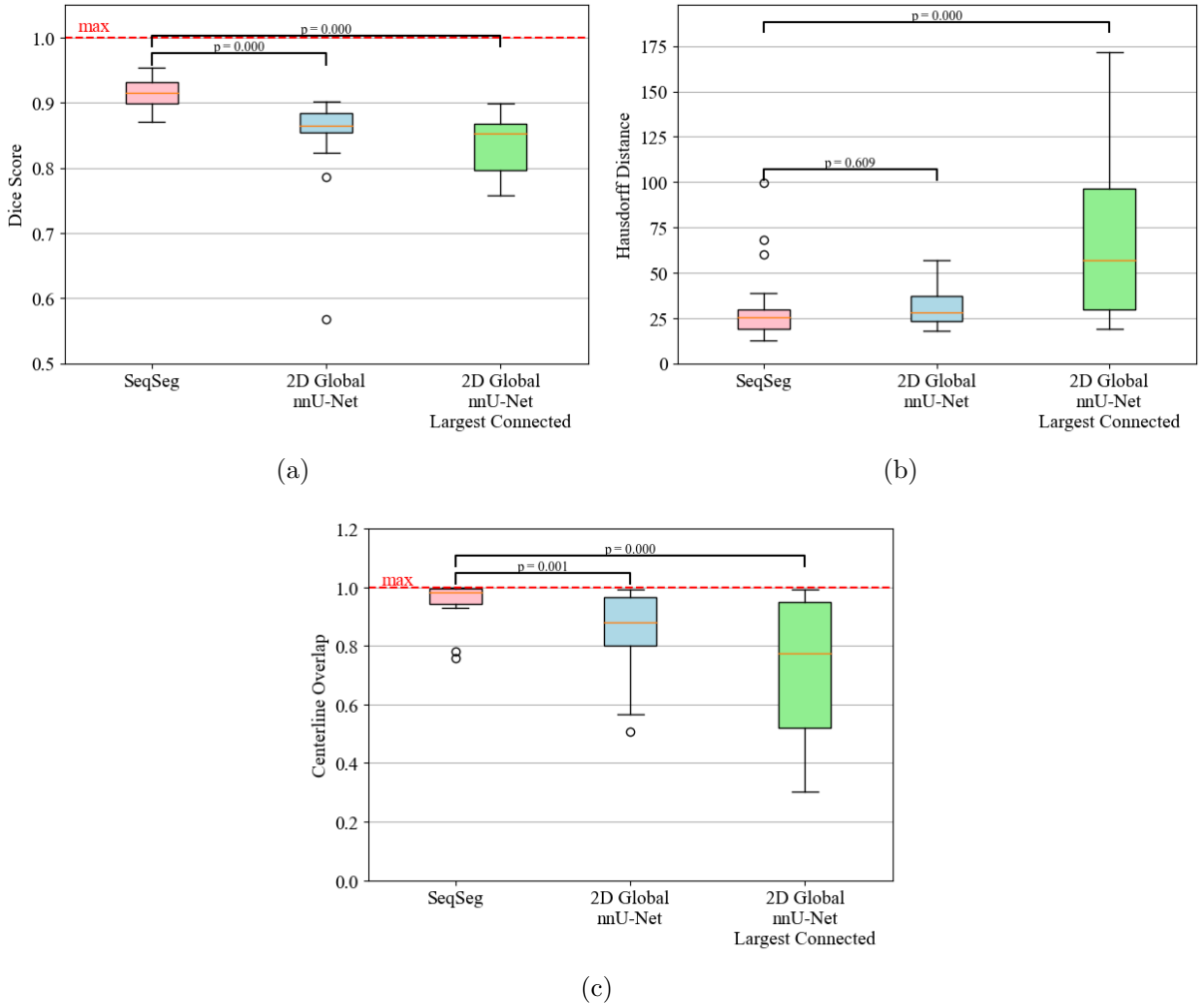


Figure 2.8: Quantitative metric scores for the AVT test dataset for SeqSeg and the 2D nnU-Net benchmark, both raw and after largest connected component filtering. (a) Dice (b) Hausdorff distance (in pixels) (c) Centerline Overlap. See Sec. 2.3 for definition of p-values.

and the nnU-Net benchmarks performed *on average* equally well in such "less-challenging" regions. Moreover, metrics like Dice are inherently biased to larger vessels. While the ground truth segmentations could have been manually altered to extend vessels, or include missing branches, this can introduce potential bias; thus, we chose not to adulterate the ground truth from the public repository. Lastly, we note that most prior publications in this field [90, 14, 24, 77, 15, 37] do not report whether their improvements to prior benchmarks were statistically significant.

The application of SegSeg to the AVT CT test dataset offered an interesting application.

Whereas SegSeg and the benchmarks were trained on VRM data, with a subset of the VMR data held out for testing, the AVT data was a completely *de novo* data source unrelated to the training. For the application to this *de novo* data, SegSeg more convincing outperformed the nnU-Net benchmarks, achieving statistically significant higher accuracy in terms of *all* quantitative metrics. Moreover, the qualitative comparisons shown in Fig. 2.12 demonstrate that SegSeq was able to segment far more aortic branch arteries, and extend arteries further distally than the benchmark, or, in fact, even than the ground truth. It is unknown if this superior performance extends to similar *de novo* MR data since such data was not available.

The advantage of SeqSeg is that it focuses the segmentation task locally around a vascular segment. Indeed, the same neural network architecture and training strategies were used for SeqSeg and the benchmarks. The centerline overlap metric, which give increased weight to smaller branches and bifurcations, showed generally better performance for SeqSeg. In fact, after largest connected region filtering, the difference in centerline overlap metric was statistically significant for all test datasets. This is likely because the nnU-Net benchmarks may sacrifice smaller branches in order to accurately predict larger branches that carry more pixels. In contrast, SeqSeg deals with all branches, if detected, more equitably, by focusing on one segment at a time. Figures 2.10, 2.11, and 2.12 show another major advantage of assembling a vascular network piece-wise: ensuring connectivity of the resulting model, which is crucial for blood flow and tissue mechanics simulation purposes.

Since the overall objective is to produce models capable of physics based simulation, it is notable to mention SeqSegs superiority towards that goal. Firstly, as mentioned above, SeqSeg surpasses the benchmark's' ability to generate expansive *and* single connected body models. Secondly, since SeqSeg tracks the vasculature, it maintains information on branches and their connections relative to the global vascular organization. This can be used to place boundary conditions (inlet and outlet conditions), necessary for physics-based simulation setup. This information is not available for typical CNN segmentation methods since all pixels are treated equally and vasculature organization is ignored. The authors note that this study does not directly look at the effects of these methods on actual physics-based simulation results, which is beyond the scope of the current study.

One might assume an advantage of SegSeq is that because it uses local patches of the image volume, the number of inputs for training is higher than for the global nnUnets. However, during training nnUnet partitions the image volume into patches and uses extensive augmentations, which greatly increases the "samples" for training.

The results from the benchmark 2D and 3D nnU-Nets show the limitations of 3D convolutional neural networks for global vasculature segmentation–the problems of class-imbalance and image size. Because global image volumes surpass GPU memory, methods are forced to either downsample or split the image into patches to fit on a GPU. Our method excels within the constraints of GPU memory because it processes smaller subvolumes at each time, which rarely exceeds the GPU memory limit, see Table 2.2 for larger possible batch sizes for example. Furthermore, in a global image volume, the vascular pixels represent only a fraction of the total pixels, making training difficult. Our method focuses on the pixels around the vasculature, which, by definition, alleviates class imbalance.

On the other hand, the results also indicate that the source of better segmentation is not simply locality. The benchmark models were trained on small patches that do not undergo downsampling. By training locally *and* incorporating prior learned information, i.e. the location and size of the vessel, SeqSeg is generally able to segment with greater detail and accuracy, particularly in smaller vessels.

Another limitation that impacts global segmentation learning is that ground truth segmentation, being human-generated, in most cases did not contain segmentation of all branches or portions. This implies that some training data had certain arteries, e.g., the renal arteries, segmented while others did not, which can result in poor segmentation of test data. Since SeqSeg can utilize training patches around vessels, the training mostly encounters positive examples of arteries, e.g. the renal arteries only if they are present, and will not encounter negative (wrong) examples from less segmented images, e.g. where the renal arteries were not segmented. Thus, SeqSeg can be more efficient with training data, which is beneficial since annotated data collection is costly and time-consuming. This could also help explain the ability of SeqSeg to segment a greater number of smaller branches, even those not present in all training examples.

Additionally, SeqSeg may have been able to generalize to regions not present in the training data because vessels share similar image features when viewed locally. For example, Figures 2.10 and 2.11 show how SeqSeg managed to track and segment small bifurcations not included in the ground truth as well as elongate already segmented vessels. Inspection confirmed that these vessels were present in the image data. In fact, the authors further confirmed this qualitatively by training a model solely on *one* branch per case (the aorta), and SeqSeg was able to generalize to other branches on test data. Furthermore, SeqSeg manages to detect and handle bifurcations, which has been an challenge for blood vessel tracking and tracing methods [46, 127, 1, 97, 57]. Unlike other works, SeqSeg does not depend on explicit bifurcation detection, but instead handles them implicitly through surface representations expressing them. This makes handling complex junctions with multiple branches more tractable.

For further comparison to previous works, SegSeg achieved better global Dice scores than Maher et al. who trained neural networks for 2D lumen segmentation on similar datasets [68]. Furthermore, the method of Maher et al. depended on previously user-generated centerlines, whereas our method automatically generates the centerlines while simultaneously segmenting the vasculature. This is significant since centerline generation is often the most time-consuming step of image-based model construction.

SegSeg used a 3D U-Net neural network architecture for local segmentation predictions, however, other architectures, such as vision transformers [37], transfer learning models such as 3D MedNet [15] or V-Net [77], with residual connections, could possibly be used to perform this task. Similarly, future developments could include learning methods to determine step size or other parameters that are derived from deterministic procedures in our presented model. For example, deep learning can be applied to directly predict subsequent points [23], local centerline segments or surfaces using template-based approaches similar to what has been done for cardiac model construction [52]. Additionally, the SeqSeg method can be

trained and tested for generalization to other vascular anatomies such as coronary arteries, pulmonary arteries and cerebrovascular models. Since the training and testing occur locally, new data from different anatomies can be readily incorporated into the existing framework.

Limitations of the presented method include the dependence on voxel-based segmentation, the dependence on capturing bifurcation roots, and the possible computational cost. Voxel-based segmentation inevitably leads to staircase artifacts on the final surface, as shown in Figures 2.10 and 2.11. Since our stepwise approach relies on accurately capturing bifurcation roots, there is the possibility of missing whole branches if the root is difficult to segment, e.g. because of image artifacts. The method also requires neural network inference at each step, which has the potential to scale poorly for extensive vascular networks.

2.6 Conclusion

Despite its growing importance, reconstructing vascular models from medical image data in an accurate and rapid manner remains an open area of research. In this work we present SeqSeg; a novel image-based vascular model construction method based on building the vascular network in a stepwise manner to facilitate learning. SeqSeg is capable of automatically tracking and assembling a global segmentation and surface, depending only on a single seed point. We tested the method on CT and MR images of aortic and aortofemoral models and compared to state-of-the-art benchmark 2D and 3D U-Net segmentation methods, SegSeg had similar or better accuracy in terms of Dice score, Hausdorff distance, and centerline overlap, but more notably was more robust and able to connect a greater extent of the vasculature. Our future work includes training and testing using other vascular anatomies as well as further optimizing local segmentation and bifurcation detection.

2.7 Appendix

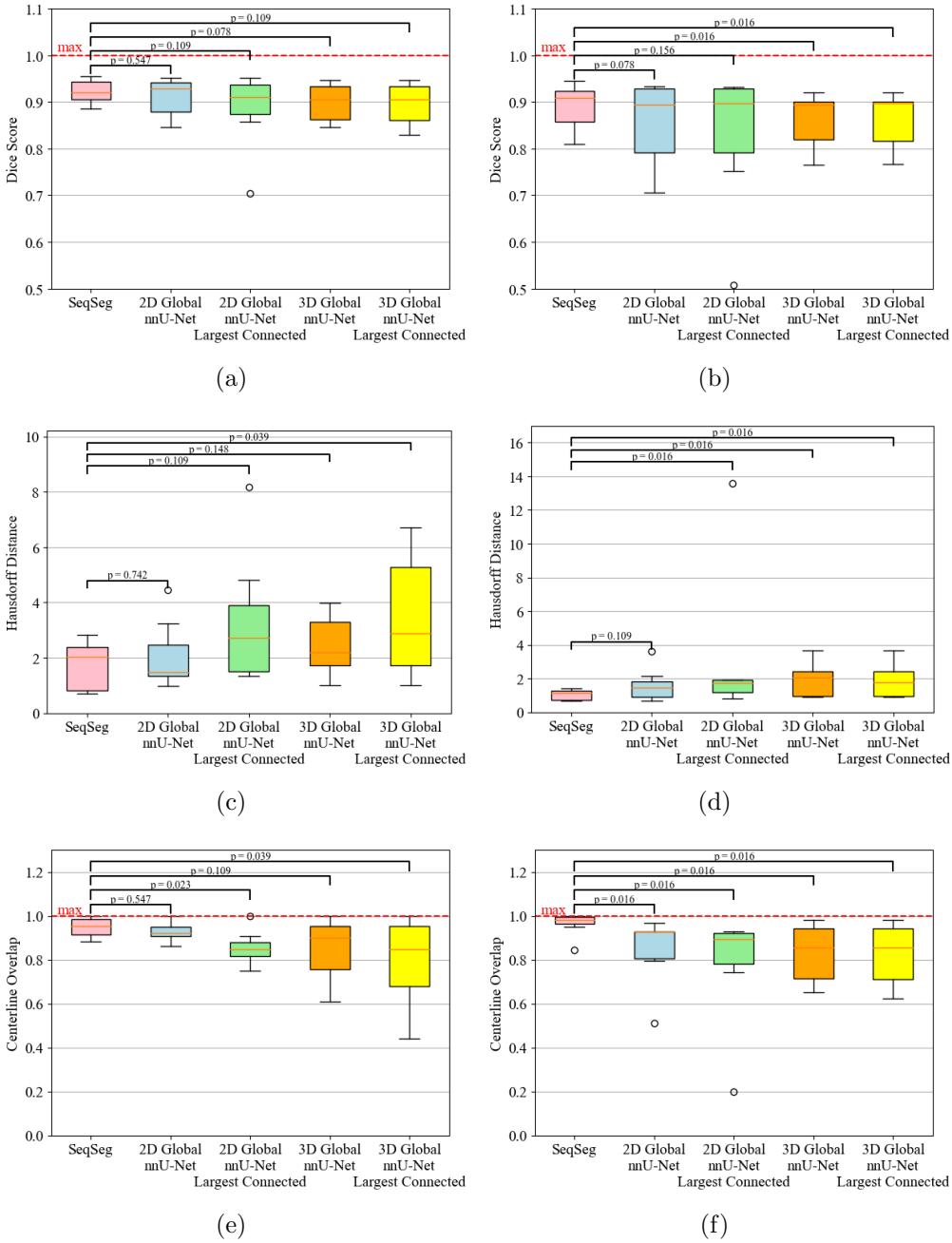


Figure 2.9: Quantitative metrics for VMR test data; (a) CT Dice (b) MR Dice (c) CT Hausdorff (in pixels) (d) MR Hausdorff (in pixels) (e) CT Centerline Overlap (f) MR Centerline Overlap. See Sec. 2.3 for definition of p-values.

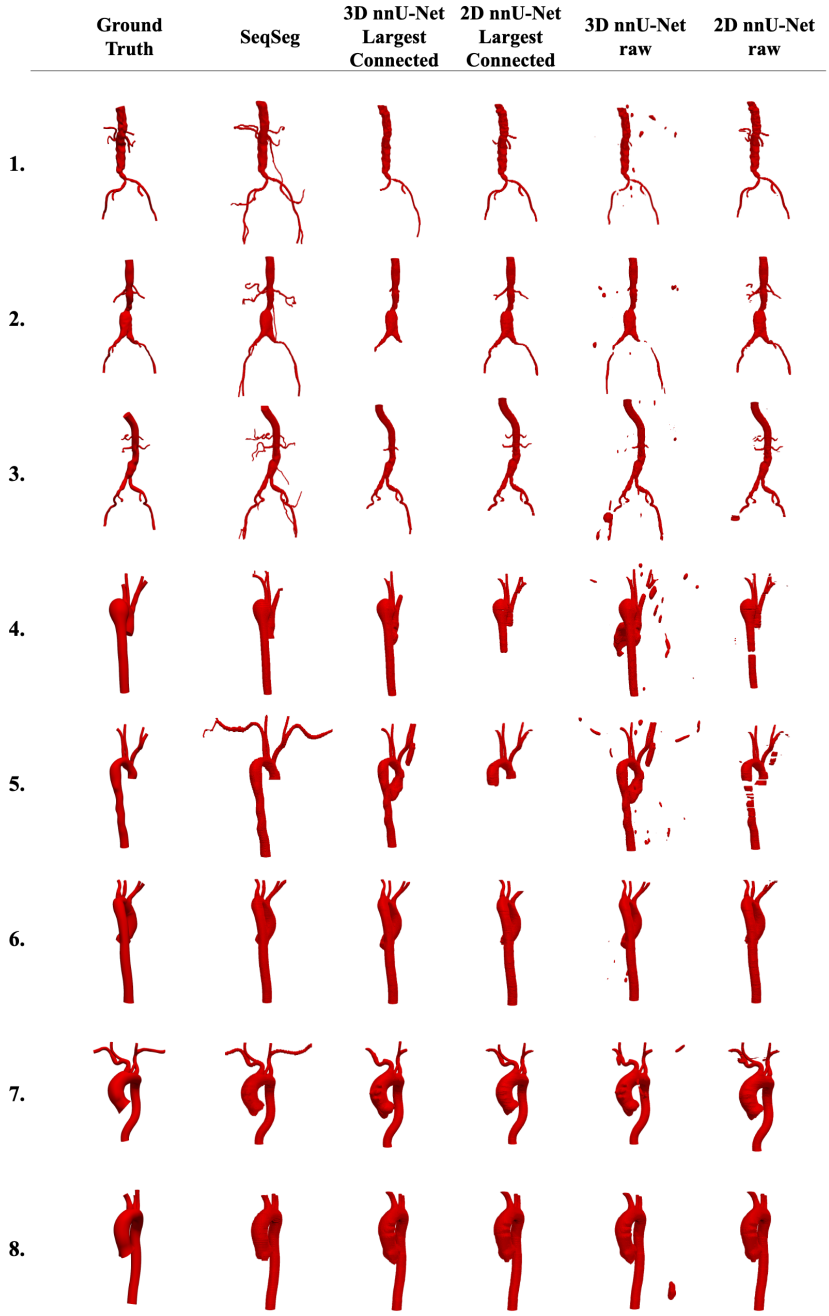


Figure 2.10: Resulting meshes from complete VMR CT test dataset. Each row represents a different vascular model, labeled consistently with Tables 2.3, 2.4

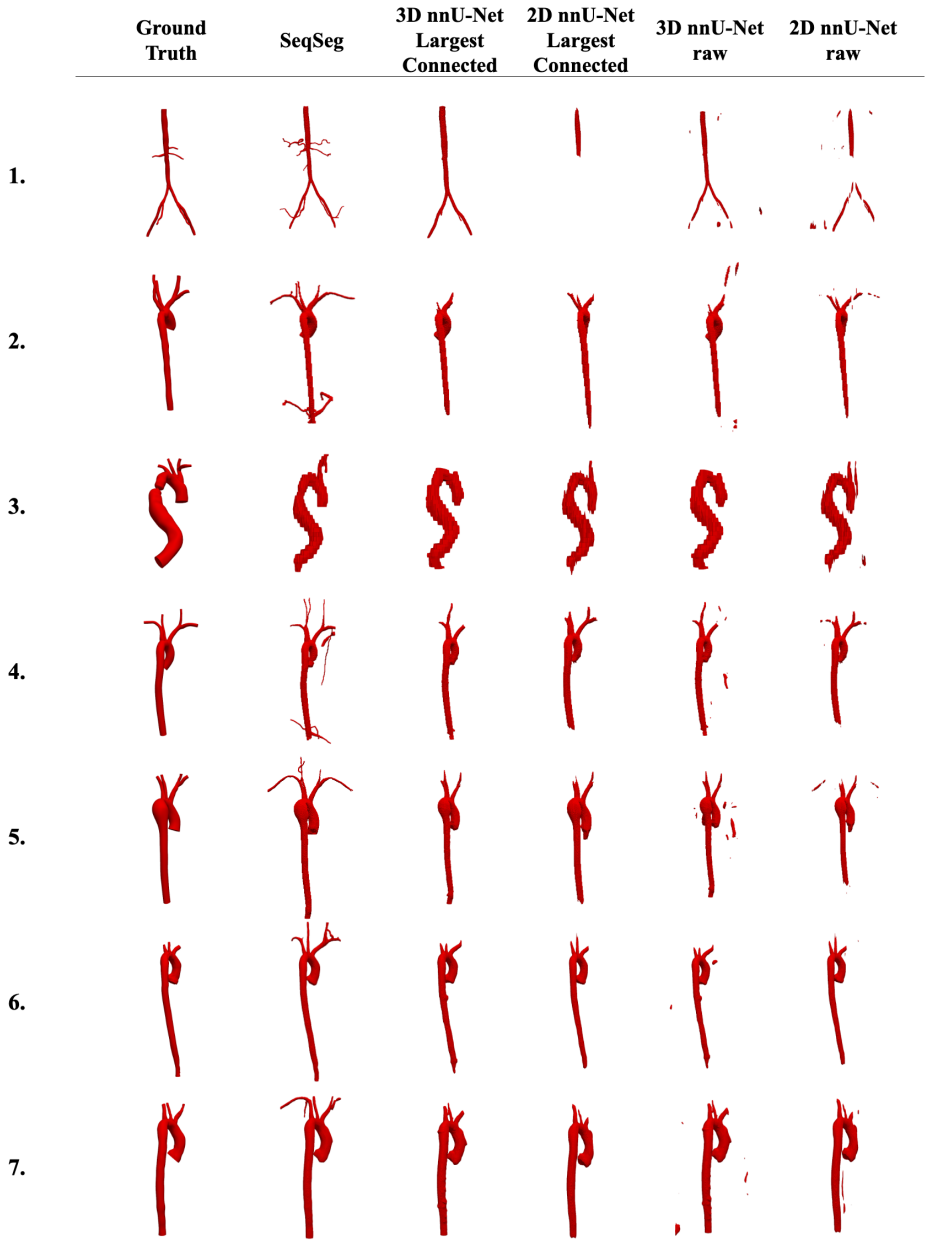


Figure 2.11: Resulting meshes from complete VMR MR test dataset. Each row represents a different vascular model, labeled consistently with Tables 2.3, 2.4

Table 2.2: The U-Net architecture and training specifications, for both the SeqSeg models and global benchmark models

Parameter	SegSeg CT	SeqSeg MR	3D CT	3D MR	2D CT	2D MR
Intensity Normalization	0.5/99.5% clip + foreground z-score	all image z-score	0.5/99.5% clip + foreground z-score	all image z-score	0.5/99.5% clip + foreground z-score	all image z-score
Image Target Spacing	0.200, 0.0547, 0.0547	0.0859, 0.0625, 0.0625	0.0800, 0.0488, 0.0488	0.0859, 0.0586, 0.0586	0.0488, 0.0488	0.0586, 0.0586
Patch Size	[20,80,80]	[40,48,48]	[96,160,160]	[56, 256, 160]	[512, 512]	[512,384]
Batch Size	33	57	2	2	12	16
Max Nr. Features	320	320	320	320	512	512
Nr. Stages Encoder	5	4	6	6	8	7
Nr. Stages Decoder	4	3	5	5	7	6
Nr. Layers per Stage	2	2	2	2	2	2
Nr. Pooling Ops. per Axis	[2,4,4]	[3,3,3]	[3,5,5]	[4,5,5]	[7,7]	[6,6]
Conv. Kernel Size	[3,3,3]	[3,3,3]	[3,3,3]	[3,3,3]	[3,3]	[3,3]

Table 2.3: Quantitative comparisons for the VRM test dataset between the two benchmark U-Net segmentation methods (2D, 3D) and SeqSeg using the Dice score ($\mathcal{D}$), Hausdorff distance ($\mathcal{H}$) and centerline overlap ($\mathcal{CO}$). The case types were either aortofemoral (AF) or aortic (AO), and the number of branches segmented is also shown (Nr. Br.). * indicates statistically significant difference ($p < 0.05$)

Mod.	Case	Type	Nr. Br.	$\mathcal{D}\uparrow$			$\mathcal{H}\downarrow$			$\mathcal{CO}\uparrow$		
				Seq-Seg	2D U-Net	3D U-Net	Seq-Seg	2D U-Net	3D U-Net	Seq-Seg	2D U-Net	3D U-Net
CT	1	AF	9	**0.907**	0.885	0.846	**1.930**	2.208	3.526	**0.939**	0.906	0.657
	2	AF	10	0.931	**0.941**	0.909	1.951	**1.406**	2.442	0.884	**0.928**	0.611
	3	AF	10	**0.885**	0.860	0.855	**2.339**	4.452	3.973	**0.959**	0.864	0.791
	4	AO	5	0.902	**0.923**	0.901	2.522	**1.281**	1.976	0.919	**0.951**	0.939
	5	AO	5	**0.940**	0.845	0.865	**0.717**	1.544	1.804	**1.000**	0.916	0.992
	6	AO	6	**0.951**	0.947	0.946	**0.867**	0.991	0.999	**0.980**	0.951	0.942
	7	AO	5	**0.955**	0.951	0.938	**0.725**	3.237	3.216	**0.994**	0.911	0.864
	8	AO	4	**0.954**	0.934	0.932	**0.708**	1.358	1.473	0.990	**1.000**	**1.000**
	Avg.	-	-	**0.928**	0.911	0.899	**1.470**	2.060	2.426	**0.958**	0.928	0.849
	p-value	-	-		0.547	0.078		0.742	0.148		0.547	0.109
MR	1	AF	9	**0.877**	0.706	0.816	**1.429**	3.613	3.652	**0.977**	0.510	0.650
	2	AO	5	**0.810**	0.759	0.766	**1.154**	1.515	2.337	**0.950**	0.796	0.719
	3	AO	5	**0.836**	0.824	0.822	**0.657**	0.807	0.924	**0.844**	0.816	0.711
	4	AO	5	0.923	**0.933**	0.898	1.132	**0.975**	2.487	**0.981**	0.929	0.855
	5	AO	5	**0.909**	0.894	0.895	**1.371**	2.141	2.048	**1.000**	0.930	0.960
	6	AO	5	0.923	**0.932**	0.920	0.704	**0.653**	0.986	**0.990**	0.966	0.927
	7	AO	4	**0.945**	0.925	0.904	**0.780**	1.474	0.895	**1.000**	0.929	0.983
	Avg.	-	-	**0.889**	0.853	0.860	**1.032**	1.597	1.904	**0.933**	0.839	0.829
	p-value	-	-		0.078	0.016*		0.109	0.016*		0.016*	0.016*

Table 2.4: Quantitative comparison for the VMR test dataset *after* largest connected body filtering between the two benchmark U-Net segmentation methods (2D, 3D) and our method, SeqSeg, using the Dice score ($\mathcal{D}$), Hausdorff distance ($\mathcal{H}$) and centerline overlap ($\mathcal{CO}$). The case types were either aortofemoral (AF) or aortic (AO), and the number of branches segmented is also shown (Nr. Br.). * indicates statistically significant difference ($p < 0.05$)

Mod.	Case	Type	Nr. Br.	$\mathcal{D}\uparrow$			$\mathcal{H}\downarrow$			$\mathcal{CO}\uparrow$		
				Seq-Seg	2D U-Net	3D U-Net	Seq-Seg	2D U-Net	3D U-Net	Seq-Seg	2D U-Net	3D U-Net
CT	1	AF	9	**0.907**	0.879	0.830	**1.930**	2.989	6.723	**0.939**	0.824	0.521
	2	AF	10	0.931	**0.932**	0.893	1.951	**1.544**	5.966	**0.884**	0.800	0.439
	3	AF	10	**0.885**	0.858	0.846	**2.339**	4.810	5.060	**0.959**	0.831	0.733
	4	AO	5	**0.902**	0.887	0.916	**2.522**	3.596	2.878	**0.919**	0.869	0.865
	5	AO	5	**0.940**	0.704	0.865	**0.717**	8.163	1.804	**1.000**	0.750	0.992
	6	AO	6	**0.951**	0.945	0.946	**0.867**	1.335	0.999	**0.980**	0.865	0.942
	7	AO	5	0.955	**0.952**	0.939	**0.725**	2.464	2.875	**0.994**	0.907	0.831
	8	AO	4	**0.954**	0.934	0.932	**0.708**	1.358	1.473	0.990	**1.000**	**1.000**
	Avg.	-	-	**0.928**	0.886	0.896	**1.470**	3.282	3.472	**0.958**	0.856	0.790
	p-value	-	-		0.109	0.109		0.109	0.039*		0.023*	0.039*
MR	1	AF	9	**0.877**	0.508	0.810	**1.429**	13.575	3.652	**0.977**	0.199	0.623
	2	AO	5	**0.810**	0.752	0.766	**1.154**	1.874	2.337	**0.950**	0.744	0.713
	3	AO	5	**0.836**	0.832	0.822	**0.657**	0.807	0.924	**0.844**	0.816	0.711
	4	AO	5	0.923	**0.931**	0.898	**1.132**	1.743	2.524	**0.981**	0.894	0.855
	5	AO	5	**0.909**	0.897	0.898	**1.371**	1.925	1.778	**1.000**	0.930	0.960
	6	AO	5	0.923	**0.930**	0.920	**0.704**	0.831	0.986	**0.990**	0.916	0.927
	7	AO	4	**0.945**	0.928	0.904	**0.780**	1.540	0.895	**1.000**	0.929	0.983
	Avg.	-	-	**0.889**	0.825	0.860	**1.032**	3.185	1.871	**0.963**	0.776	0.825
	p-value	-	-		0.156	0.016*		0.016*	0.016*		0.016*	0.016*

Table 2.5: Quantitative comparison for the AVT dataset between the benchmark 2D U-Net segmentation method, raw output and after largest connected body filtering (LC), and SeqSeg using the Dice score ($\mathcal{D}$), Hausdorff distance ($\mathcal{H}$) and centerline overlap ($\mathcal{CO}$). The number of branches segmented is shown (Nr. Br.) and * indicates statistically significant difference ($p < 0.05$)

Mod.	Case	Nr. Br.	$\mathcal{D}\uparrow$			$\mathcal{H}\downarrow$			$\mathcal{CO}\uparrow$		
			Seq-Seg	2D U-Net	LC 2D U-Net	Seq-Seg	2D U-Net	LC 2D U-Net	Seq-Seg	2D U-Net	LC 2D U-Net
CT	1	15	**0.924**	0.87	0.843	67.9	**51.7**	172	**0.782**	0.506	0.301
	2	14	**0.951**	0.901	0.881	**28.2**	42.6	36.9	**1**	0.899	0.813
	3	14	**0.925**	0.567	0.299	60.4	**56.8**	132	**0.929**	0.568	0.338
	4	10	**0.912**	0.857	0.858	**17.4**	29.9	29.9	**1**	0.967	0.966
	5	10	**0.898**	0.823	0.758	**27.2**	30.4	96.5	**0.976**	0.834	0.581
	6	8	**0.87**	0.854	0.835	38.8	**29.7**	44.7	**0.973**	0.928	0.783
	7	13	**0.936**	0.787	0.778	**26.4**	39.2	42.7	**0.933**	0.887	0.812
	8	8	**0.911**	0.857	0.799	25	**17.7**	130	**0.981**	0.743	0.412
	9	8	0.875	0.884	**0.885**	26.3	**23.3**	22.3	0.988	**0.989**	**0.989**
	10	11	0.872	**0.874**	0.868	99.6	**52.4**	96.4	0.757	**0.874**	0.764
	11	11	**0.935**	0.86	0.795	19.9	**19.7**	120	**0.98**	0.705	0.459
	12	8	**0.892**	0.846	0.846	**22**	29.4	29.4	**0.992**	0.969	0.969
	13	9	**0.953**	0.892	0.867	**12.8**	23.9	72.2	**0.93**	0.815	0.502
	14	5	**0.918**	0.883	0.886	**23.4**	26.8	26.8	**1**	0.992	0.992
	15	7	**0.916**	0.902	0.899	30.1	22.5	**18.8**	**0.995**	0.978	0.894
	16	8	**0.914**	0.866	0.858	**17.3**	26.4	29.3	**0.995**	0.966	0.966
	17	11	**0.904**	0.863	0.792	**18.6**	25.6	69.2	**0.983**	0.796	0.572
	18	6	**0.941**	0.897	0.867	**12.9**	18.8	75.5	**0.998**	0.837	0.632
	Avg.	-	**0.914**	0.849	0.812	31.899	**31.498**	69.120	**0.955**	0.847	0.708
	p-value	-		5.3e-4*	3.8e-5*		0.609	4.2e-4*		3.8e-5*	2.3e-5*

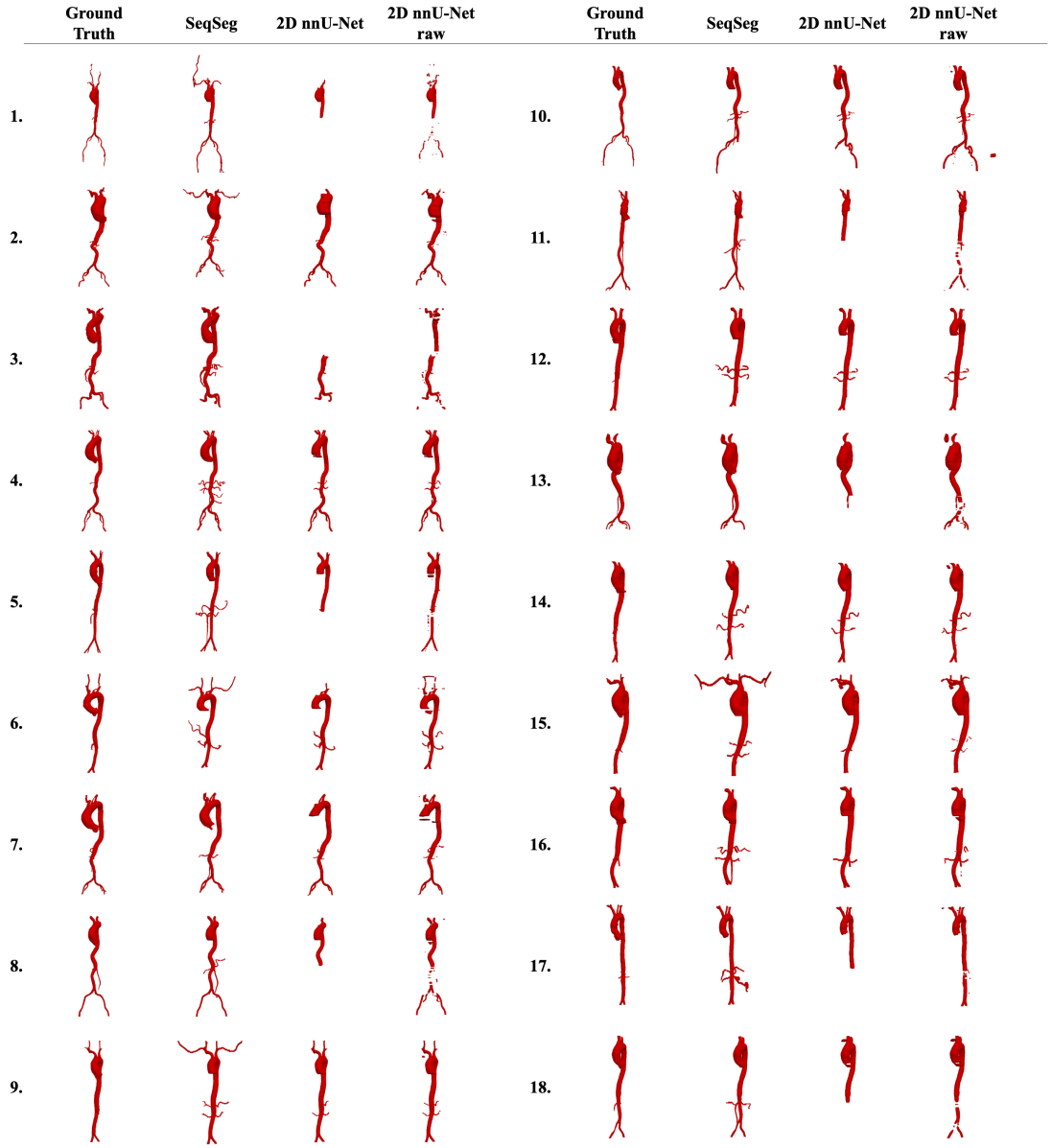


Figure 2.12: Resulting meshes from complete AVT CT test dataset. Each row represents a different vascular model, labeled consistently with Table 2.5

Chapter 3

MeshGrow: Integrated Framework for Unified Cardiac and Vascular Mesh Construction from Medical Images

3.1 Abstract

Patient-specific cardiovascular simulations have become an integral part of cardiovascular research. Cardiovascular simulations provide detailed information difficult or impossible to attain clinically. A primary factor hindering large patient cohort studies and clincal impact of cardiovascular simulations is their dependence on accurate patient-specific three dimensional geometric models which remain time-consuming and costly to costruct from medical image data. Methods have been proposed to automate the model construction process, for either vascular or cardiac purposes. We propose a novel method, MeshGrow, that, automatically, reconstructs both the cardiac chambers as well as the aorta and its main sub-branches, and returns a simulation ready mesh with defined aortic valve connecting the two. We deploy two different methods of model constuction for the cardiac chambers and vascular regions independently to address the specific challanges involved with each. For the cardiac segmentation, we utilize a template deformation based method whereas for the vasculature we implement a step-wise "growth" based approach. We present test results of our method on MR and CT image data.

3.2 Introduction

Patient-specific computational modeling of cardiovascular function is a cornerstone of cardiovascular research and, increasingly, through clinical translation, a part of patient care [32, 25, 22, 107, 73]. Specifically, patient-specific simulations of tissue mechanics and fluid dynamics provide insights into physiological functions that are otherwise difficult or impossible to observe, such as tissue deformation, cardiac electrodynamics, and vessel wall shear

stress. These simulations require anatomically accurate geometric models representing the domain within which they operate. Typically, these models are constructed from medical image scans, where tissue boundaries are delineated using pixel or voxel data as a reference.

A primary challenge limiting the integration of patient-specific computational simulations into clinical practice is the time-consuming and costly process of geometric model construction. This bottleneck hampers large-scale patient cohort studies and prevents real-time inference, both of which are critical in clinical settings [32]. Additionally, most existing methods rely on expert-driven manual input, resulting in a lack of standardization where the final model can vary depending on the modeler [93, 84].

Several software tools have been developed for patient-specific cardiovascular modeling. Most, such as SimVascular and CRIMSON [123, 5], focus on vascular modeling. The standard workflow involves manually generating pathlines along vessel centers, followed by defining vessel lumen contours in planes perpendicular to these pathlines. These 2D contours are then lofted to construct a 3D surface mesh of the vasculature. However, this process heavily depends on manual input, requiring expert users to construct pathlines and refine 2D segmentations. Even with advancements in 2D segmentation techniques, such as level-set methods and machine learning [134, 68, 67], manual correction remains necessary. Additionally, selecting segmentation points along pathlines can lead to contour misalignment, complicating the 3D lofting process. Therefore, there is a clear need for accessible, automated vascular model construction methods.

Cardiac modeling software has not been developed to the same extent as vascular modeling tools. Most modelers rely on general medical image segmentation software such as 3D Slicer [28]. Cardiac segmentation presents unique challenges due to the unclear boundaries between heart chambers in medical images. Moreover, temporal modeling introduces additional complexity, requiring multiple consistent cardiac models across different time points in a sequence. Current methods primarily use basic segmentation techniques, including manual pixel labeling and thresholding [63]. While semi-automated approaches, such as level-set methods and region growing, offer potential improvements, they still require manual correction and parameter tuning [63, 96, 33]. Thus, a comprehensive automated cardiac modeling method remains an unmet need.

Machine learning-based segmentation methods have recently gained attention as a promising approach for automating model construction. Deep neural networks (DNNs) have demonstrated strong performance in medical image segmentation, particularly for pixel-wise classification tasks using architectures such as U-Net [102]. Convolutional neural networks (CNN) have been employed for cardiac segmentation, achieving high pixel classification accuracy. However, they often produce staircase artifacts and disconnected regions, both of which are undesirable for simulation purposes [50, 52, 133]. Alternative approaches, such as graph neural networks, have been used to deform mesh templates, improving topological consistency [51]. However, this method introduces overlapping elements, which LinFlo-Net addresses by enforcing continuous and smooth deformations [87].

For vascular segmentation, deep neural networks have been trained to classify pixels corresponding to coronary arteries, the aorta, and other structures [83, 31, 84, 121]. While

these methods achieve high accuracy for large vessels with well-defined boundaries, they struggle with smaller branches, often resulting in disconnected structures. This limitation poses a significant challenge for simulation applications, where a unified mesh is required. SeqSeg attempts to address this issue by enhancing local vessel segmentation and prioritizing smaller branches [113]. However, pixel-based classification methods still introduce staircase artifacts, necessitating post-processing to generate physiologically accurate surface meshes. Beyond pixel classification, CNN models have also been applied to vascular tracking [127] and bifurcation detection [57, 97], but these approaches require integration with 2D or 3D lumen predictions to create complete vascular models.

To our knowledge, cardiac and vascular model construction methods have been developed largely independently. This is likely due to the historical separation of cardiac and vascular simulation applications. However, the physiology and pathology of the heart and its primary branches are closely interconnected. Future cardiovascular modeling efforts must integrate both cardiac and vascular components to produce comprehensive 3D models with well-defined boundaries.

In this paper, we introduce MeshGrow, a novel method capable of jointly modeling cardiac and vascular structures. Given the distinct topological differences between these anatomical regions, we employ two specialized modeling approaches. For the cardiac chambers, we deform a template mesh to fit the patient-specific anatomy. For the vasculature, we use a seed-based growth approach, expanding outward from the vessel root. Figure 3.1 illustrates the rationale for this two-step approach, highlighting inter-patient anatomical variability and the necessity for distinct modeling strategies.

MeshGrow offers the following key features:

- Automated cardiac localization and modeling: Identifies the cardiac region in medical images and generates patient-specific cardiac models.
- Automated vascular initialization: Locates the aortic root from the predicted cardiac model and initializes vascular tracking and segmentation.
- Extensive aortic tree segmentation: Constructs a detailed segmentation of the aortic tree.
- Patient-specific cardiovascular meshing: Produces a unified cardiovascular mesh with defined inlets, outlets, and an aortic valve for boundary condition placement.

By integrating these capabilities, MeshGrow represents a significant step toward fully automated cardiovascular model construction, facilitating both large-scale studies and real-time clinical applications.

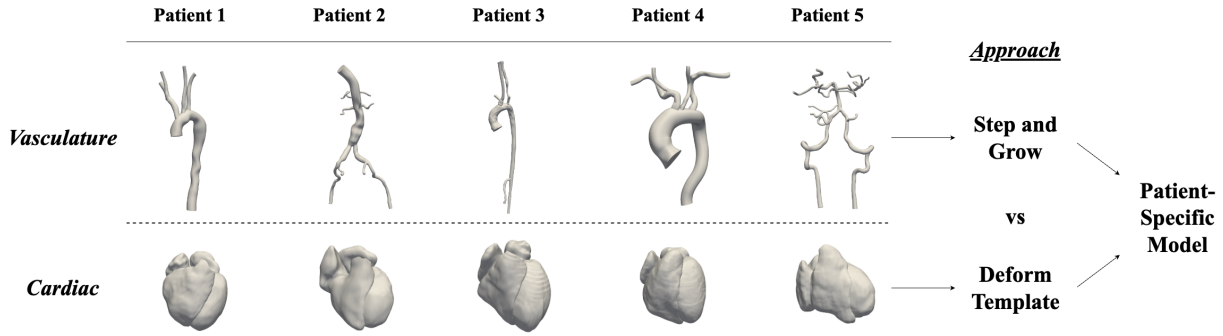


Figure 3.1: The motivation for different modeling approaches for vasculature as opposed to cardiac chambers. Vasculature models vary more extensively across patients and anatomies, both in number of bifurcations and bifurcation locations, whereas cardiac models exhibit stronger topological similarities.

3.3 Methods

In this section, we outline each individual step in the MeshGrow pipeline for comprehensive cardiovascular geometric modeling. We start by explaining an overview of the method's workflow and then describe each step in further detail. Starting with cardiac localization in the global image volume and subsequently how we perform cardiac segmentation using a template based approach called LinFlowNet[87]. Then we describe how we use the cardiac geometric model to initialize the vascular segmentation method SeqSeg along with an explanation of the main components of SeqSeg[113]. Finally, we outline how we unify the cardiac and vascular geometric models into one, and specifically how we define a simulation suitable aortic valve interface between the two.

Workflow

Figure 3.2 shows the method workflow. The input to the proposed method is a medical image scan, CT or MR image data. This ´global´ image volume is fed into a heart localization module which deploys a neural network to perfom a ´rough´ cardiac segmentation. This segmentation is used to crop the global image volume to produce a smaller volume specifically containing the heart and its nearby tissue but ignoring the rest. This cardiac subvolume is used for a detailed 9-class cardiac chamber segmentation using LinFlo-Net method. Using this cardiac segmentation, we place a seed point in the aortic root to initalize SeqSeg, a method to perform vascular segmentation of the aorta and its subbranches. The cardiac and vascular segmentations are finally combined into one using an assembly module. A final surface mesh containing cardiac chambers and the aorta is then achieved using marching cubes method with additional mesh post processing functionality.

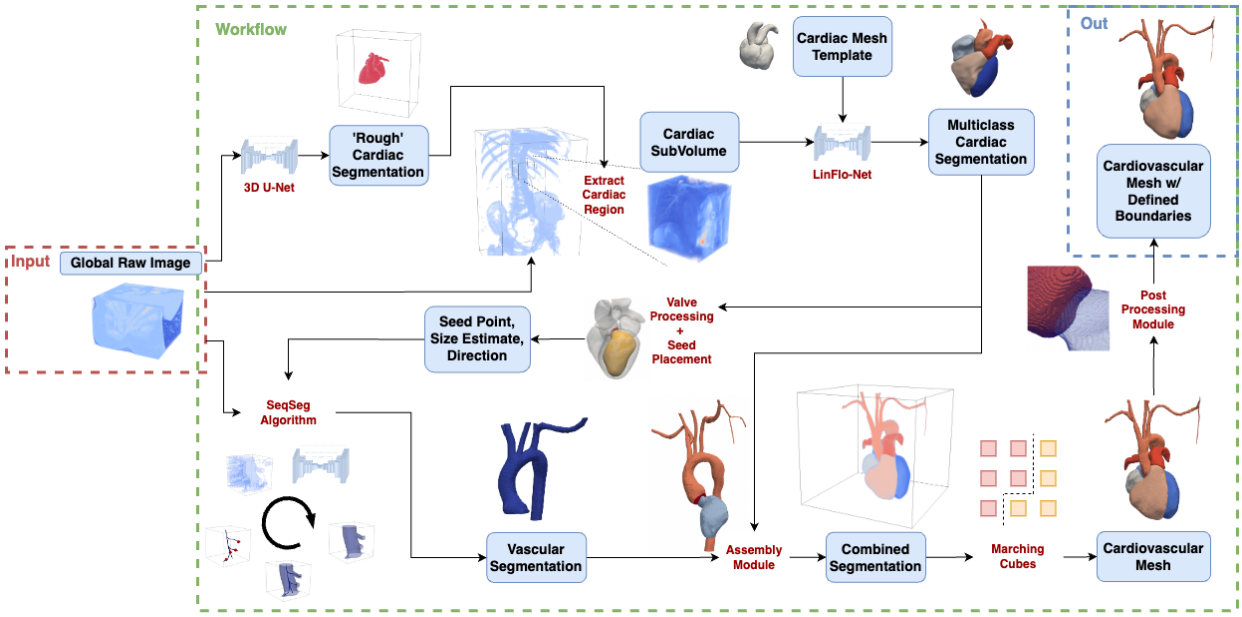


Figure 3.2: The MeshGrow method workflow. MeshGrow takes as input a medical images scan (CT or MRI data) and returns a patient-specific cardiovascular mesh.

Heart Localization

First step is to locate the heart in the global image volume. We utilize a U-Net convolutional neural network (CNN) for the task. We train a model to perform binary voxel classification on the global image volume, where each voxel is labelled as either 1 (belonging to cardiac chambers) or 0 (belonging to any other tissue or organ). Specifically, voxels belonging to right or left ventricle or right or left atrium are labelled 1. We utilize the nnU-Net framework for hyperparameter choosing and training [42]. Table 3.1 shows the specifications of model architecture and data processing.

Cardiac Segmentation

We use LinFlo-Net [87] for cardiac mesh generation. This method aims to transform a template cardiac mesh to capture the patient-specific cardiac morphology as seen in a 3D clinical image (e.g. CT or MRI scan). Briefly, the method utilizes a two-stage mesh deformation process consisting of a learned affine transformation (scaling, translation, and rotation) followed by a diffeomorphic mesh deformation process wherein the vertices of the template mesh are integrated along a learned static 3D flow vector field.

The affine transformation uses a 3D convolutional neural network and predicts the 9 parameters of the transformation (3 parameters each for scale, translation, rotation). This module is trained to minimize the chamfer distance in the L1-norm between the template

Table 3.1: The U-Net architecture and training specifications for the nnU-Net models

Parameter	CT (SeqSeg)	MR (SeqSeg)	CT (Cardiac)	MR (Cardiac)
Intensity Normalization	0.5/99.5% clip + foreground z-score	all image z-score	0.5/99.5% clip + foreground z-score	all image z-score
Image Target Spacing	0.200, 0.0547, 0.0547	0.0859, 0.0625, 0.0625	1.5 1.5 1.5	1.05 0.972 0.972
Patch Size	[20,80,80]	[40,48,48]	[192,112,112]	[96,160,160]
Batch Size	33	57	2	2
Max Nr. Features	320	320	320	320
Nr. Stages Encoder	5	4	6	6
Nr. Stages Decoder	4	3	5	5
Nr. Layers per Stage	2	2	2	2
Nr. Pooling Ops. per Axis	[2,4,4]	[3,3,3]	[5,4,4]	[4,5,5]
Conv. Kernel Size	[3,3,3]	[3,3,3]	[3,3,3]	[3,3,3]

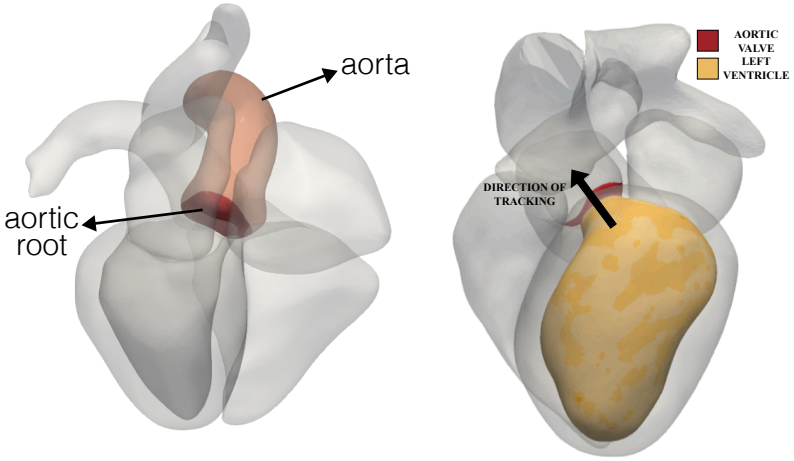
mesh and target mesh. The diffeomorphic mesh deformation module consists of a U-Net architecture that is trained to produce a dense flow vector field in the image space. This module is trained to minimize a weighted sum of various losses including:

1. Chamfer distance in the L1-norm.
2. Normal consistency between template and ground truth meshes.
3. Physics based loss term that prevents volume collapse.
4. Mesh regularization, which includes edge length, normal consistency across faces, and Laplacian smoothing loss.

The reader is referred to the original article for further details on each of these loss terms.

The template mesh is a multi component surface mesh representing the blood pools of the four cardiac chambers, the myocardium of the left ventricle, and segments of the pulmonary artery and aorta where they attach to the heart. Since the vascular segmentation method that we use requires a seed-point for initialization, we add an additional region to the template

mesh consisting of the aortic root. Figure 3.3 shows the template mesh with the newly added aortic root region. The new region was created using MeshLab [16] by cutting the aorta template mesh with a plane.



(a) (b)

Figure 3.3: (a) Template mesh used for cardiac segmentation showing the added aortic root region. (b) Automatic initialization of aorta tracking starting from the aortic valve, determining direction using the aortic valve and left ventricle mesh regions.

Vascular Segmentation

Vascular segmentation is achieved using the Sequential Segmentation (SeqSeg) method [113]. The method constructs vessel segmentation by piece-wise addition of local segmentations predicted using a deep learning based U-Net convolutional neural network. SeqSeg automatically tracks vasculature by inferring direction and bifurcations from local vessel features. The method works for any vasculature but requires a seed point, direction and vessel radius estimate for initialization. For MeshGrow, we implement a new way to automatically initialize SeqSeg around the aortic root with information calculated the cardiac mesh prediction. Each of these components are described in further detail below.

Neural Network Model and Training Strategy

We deploy a deep neural network to perform vascular segmentation of local subvolumes. We use a 3D U-Net convolutional neural network with hyperparameters determined using the nnU-Net framework, see [42] for further details on methodology and [113] for specific parameter values chosen for local aortic segmentation. For training we sample each global vascular segmentation in the training dataset solely around the global vascular centerline.

This allows us to create a new dataset of subvolumes containing only pairs of raw image data and vessel segmentations located around the annotated areas of the global volume. This both makes improved use of available annotated data and introduces a beneficial inductive bias in our neural network model that allows it to generalize to vessels not annotated in training datasets, see [113] for further details.

For training, we use a combined loss l of Dice score $D(x, y)$ and binary cross entropy $BCE(x, y)$ calculated between a pair of raw image data x and its ground truth binary vascular segmentation map y:

$$l(x, y) = 1 - D(x, y) - BCE(x, y) \tag{3.1}$$

Automatic Initialization

The SeqSeg method requires a seed location, direction of tracking and vessel radius estimate for initialization. We automatically initialize SeqSeg at the aortic root by processing the cardiac mesh prediction. The part of the cardiac template mesh representing the aortic valve as a surface is labelled as an additional ninth class. Let us denote regions of mesh vertices as sets S_j, where $S_j := \{v_i | C(v_i) = j\}$ where $v_i \in \mathbb{R}^3$ is a vertex and $C(v)$ is the cardiac class label map. For initialization we use the regions representing left ventricle (LV), labelled $j = 8$ and aortic valve (AV), labelled $j = 9$. Additionally, define corresponding sets N_j of normal vectors $n_i \in \mathbb{R}^3$, $N_j := \{n_i | C(v_i) = j\}$ for each class j. For automatic initialization, these labelled regions are processed as follows:

1. The average node location is calculated for AV and LV

$$\bar{v}_j = \frac{1}{|S_j|} \sum_{v_i \in S_j} v_i \ , \ j \in [8, 9] \tag{3.2}$$

2. The average normal vector is calculated for AV

$$\bar{n}_9 = \frac{1}{|N_9|} \sum_{n_i \in N_9} n_i \tag{3.3}$$

3. Since there are two possible normals to the surface, the direction of tracking $\bar{d} \in \mathbb{R}^3$ is chosen as the AV average normal $\bar{n}_9$ facing away from center of LV $\bar{v}_8$

$$\bar{d} = \begin{cases} \bar{n}_9, & \text{if } (\bar{v}_9 - \bar{v}_8) \cdot \bar{n}_9 > 0 \\ -\bar{n}_9, & \text{otherwise} \end{cases} \tag{3.4}$$

Initial seed point is chosen as two radii distance into the aorta; $s_0 = \bar{v}_9 + \bar{d} * 2R$ where R is aortic root radius estimate. We observe that only a rough radius estimate is enough for initialization, we use $R = 1.3$cm.

Tracking and Bifurcation Detection

Once initialized with above mentioned information, SeqSeg extracts the first subvolume around the aortic root for segmentation. The neural networks outputs a binary segmentation with voxels either predicted as vessel, 1, or not, 0. A surface mesh is extracted from the segmentation using marching cubes [64] and inlet and outlets are automatically detected by defining holes in the surface mesh. These inlet and outlets are then used as source and target points for centerline extraction, see [113] for further details. The centerline is then used to determine direction and size of the vessel, which is then used to move and to extract a new subvolume proportional to the vessel. In the case of multiple branches, one is chosen to move to next and the others are saved in a bifurcation queue for subsequent tracking.

For centerline extraction, we deploy a levelset based method from [35]. A wave propagation is modeled starting from the inlet and moving within the segmentation, with higher propagation speed further away from the vessel boundaries. In mathematical terms, we solve the nonlinear Eikonal equation:

$$|\nabla T(x)|F(x) = 1 \tag{3.5}$$

where $T(x)$ is the time it takes for the wavefront to reach location $x \in \mathbb{R}^3$ and $F(x)$ is the speed function defined at each point. The speed is set to be higher in the center of the vessel compared to locations closer to its walls, so we end up getting a solution $T(x)$ with lower values in the center. Finally, with $T(x)$, we perform gradient descent starting from each outlet point, step-wise leading us back to the start of the wave propagation (source point). See [35] for further details. The output from the centerline extraction is set of connected nodes representing branches and their connections, each node containing a radius estimate of the vessel at that point. The automatic tracking ends when one of three instances occurs; 1) we reach the global image boundary, 2) the segmentation quality is not sufficient enough for centerline extraction or 3) number of steps taken has surpassed maximum number allowed. Post tracking, a global binarized segmentation of the vasculature is assembled.

Global Vascular Segmentation Assembly

The global vascular segmentation is assembled together from all local segmentations predicted while tracking. Note that many of these local segmentations overlap. The probabilistic map output prediction from the neural network model, with each voxel ranging $[0, 1]$, is stored while tracking and is finally averaged together. Specifically for set $\mathcal{Z}$ of local segmentations that included voxel $y_{i,j,k}$ with i, j, k referring to its global index, the final voxel value is:

$$y_{i,j,k} = \frac{1}{|\mathcal{Z}|} \sum_{s \in \mathcal{Z}} z_{i,j,k} \tag{3.6}$$

where $z_{i,j,k} \in [0, 1]$ are corresponding predicted probability the voxel belongs to vessel from each local segmentation. This is done on using the original global image volume resolution. As a last step, we threshold the averaged global probabilistic using $t = 0.5$ to get a binarized

segmentation:

$$y_{i,j,k} = \begin{cases} 1 & \text{if} \quad y_{i,j,k} \geq t \\ 0 & \text{if} \quad y_{i,j,k} < t \end{cases} \tag{3.7}$$

A global vascular surface mesh is created using marching cubes and is smoothed. Finally, a global centerline is also extracted from the global vascular segmentation. This centerline is then used to automatically define caps on the surface mesh for setting boundary conditions.

Automatic Capping at Outlets

Outlets of the global vasculature mesh are detected using the global centerline. Where outlets are detected, caps are defined using a box clip boolean operation on the surface mesh to obtain a planar surface for boundary condition setting. The outlets are located at centerline nodes only connected to *one* other node. The location of the node and the tangent of the centerline there are used to position and rotate the box clip operator. The box is rotated so that one of its sides is perpendicular to the vessel tangent. The size of the box is proportional to the size of the vessel there to prevent accidental clipping of other vessels, specifically the box sidelength l is calculated according to $l = 3 \times R$ where R is the approximated radius of the vessel at that point. Once the box operator is defined, the nodes belonging to the vascular mesh that are located inside the box operator are removed. We then place new elements onto the cut plane to fill the hole in the mesh produced by the box clip operation. These new elements are now labelled as elements belonging to an outlet and are used to set boundary conditions. After defining the outlets of the vascular geometry, we combine the vascular geometry with the cardiac geometry to produce a single unified model.

Union and Mesh Generation

Once we have both a cardiac prediction (from a template-based method) and a vascular prediction (from a tracking voxel-based approach), we unify the two geometric models into one.

Union of Segmentations and Valve Definition

Like mentioned above, we obtain two aortic segmentations; one from LinFlowNet, Y_{LF}, and another from SeqSeg, Y_{SS}. We combine construct the unified model using both segmentations. Specifically, we prioritize the Y_{LF} segmentation in a volume defined around the aortic valve to guarantee correspondence to the predicted cardiac chambers and elsewhere we use Y_{SS}. In full detail, we perform the following:

1. Calculate the center and extent of the aortic valve. We have the elements belonging to the aortic valve labelled in our original cardiac template. Post deformation of the template, we calculate the area of the resulting deformed valve elements, A_{AV}. Additionally we calculate the centroid of the aortic valve surface, C_{AV}.

2. Define a rectangular volume around the aortic valve based on C_{AV} and A_{AV}. Specifically, the dimensions of the rectangle are defined as the bounding box of the aortic valve with added pixels in each dimension to enlarge: $L = 30, W = 30, H = 15$ pixels were added in each dimension, where L, W, H are length, width and height, respectively.

3. In a volume defined around the vascular valve, we use Y_{LF} segmentation and elsewhere we use Y_{SS}

Like mentioned, in a rectangular volume defined around the aortic valve, the LinFlo-Net aortic segmentation is prioritized. Since the aortic valve is labelled in the LinFlo-Net prediction, this allows the aortic valve interface to be defined in the final combined model.

Mesh Generation

Once the final multiclass cardiovascular segmentation is defined, marching cubes algorithm is applied to obtain a surface mesh representation of the geometric model[64]. Finally, we perform global Laplace smoothing to get a smooth surface mesh ready for volumetric meshing and simulation. The end product is a global surface mesh with defined regions for different cardiac and vascular regions, as well as the aortic valve boundary and vascular caps.

Experimental Setup

In this section, we list the datasets used for training and testing of our method. Additionally, we explain the ground truth used for comparison of results as well as the metrics used for quantitative analysis.

Training data

We use the same data as [87] to train LinFlo-Net. The training data consists of data from four public datasets including the multi-modality whole heart segmentation challenge (MMWHS) [133], orCaScore challenge [126], left atrial wall thickness challenge (SLAWT) [48], and left-atrial segmentation challenge (LASC) [122]. In total we had 101 CT samples and 47 MR samples in our dataset.

We use the same data as [113] to train the neural network in SeqSeg. The data is a subset of the Vascular Model Repository (VMR), which is available at `vascularmodel.com`. Specifically, we train a SeqSeg model on a total of 33 aortic models from VMR. Five aorta cases are kept separate for testing purposes.

Test data

Since the objective of this work is to present a unified method to generate consistent cardiac and vascular segmentations, it is important to consider image samples that capture the structures of interest sufficiently. In particular, we are interested in cardiac + aortic segmentations, therefore we consider image samples that capture the whole heart along with the

descending aorta. We include 5 CT and 5 MR images that meet this criteria as part of our test data. CT test samples were selected from the Vascular Model Repository (VMR) (vascularmodel.com) and MR test samples were selected from the MMWHS [133] test dataset. We report the names of the samples in table 3.2.

CT (VMR)	MR (MMWHS)
0027_H_AO_MFS	mr_test_2003
0025_H_AO_MFS	mr_test_2004
0075_H_AO_H	mr_test_2005
0153_H_AO_H	mr_test_2006
0157_H_AO_H	mr_test_2006

Table 3.2: Test samples used to evaluate the performance of the proposed workflow. CT samples were taken from the VMR dataset and MR samples were taken from the MMWHS test dataset.

Ground-truth segmentations of the vasculature is available for the CT samples through the VMR dataset. Similarly, ground-truth segmentations of the cardiac regions are available for the MR samples through the MMWHS dataset. (Note that the MMWHS dataset does not provide the ground-truth explicitly. Rather, the challenge organizers provide encrypted executable scripts which may be used to evaluate the accuracy of model predictions.) Therefore, we had to generate ground-truth cardiac segmentations for the CT test data and ground-truth aortic segmentations for the MR data.

We used TotalSegmentator [125] to generate the ground-truth cardiac segmentations using the `heartchambers_highres` subtask followed by manual verification. The generated segmentations for all samples except 0157_H_AO_H appeared reasonable to us. Sample 0157_H_AO_H required manual correction of the segmentation of the myocardium, see Figure 3.4. Indeed, this particular failure case underscores the efficacy of a template based method since mesh templates provide a strong shape prior for model predictions, enforcing topological consistency and anatomical accuracy.

For aortic ground truth segmentations, we used SimVascular modeling pipeline [123]. The models were constructed by manual pathline generation and subsequent 2D segmentation of lumen cross sections. 2D segmentations were done using thresholding and manual spline point placements.

Metrics

For quantitative comparison between the segmentation results from our method, Y and the ground truth segmentations, X, we compute Dice score:

$$\mathcal{D}(X, Y) == \frac{2 \cdot \|X \cap Y\|}{\|X\| + \|Y\|} \tag{3.8}$$

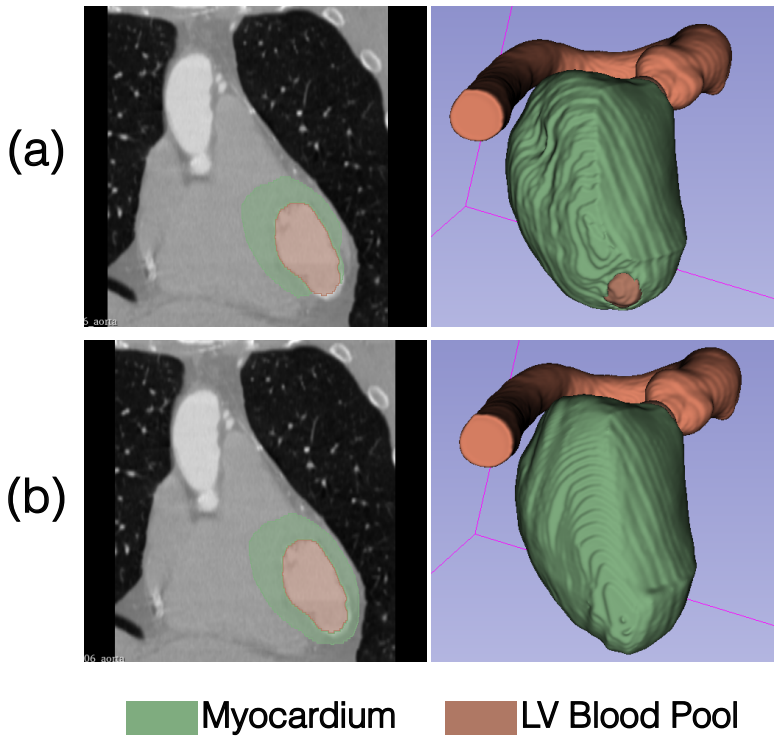


Figure 3.4: (a) Cardiac segmentations of sample 0157_H_AO_H produced by TotalSegmentator contain a hole at the apex of the myocardium. (b) We manually fixed this segmentation for evaluation purposes.

For aortic vascular segmentation results, we do a more comprehensive analysis and compute Hausdorff distance, $\mathcal{H}$ and centerline overlap, $\mathcal{CO}$, as well:

$$\begin{aligned} \mathcal{H}(X,Y) &= \max\{d(X,Y), d(Y,X)\}, \\ \text{where} \quad d(X,Y) &= \sup_{x\in X}\inf_{y\in Y} d(x,y) \end{aligned} \tag{3.9}$$

$$\mathcal{CO}(Y, C_t) = \frac{\int_{C_t} Y\, dx}{\int_{C_t} dx} \tag{3.10}$$

where C_t is the ground truth centerline. Since our vascular segmentations are, in most cases, more extensive than the ground truth being compared to, we clip the prediction to place outlets at the same locations. Figure 3.5 shows the process for clipping the vascular segmentations to match outlets with the ones chosen in ground truth models.

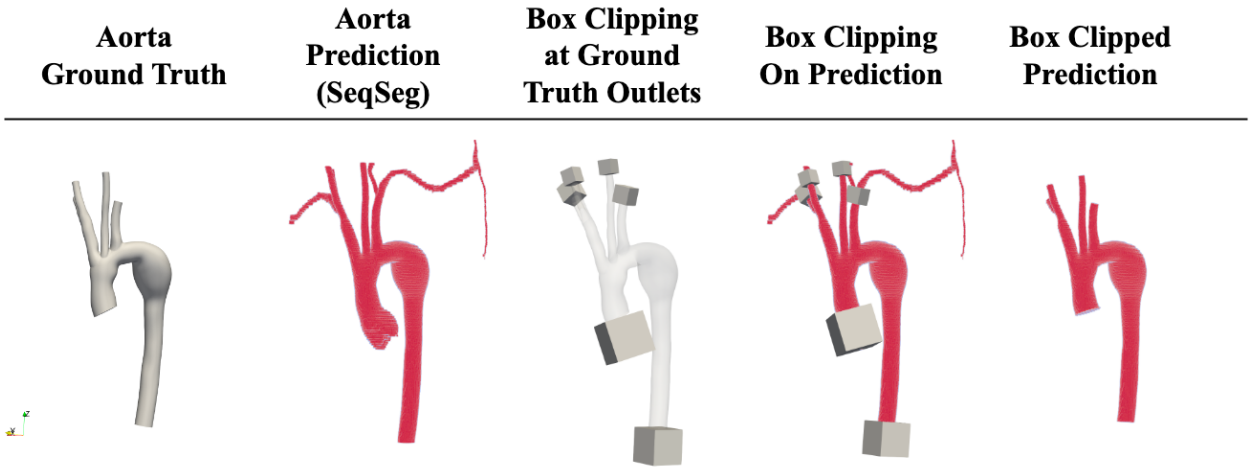


Figure 3.5: Box clipping resulting vascular segmentations to place outlets at same locations as chosen for manual ground truth case for geometric comparison.

3.4 Results

We test the method on 10 withheld test cases, 5 CT and 5 MR image data. Tables 3.3, 3.4 show the quantitative metric results for our method on these test cases, cardiac and vascular, respectively.

Binary Cardiac Segmentation

The method begins by doing a 'rough' binary cardiac segmentation in the global image volume to estimate the location and bounds of the cardiac structures. See Figure 3.6 for qualitative comparison of the binary segmentation with ground truth cardiac models for the five held out CT test cases.

Cardiac Model

After localizing the cardiac structures and extracting a subvolume arounds its bounds, we predict a multiclass cardiac model for it. The Dice score results for the different cardiac chambers is presented in Table 3.3, for the myocardium, left and right ventricle, left and right atrium, pulmonary artery and aorta. The average Dice score for each chamber is shown in bold font. The Dice score was highest for the ventricles; on average 0.94 and 0.903, for the left and right, respectively. On average, the Dice score for atrium predictions were 0.892 and 0.884 for the left and right, respectively. The resulting meshes are presented for comparison with ground truth models in the first two columns in Figure 3.7.

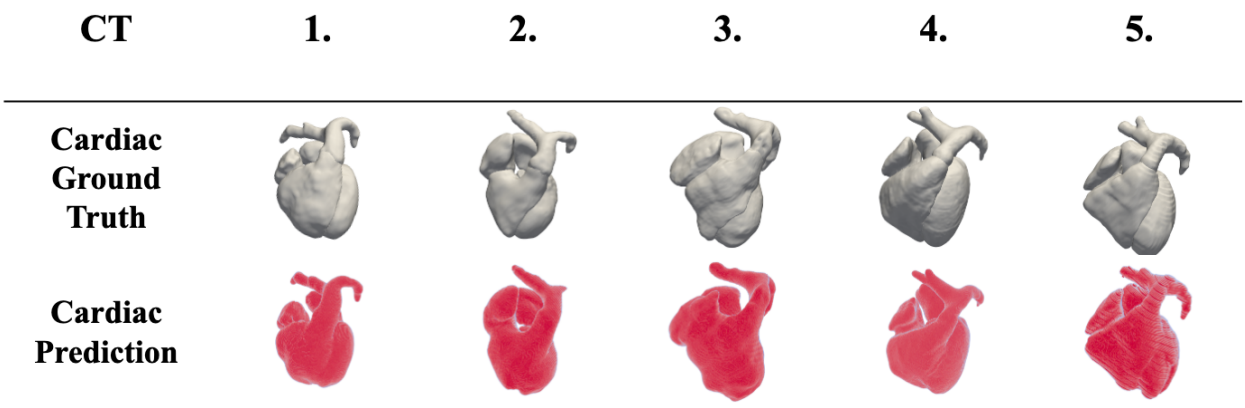


Figure 3.6: Comparing the 'rough' binary cardiac segmentations with the ground truth for the CT data. The binary segmentations are used to approximate the bounds of the cardiac region, which is subsequently used for detailed multiclass cardiac model prediction.

Vascular Model

As explained in Sec 3.3, we place a seed point for performing aortic vascular segmentation using the aortic root predicted in the cardiac model. The quantitative metric results for the five CT test cases are presented in Table 3.4. On average, our method gets a Dice score of 0.959, Hausdorff distance of 0.702 (in pixels), and centerline overlap of 0.978. For a qualitative comparison, the resulting aortic meshes and the ground truth as shown in Figure 3.7, third and fourth columns.

For comparison between a step-wise localized vascular segmentation method (SeqSeg) and a template based method (LinFlo-Net), we present the respective resulting meshes in Figure 3.8. The template deformation prediction outputs only a truncated

Combined Cardiovascular Model

The resulting cardiac and aortic vascular segmentations are combined into unified cardiovascular models, shown as last column in Figure 3.7.

Table 3.3: Quantitative metric results on cardiac chambers of the geometric model; MYO: myocardium, LA: left atrium, LV: left ventricle, RA: right atrium, RV: right ventricle, PA: pulmonary artery, AO: aorta. Metric, $\mathcal{D}$: Dice score as defined in Sec. 3.3

			$\mathcal{D}\uparrow$						
	Case	Name	MYO	LA	LV	RA	RV	PA	AO
CT	1	0174	0.866	0.915	0.955	0.916	0.943	0.856	0.964
	2	0176	0.806	0.814	0.909	0.891	0.913	0.692	0.953
	3	0188	0.831	0.934	0.940	0.823	0.840	0.852	0.957
	4	O15032	0.880	0.937	0.959	0.927	0.926	0.678	0.949
	5	O34421	0.861	0.862	0.937	0.866	0.892	0.893	0.972
	Ave.	-	**0.849**	**0.892**	**0.940**	**0.884**	**0.903**	**0.794**	**0.876**
MR	1								

Table 3.4: Quantitative metric results for aortic vascular regions of the geometric model; AO: aorta from SeqSeg. Metrics shown; $\mathcal{D}$: Dice score, $\mathcal{H}$: Hausdorff distance score, $\mathcal{CO}$: centerline overlap score, as defined in Sec. 3.3

	Case	Name	Nr. Steps	Nr. Branches	$\mathcal{D}\uparrow$	$\mathcal{H}\downarrow$	$\mathcal{CO}\uparrow$
CT							
	1	0174	232	6	0.972	0.412	0.984
	2	0176	295	6	0.949	0.921	0.994
	3	0188	76	6	0.953	1.065	0.964
	4	O15032	138	5	0.957	0.668	0.988
	5	O34421	29	4	0.964	0.443	0.961
	Ave.	-			**0.959**	**0.702**	**0.978**
MR	1						

3.5 Discussion

We present a novel automatic cardiovascular model construction method. The method takes in a medical image scan of a patient and outputs a patient-specific three-dimensional simulation ready mesh of the cardiac chambers and the aorta with defined inlets, outlets and aortic valve interface. This is, to our knowledge, the first comprehensive cardiac *and* vascular model construction method.

We deploy two different deep learning based modeling approaches for cardiac and vascular model construction independently. Both modeling approaches require neural network training. We train and test the method on a dataset of CT image scans. We compare the outputs of a held out test set to manually constructed 3D models. The cardiac chamber segmentations achieve, on average, Dice scores of 0.849, 0.892, 0.940, 0.884, and 0.903 for the myocardium, left atrium, left ventricle, right atrium and right ventricle, respectively.

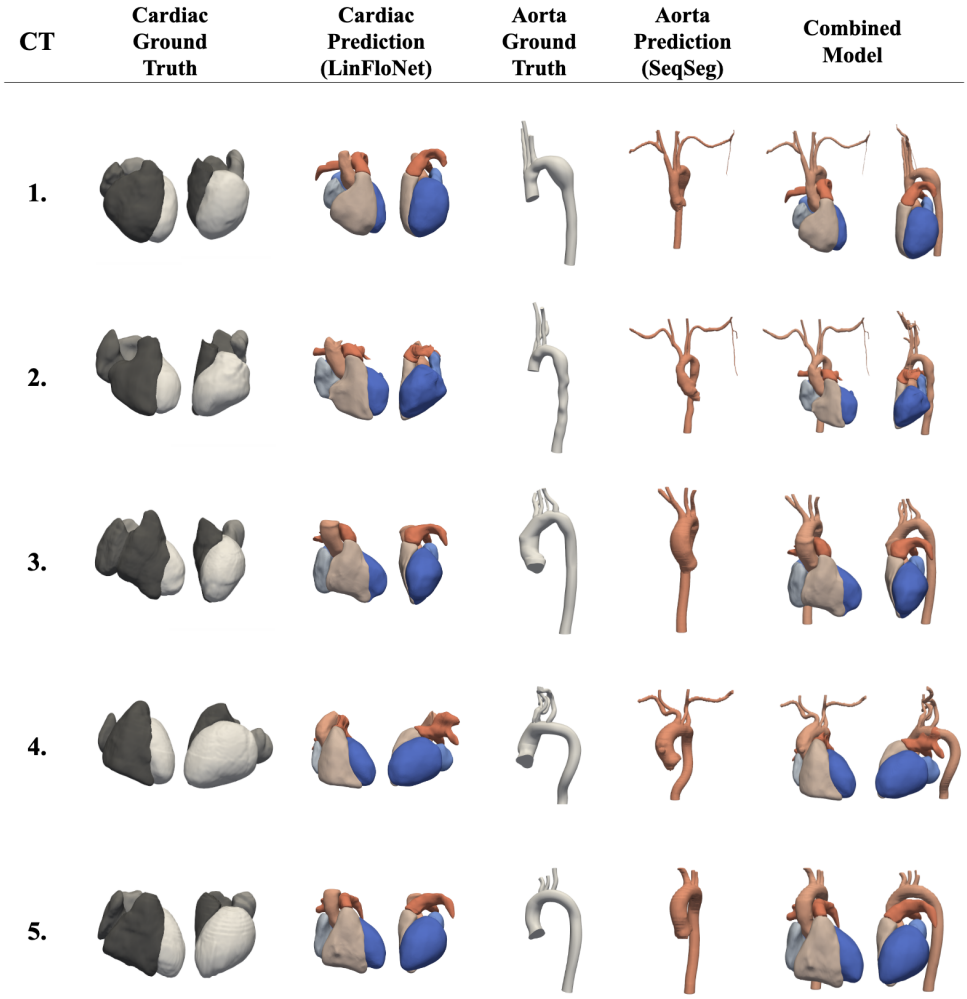


Figure 3.7: Qualitative results for our method compared to ground truth segmentations. From left: cardiac region ground truth, cardiac region prediction results (from LinFloNet), aorta ground truth, aorta prediction results (from SeqSeg), and the final combined cardiovascular model.

Because of the nature of SeqSeg being a step-wise growth approach, the vascular segmentations often extend further than the ground truth cap placements. After placing caps at the same locations, the vascular segmentations, on average, get a Dice score of 0.959, Hausdorff distance score of 0.702 pixels, and centerline overlap score of 0.978.

We demonstrate the advantage of anatomy specific modeling approaches. Motivated by template deformation approaches for cardiac modeling by Kong, Narayanan, and Shadden, we apply it for our purposes here [51, 87]. Because of increased variability in vascular

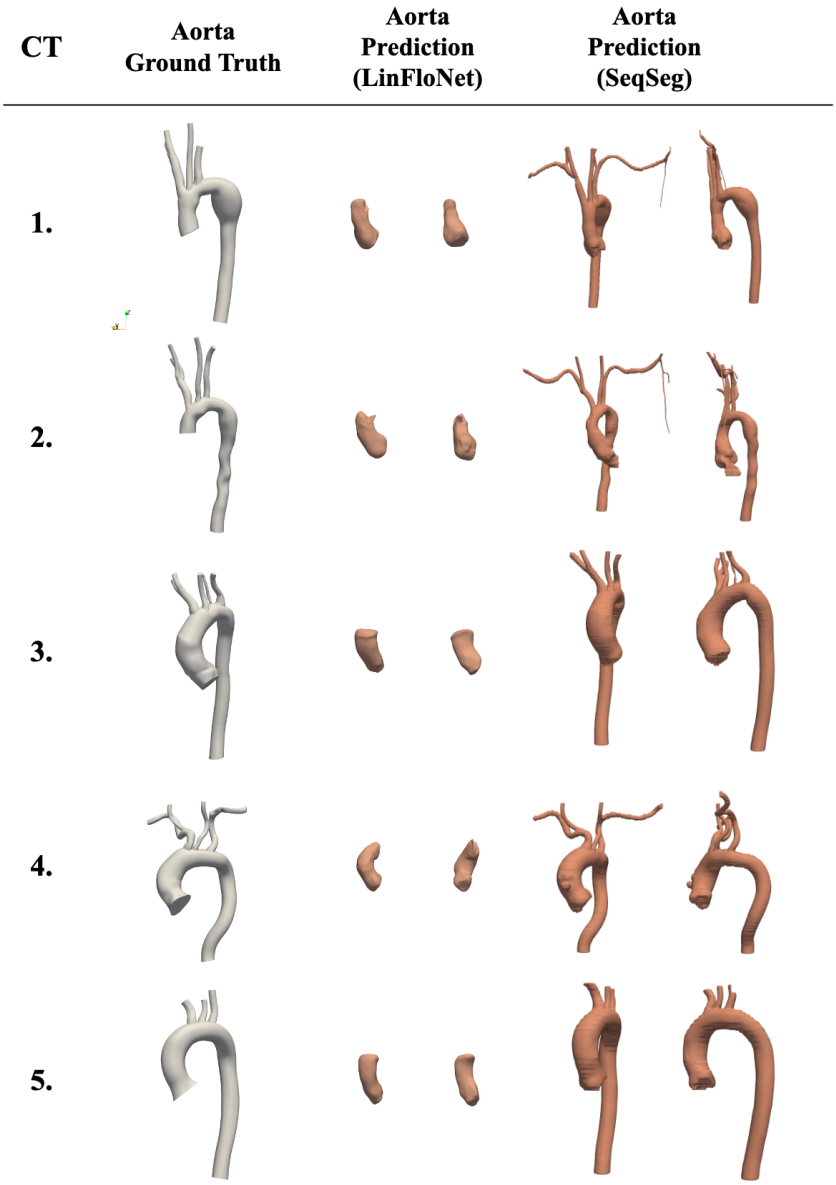


Figure 3.8: Qualitiative comparison between the ground truth, manually segmented aorta and the results from our methods. Here we show the limitation of a template based approach (LinFloNet) for aortic segmentation, as compared to a stepping and growth based approach (SeqSeg). From left: ground truth model, aortic segmentation from LinFloNet method and aortic model results from SeqSeg method.

topology, we deploy a vessel tracking “step taking“ based modeling approach. We compare

the vascular segmentation of a template based approach with ours (SeqSeg[113]) and show that template deformation struggles with outputting extensive vasculature models.

In this work, we have well defined starting locations of downstream vasculature of interest, the valves, making tracking approaches requiring seed points, like SeqSeg, easier to implement. Instead of having to locate the vascular root from a global image volume, we utilize the predicted cardiac model for seed point placement. Our method depends on locating the cardiac region in an image volume, but that is an easier task (we show by a simple U-Net binary segmentation) than directly predicting vascular seed location.

A main limitation of this work is the small test dataset. A dataset of combined cardiac and vascular models is not yet available for general research, hence motivating the work presented, so we had to construct the ground truth test data in house and could not make it larger because of limited resources. Another main limitation is the missing investigation of the effect of these machine learning based modeling approaches on the downstream simulation results, and specifically comparing them to simulation results from manually constructed models. This remains an open area for research. In this work, we demonstrate an application using these geometric models, a fluid-structure interaction simulation involving the left ventricle and the aortic tree vessels, but we do not demonstrate applying boundary conditions to the aortic valve directly. Additionally, the motion used to prescribe deformation boundary conditions in the simulation is estimated, and a real motion has still yet to be tested. However, the authors believe this work is beyond the scope of this paper, where the purpose of the simulation was to show an example of using the paper's proposed method for a downstream task.

In conclusion, we present an automatic method to construct comprehensive three dimensional cardiovascular geometric models based on medical images. We hope this work can facilitate patient-specific cardiovascular simulation research and its entry into clinical settings for patient care. Future directions of the work include investigating the effects of machine learning based automated model construction methods on simulation results as well as incorporating additional anatomical features, e.g. valves and veins.

Chapter 4

MIROS: Patient-Specific Reduced Order Model Simulation in Minutes

4.1 Abstract

Patient-specific cardiovascular simulations have become an integral part of cardiovascular research. While three dimensional fluid structure interaction simulations result in detailed high resolution hemodynamic solutions, reduced order models (ROM) offer a lower computational cost option while capturing considerable dynamic behavior. However, despite these ROMs, either lumped parameter networks (0D) or Navier-Stokes averaged across the cross section (1D), having the ability to model complex blood flow in vascular networks, they require patient-specific geometric models (which can take hours to construct) and several different packages to run, making their large scale use infeasible. We aim to address this issue by presenting MIROS (Medical Image to Reduced Order Simulation) framework. This method is a fully integrated, semi automatic workflow to go from a medical image scan to reduced order simulation results in minutes time. We utilize machine learning methods (SeqSeg) for automatic geometric model construction based on image data and automatically place vessel caps for inflow and outflow boundary conditions. We validate our method on datasets of CT and MR aortic and aortofemoral models, achieving an average error $< 3\%$ for 0D simulations and $< 4\%$ for 1D simulations. We hope this work may facilitate the use of reduced order models both in cardiovascular research and, ultimately, patient care.

4.2 Introduction

In the past three decades, the development of image-based modeling and computational fluid dynamics (CFD) for simulating blood flow [117] [85] has greatly enhanced our understanding of pathogenesis [78] [7], enabled more personalized diagnostics [95] [107], and improved treatment planning for various cardiovascular diseases [95] [74]. Detailed fluid-structure interaction (FSI) simulations and digital twins show great potential to serve as noninvasive

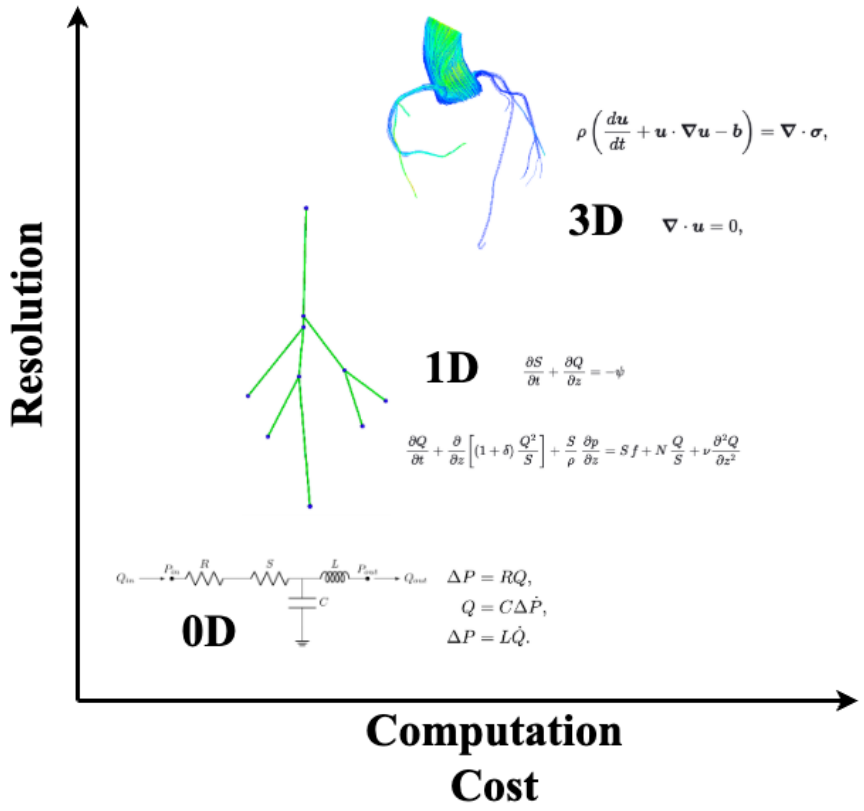


Figure 4.1: Increased resolution in hemodynamic modeling requires additional compution cost. Reduced order models, 0D and 1D, offer an alternative to the costly 3D simulations, requiring much lower computational resources but losing out on solution resolution. Parts of figure are borrowed from SimVascular documentation, `https://simvascular.github.io`[123].

tools for clinicians [9]. As a result, these techniques not only help advance basic research, they also expand into patient care, supporting more efficient and tailored therapeutic interventions.

One key advantage of these FSI models and digital twins lies in their ability to capture the complex mechanics between blood flow and vessel walls, thereby offering deeper insights into the onset and progression of vascular pathologies [116]. This physics based fluid-structure interaction perspective is imperative for modeling disease mechanisms, e.g. thrombosis formation [101], fibrosis remodeling [11] [118], and aneurysm evolution [110]. Moreover, through high-fidelity simulations, clinicians can compare multiple therapeutic or surgical interventions in silico, testing "what-if" scenarios specific to each patient's vascular anatomy without exposing them to invasive procedures [116]. However, these detailed simulations often demand specialized expertise in numerical methods [54], large computing

resources[82], and substantial processing times [72], thus presenting a major hurdle for clinical adoption. In addition, building a patient-specific 3D anatomical model from volumetric angiography (CT or MRI) typically also requires specialized expertise and a significant time investment, further limiting the scalability of these methods for routine clinical use. Therefore, achieving faster, more automated, and cost-effective modeling frameworks is critical for enabling patient-specific simulation in cardiovascular research and care.

Reduced-order models (ROMs) offer a promising alternative to full-scale CFD simulations by substantially lowering computational costs and turnaround times[80] [94]. By simplifying the governing equations—e.g., through lumped-parameter (0D) or one-dimensional (1D) approaches—ROMs can capture essential hemodynamic behavior while bypassing the intricacies that make FSI simulations computationally expensive. See Figure 4.1 for a schematic comparison between 0D, 1D and 3D simulations, both in terms of computation cost and solution resolution. Even if the results are approximate and at lower resolution, ROM simulations can be highly informative in several scenarios including timely decision support, screening, boundary condition tuning, uncertainty quantification and treatment design.

Yet, despite the advantages of ROM simulations for hemodynamic modeling, the lack of automated and fully integrated workflows makes extensive patient cohort studies and clinical research infeasible. Creating an image-based simulation is labor-intensive and requires a spectrum of expertise. On one hand, generating an anatomically accurate and simulation-suitable model is time-consuming, often taking hours, and becomes even more challenging for complex and highly branched geometries (e.g., coronary arteries and vascular trees). On the other hand, selecting an appropriate ROM solver and configuring it correctly to produce consistent physiological simulations demands expertise in coding, numerical methods, hemodynamics, and data visualization. Finally, integrating these steps into a seamless pipeline remains a significant challenge, as researchers often rely on fragmented tools that require substantial manual intervention and customization at each stage. This lack of a cohesive workflow limits the scalability of ROM simulations for research and creates significant barriers to their adoption in clinical practice.

With this in mind, we developed a streamlined process to produce reduced-order model (ROM) simulations of patient-specific hemodynamics directly from volumetric angiography. Our framework integrates lumped-parameter and 1D Navier–Stokes solvers—both of which are readily available in the open-source SimVascular software [123]—with machine learning based segmentation techniques, Sequential Segmentation (SeqSeg) [113], to automate vascular model construction. The result is a workflow that significantly reduces the time and expertise needed to go from medical images to informative simulations of blood flow. This cumulates in patient-specific simulation results ready on the order of minutes after acquiring a medical image scan.

To evaluate the effectiveness of the medical-image-to-reduced-order-simulation (MIROS) framework, we present a comparative study contrasting the ROM simulation outcomes generated from SeqSeg-constructed vascular models against those obtained from traditional, manually constructed counterparts from the open access Vascular Model Repository (VMR) at `https://vascularmodel.com`. Our findings reveal that the automated MIROS pipeline

delivers accuracy and reliability on par with manual workflows–well-predicting bulk flow and pressure waveforms–underscoring its potential for diverse applications. By leveraging the strengths of ROMs—lower computational costs and faster turnaround—this integrated approach can not only facilitate fundamental research endeavors, including boundary condition tuning and uncertainty quantification, but also advance potential practical hemodynamic assessments in clinical settings.

To summarize, we herein present a fully integrated and almost automated framework for patient specific reduced order modeling capable of the following:

1. After being initialized with a seed point, automatically constructing a patient-specific geometric model of vasculature from patient medical image scan (CT or MRI)

2. Automatically processing the patient-specific geometric model and placing outlet and inlet caps for boundary conditions

3. After boundary condition placement, automatically setting up reduced order models (0D or 1D) of the representative vasculature and solving for resulting flow and pressure

4. Outputting the simulation results in easy to visualize format for downstream analysis

4.3 Methods

We present MIROS, a novel automated framework for patient-specific blood flow reduced order model simulation results. In this section we begin by covering the overall workflow of the framework. Subsequently, we discuss the reduced order models for blood flow we use, 0D and 1D, followed by the methods we use for automated patient-specific geometric modeling, and post processing of the resulting geometric information. Finally, we cover the solvers we use and the experiments we ran and for which we present results in subsequent sections.

Workflow

The workflow of the framework can be viewed in Figure 4.2. The input is a medical image scan of a patient along with a seed point and radius estimate in the vessel of interest. The seed information is used to initialize the automatic vascular model construction method, SeqSeg [113], which automatically tracks the downstream vasculature and assembles a vascular segmentation to match the medical image. The resulting segmentation is converted to a surface mesh using marching cubes algorithm which is then used to extract a centerline with accompanied radius estimates. The centerline is then input for further processing to acquire the patient-specific geometric attributes necessary for reduced order modeling. After the reduced order model equations with relevant attributes have been setup, they are solved using the specific solvers (0D or 1D). Each step is explained in further detail below.

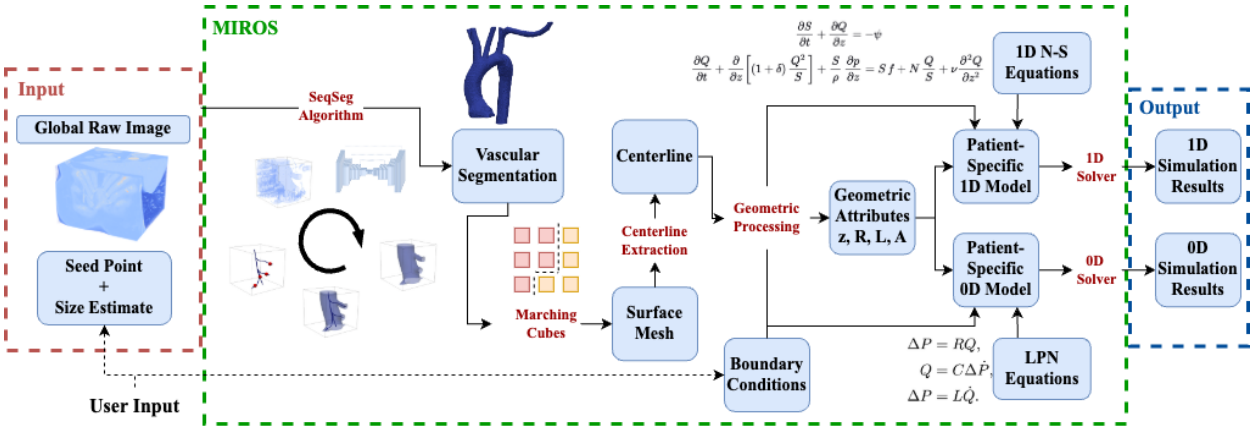


Figure 4.2: The method workflow. Starting from a medical image volume and seed point, automatic geometric model construction, post processing, reduced-order model setup and, after solving, ending with patient-specific 0D and 1D simulation results of blood flow.

Reduced Order Model

Reduced order models balance computational efficiency with the ability to capture essential hemodynamic behavior. Below, we briefly summarizes the principles of 0D and 1D modeling approaches.

1D

In reduced-order modeling of cardiovascular systems, **one-dimensional (1D) modeling** simplifies the governing Navier-Stokes equations by integrating them across the cross-section of the blood vessel, under the assumption of an axisymmetric velocity profile. The 1D solver we employ is `svOneDSolver`, and the detailed implementation is discussed in the original paper [94]. As a summary, the 1D reduced-order of Navier-Stokes equations governing equations is the follow:

$$\frac{\partial Q}{\partial t} + \frac{4}{3}\frac{\partial}{\partial z}\left(\frac{Q^2}{S}\right) + S\frac{\partial P}{\partial z} = Sf - \frac{NQ}{S} + \nu\frac{\partial^2 Q}{\partial z^2} \tag{4.1}$$

$$\frac{\partial S}{\partial t} + \frac{\partial Q}{\partial z} = 0, \quad z \in \Omega_{1D},\ t \geq 0. \tag{4.2}$$

Here, Q is the flow rate, P the pressure, and S the cross-sectional area along the axial coordinate z. The parameter N depends on the velocity profile, with $N = \frac{8\mu}{\pi}$ for a Poiseuille flow profile. Constitutive relations further define the pressure-area relationship, such as the linear or Olufsen models:

$$\text{Linear: } P(z,t) = P_0(z) + k_0 \left(\sqrt{\frac{S(z,t)}{S_0(z)}} - 1 \right), \tag{4.3}$$

$$\text{Olufsen: } P(z,t) = P_0(z) + \frac{4}{3} \left(k_1 e^{k_2 r_0(z)} + k_3 \right) \left(1 - \sqrt{\frac{S_0(z)}{S(z,t)}} \right). \tag{4.4}$$

where, E represents the Young's modulus of the vessel wall material, h denotes the thickness of the vessel wall, P_0 refers to the reference pressure, r_0 is the reference radius, and k_0, k_1, k_2, and k_3 are material constants determined empirically [92].

The solver automatically detects junctions and stenosis leveraging a series of open source python libraries such as SciPy and VTK. Boundary conditions include continuity of static pressure at vessel junctions and conservation of mass. Numerical solutions utilize implicit finite element methods, ensuring efficiency and stability for patient-specific geometries.

0D

Zero-dimensional (0D) models, or lumped parameter networks (LPNs), reduce solving 3D Navior-Stoke equation for hemodynamic analysis to temporal distributions of flow and pressure. Likewise, the detailed disuccssion of solver implementation is in the original paper [94]). To briefly summarize, analogous to electrical circuits, the basic elements include resistance (R), capacitance (C), and inductance (L):

$$\Delta P = RQ, \tag{4.5}$$

$$Q = C\Delta\dot{P}, \tag{4.6}$$

$$\Delta P = L\dot{Q}. \tag{4.7}$$

These parameters depend on vessel geometry and properties [76]:

$$R = \frac{8\mu l}{\pi r^4}, \quad C = \frac{3lr^3}{2Eh}, \quad L = \frac{\rho l}{\pi r^2}. \tag{4.8}$$

Nonlinear effects, such as those due to stenoses, are captured by augmenting Poiseuille resistance in the 0D model:

$$R_{\text{expansion}} = K_t \frac{\rho}{2S_0^2} \left(\frac{S_0}{S_s} - 1 \right)^2 |Q|. \tag{4.9}$$

where $K_t = 1.52$ is a commonly used empirical correction factor and S_0, S_s are the lumen areas proximal to and at the location of the stenosis, respectively [111][81][43].

The zero-dimensional (0D) simulation methods have been implemented as an open-source Python package available at `https://github.com/SimVascular/svZeroDSolver`. The `svZeroDSolver` framework leverages the modular nature of 0D models, enabling the construction and simulation of various configurations. Commonly used elements, such as Poiseuille-based resistors, are included. Each 0D building block is associated with a set of equations governing the local flow physics.

In both 1D and 0D models, assumptions of continuity in static pressure and conservation of mass govern junction behavior. The 1D formulation incorporates wave propagation, a feature absent in 0D models. However, this added capability comes with approximately six times the computational cost[94]. The computational efficiency of 0D models makes them particularly suited for large-scale parametric studies or uncertainty quantification, despite their inherent limitations in capturing local flow features. However, note that there is a newly iterated, upgraded, and better performing version `svZeroDSolverPlus` available.

Besides the impact of various numerical methods, the accuracy of any numerical simulation is also largely dependent on model construction. Unlike the traditional manual process of creating a 3D anatomical model, here we leveraged a recent development of a machine-learning based segmentation method that automatically constructs anatomical models from medical images.

Automatic Model Contruction

The three dimensional patient-specific geometric models are constructed automatically using SeqSeg (short for Sequential Segmentations)[113]. SeqSeg takes in as input a medical image scan of patient (CT or MR data) and a seed point and tracking direction, and outputs a three dimensional surface mesh of the resulting vasculature. The overall SeqSeg method is displayed in Figure 4.3. In this section, we will briefly explain the SeqSeg method, for more details we direct readers to the original paper[113].

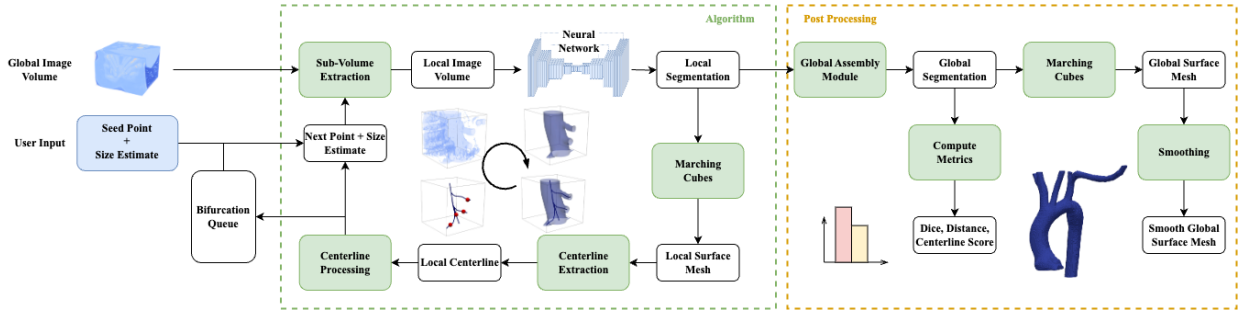


Figure 4.3: The SeqSeg method used for automatic patient-specific geometric model construction from CT,MR image scans. SeqSeg is initialized with a seed point and direction at the proximal most part of the vasculature. Figure is borrowed from SeqSeg paper[113]

Algorithm

A figure describing the algorithm can be seen in Figure 4.3. The SeqSeg algorithm is initialized with a single seed point, direction and a radius estimate in vasculature of interest. The seed point and radius is estimate is used to define the first subvolume for segmentation. The SeqSeg method performs local segmentations around a vessel segment. Once a segmentation is predicted, SeqSeg calculated a local centerline of the vessel segment. The centerline is used to choose the next point to move to and predict segmentation over. In the case of bifurcations, additional detected branches are accumulated in a 'bifurcation queue' where they are revisited once the current branch has finished. Simply put, SeqSeg loops over and moves to branch points detected from local segmentation predictions. The segmentation is done using a deep neural network trained on a dataset of vessel segmentation subvolumes.

Segmentation

The segmentation prediction is done using a nnU-Net neural network architecture [42]. The nnU-Net is a U-Net neural network architecture with hyper parameters configured for easier implementation. The U-Net is a convolutional neural network, typically used for medical image segmentation because of its impressive performance on pixel classification tasks [102]. The neural network is trained on a dataset of vascular segmentation subvolumes. Specifics on dataset use are discussed in Section 4.3. These global vascular segmentations are used to extract thousands of local subvolumes for training. The loss used in training is a combination of negative Dice score ($\mathcal{D}$) and binary cross-entropy ($\mathcal{CE}$), calculated per batch:

$$\mathcal{L} = \sum_{i}^{N_b} (1 - \mathcal{D}(Y_{p,i}, Y_{t,i}) - \mathcal{CE}(Y_{p,i}, Y_{t,i})) \tag{4.10}$$

$$\mathcal{D}(Y_p, Y_t) = \frac{2 \cdot \|Y_p \cap Y_t\|}{\|Y_p\| + \|Y_t\|} \tag{4.11}$$

$$\mathcal{CE}(Y_p, Y_t) = \frac{1}{n} \sum_{y \in Y} (y_t \cdot \log y_p + (1 - y_t) \cdot \log (1 - y_p)) \tag{4.12}$$

where N_b is batch size, Y_p and Y_t are respective prediction and ground truth segmentation masks, respectively, and n is the total number of voxels.

Centerline Extraction and Step Taking

Once segmentation is done, a centerline is calculated to describe the branch (or branches in the case of bifurcations) geometry. The centerline describes the layout of the branches with locations and radius estimation along them. The centerline is computed by solving the Eiqonal equation:

$$|\nabla T(x)| F(x) = 1 \tag{4.13}$$

for $T(x)$, a time distance field, for a given 'speed' defined at all locations[35]. The equation models wave propagation, that starts at the inlet of the vascular segmentation (previous point). With this time distance field, the centerlines are traced with gradient descent, starting at the detected outlets. For more details, see [35]. The radius is estimated using a signed distance field of the local segmentation predicted, similarly calculated by solving Eq. 4.13 for the boundary voxels in an image.

Global Assembly

As SeqSeg takes steps and performs local segmentations, they are accumulated into a global 'assembly' segmentation. The global segmentation is an average of all local segmentations. Once assembled, a global surface mesh is attained using marching cubes and smoothed using Laplace smoothing[64].

Post Processing

With the automatic model construction complete, the next step is to refine these outputs for simulation. In the following section, we describe the essential post-processing procedures—such as boundary clipping, centerline extraction, and solvers—that prepare the automatically generated models for robust hemodynamic simulations.

Boundary Clipping

SeqSeg can capture more vasculature than contained in the manually segmented VMR models. To apply boundary conditions consistently, we generally needed to truncate the MIROS models to terminate at the same approximate location as in the VMR models. We automatically compute the coordinates, radius, and unit tangent vector of each endpoint in each VMR model and then orient and scale clipping boxes to trim the corresponding MIROS model. We also keep the largest contiguous volume, which is then remeshed to produce our desired surface. This workflow is shown in part a-c of Figure 4.4.

Centerline Extraction and Geometric Attribute Calculation

The MIROS and VMR 3D image-based models were used to generate a discrete centerline representation. The centerline extraction was performed using Vascular Modelling Toolkit (VMTK) functions that generated centerlines paths as well as vessel radius information along each vessel path. Both the discrete centerlines paths and associated areas along the paths were required for the ROM flow solver.

Solver

ROM simulation of flow and pressure was performed by SimVascular's 1D and 0D solvers. We applied RCR boundary conditions at all outlets with RCR values tuned to clinical mea-

surements and provided in the VMR repository. In cases where the VMR contained a branch not included in the MIROS model, we ignore that boundary. Although this does not preserve the global consistency of the boundary conditions, we do this to keep the simulation consistent between MIROS and VMR models. The overall workflow that MIROS models undergo is shown in Figure 4.4 for a representative model.

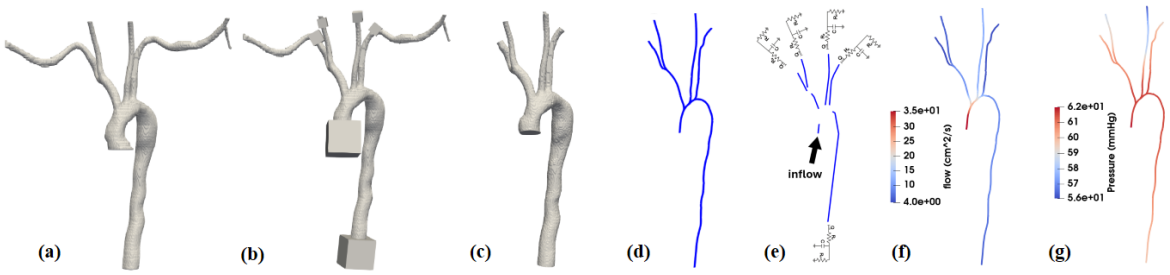


Figure 4.4: Post processing steps, simulation setup and solution; (a) ML based surface (b) boundary clipping (c) trimmed surface (d) centerline extraction (e) 1D model generation with boundary conditions (f)&(g) simulated flow and pressure mapped to centerline

Experimental Setup

Here we describe the datasets and metrics used to evaluate our method. We first evaluate the automatically constructed models geometrically using three different metrics. Secondly, we compare the simulation results of 0D and 1D simulations of manually constructed models with our automatically constructed ones. We use the same boundary conditions in both instances. Finally, we compare the simulation results with ones from a second set of same manually constructed models. These are constructed by a second expert. This analysis gives insight into the variability between different modelers and how the automatically constructed model simulation results compare to it.

Datasets

We train and test the method on a dataset of aortic and aortofemoral vascular models. The data used is from Vascular Model Repository (VMR), see Table 4.1. The nnU-Net neural network used in SeqSeg is trained on 33 CT cases and 37 MR cases. CT and MR data was handled separately, where two different machine learning models were trained to handle each modality independently.

Metrics

We evaluate the geometric models quantitatively by computing the following metrics in comparison to the ‘ground truth‘ manually constructed models; Dice score ($\mathcal{D}$), Hausdorff

Table 4.1: The datasets of patients used for model training and method testing. Abbreviation are as follows: Mod.: Modality. Datasets; VMR: Vascular Model Repository. Anatomy; AO:Aorta, AF:Aortofemoral. Disease; H:Healthy, AAA:Abdominal Aortic Aneurysm, MA: Marfan Syndrome, CA:Coarctation of Aorta, AOD:Aortoiliac Occlusive Disease, SVD:Single Ventricle Defect. Sex; M:Male, F:Female, U:Unknown. Sex and age information was not available for the AVT dataset.

Dataset	Mod.	Train/Test	Anatomy	Disease	Sex	Age(yr)
VMR	CT	33/6	25 AO, 16 AF	23 H, 15 AAA,	23M 6F 3 MA	6 - 80 ave: 58 12U
VMR	MR	37/6	38 AO, 6 AF	19 H, 14 CA, 5 SVD, 2 MA	30M 14F 4 AOD	0.6 - 67 ave: 17

distance score ($\mathcal{H}$), and centerline overlap score ($\mathcal{CO}$). The metrics are defined as follows:

$$\mathcal{D}(X,Y) == \frac{2 \cdot \|X \cap Y\|}{\|X\| + \|Y\|} \tag{4.14}$$

$$\begin{aligned} \mathcal{H}(X,Y) &= \max\{d(X,Y), d(Y,X)\}, \\ \text{where } \; d(X,Y) &= \sup_{x \in X} \inf_{y \in Y} d(x,y) \end{aligned} \tag{4.15}$$

$$\mathcal{CO}(Y, C_t) = \frac{\int_{C_t} Y dx}{\int_{C_t} dx} \tag{4.16}$$

where X,Y are the ground truth and predicted segmentation maps, respectively, and C_t is the ground truth centerline.

We evaluate the simulation results by comparing to the 'ground truth' results from manually contructed models. We calculate error in simulated flow, $\mathcal{E}_Q$, and pressure, $\mathcal{E}_p$, using the average relative absolute difference in value at each vessel cap:

$$\mathcal{E}_Q = \frac{1}{T} \int_T \frac{|Q_T - Q_P|}{Q_T} \tag{4.17}$$

$$\mathcal{E}_p = \frac{1}{T} \int_T \frac{|p_T - p_P|}{p_T} \tag{4.18}$$

where T is total cardiac cycle time, Q_T, p_T are ground truth flow and pressure and Q_P, p_P are predicted flow and pressure values, respectively.

4.4 Results

We present comparison results of reduced order model simulations from 'ground truth' manually constructed geometric models with ones from automatically constructed models (ours). We first, compare the two sets geometrically, using the metrics explained in Sec. 4.3 for a quantitative analysis. Secondly, we compare the simulation results values, flow and pressure, computing errors between the two sets over the cardiac cycle.

Geometric

Geometric quantitative metric results for the test set, CT and MR, are presented in Table 4.2. Dice scores range from 0.881 to 0.965 with an average of 0.944 and 0.918 for CT and MR test data, respectively. Hausdorff distance score was on average 1.54 and 1.947 pixels for CT, MR data, respectively. Finally, centerline overlap scores were on average 0.958 and 0.974 for the CT and MR data, respectively.

For a qualitative comparison, we have the resulting vascular geometric models plotted in Figure 4.6. We show the machine learning based models compared with the 'ground truth' manually constructed ones, both CT and MR test datasets.

Table 4.2: Quantitative metric results for the vascular geometric models; Metrics shown; $\mathcal{D}$: Dice score, $\mathcal{H}$: Hausdorff distance score, $\mathcal{CO}$: centerline overlap score, as defined in Sec. 4.3

	Case	Name	Nr. Branches	$\mathcal{D}\uparrow$	$\mathcal{H}\downarrow$	$\mathcal{CO}\uparrow$
CT	1	0174_0000	5	0.965	1.379	0.916
	2	0176_0000	5	0.948	0.967	0.994
	3	0188_0001_aorta	6	0.952	1.037	0.974
	4	O150323_2009_aorta	5	0.958	0.689	0.988
	5	0139_1001	9	0.925	1.862	0.930
	6	0146_1001	10	0.918	3.307	0.948
	Ave.	-	-	**0.944**	**1.540**	**0.958**
MR	7	0006_0001	9	0.881	1.533	0.975
	8	0063_1001	5	0.874	2.131	0.946
	9	0090_0001	5	0.922	4.070	0.975
	10	0131_0000	5	0.942	2.286	0.990
	11	KDR12_aorta	5	0.929	0.496	0.978
	12	KDR33_aorta	4	0.958	1.167	0.985
	Ave.	-	-	**0.918**	**1.947**	**0.974**

Simulation

Quantative analysis results are presented in Tables 4.3, 4.4 for 0D and 1D simulations, respectively.

For 0D simulations on CT test data, flow and pressure errors were on average 1.1% and 1.0% at vessel caps, respectively. For MR test data, flow and pressure errors were, on average, 2.6% and 2.0% at vessel caps. Maximum errors were higher for flow than pressure, being on average 3.3% and 6.2% compared to 1.2% and 2.8% for CT, MR datasets. For all but four cases, maximum errors were under 5%. Of the four cases with maximum error over 5%, two were CT (cases 3 and 6) and two were MR (cases 9 and 11).

For 1D simulation results on CT test data, flow and pressure errors were on average 0.9% and 0.8%, respectively. Whereas they were higher for MR data, flow and pressure errors were, on average, 8.0% and 3.6%, respectively. Maximum flow and pressure errors were, on average, 3.0% and 1.4% for CT data and 24.2% and 5.1% for MR data. For four cases, maximum errors for either flow or pressure were over 5%, cases 3, 7, 8, 10, three of which were MR.

For further analysis of the error distributions, we plot their box plots in Figure 4.7, for both 0D and 1D results. Note that each model branch cap translates to one data point in Figure 4.7.

For qualitative comparison of the flow and pressure results over cardiac cycle, we plot them for three cases in Figures 4.9,4.11,4.13 for 0D simulation and 4.10,4.12,4.14 for 1D simulation. We chose the three cases that had the highest, median and lowest error values when compared to 'ground truth' simulation results. Figures 4.9,4.10 show results for case 7 (MR), which had highest errors, on average 20.7% for 1D simulation for example. The median case is case 3 (CT), which had an average error of 2.48% and 3.6% for flow in 0D and 1D simulation results, for example. The best case is case 2 (CT), with average flow and errors of 0.1%, 0.07% and 0.27%, 0.21% for 0D and 1D, respectively.

Comparison with Second Observer

Since there exists variability between different modelers, the 'ground truth' geometric models can vary if made by another modeler. Therefore, we include additional analysis of the three cases (worst, median, best) after another expert manually constructed them. We compare the differences betweeen the 0D and 1D simulation results of the original 'ground truth' models with these new 'second observer' constructed models in Figure 4.8, see orange box plots of errors. Figure 4.8 also shows the boxplots of the differences between the original 'ground truth' and our automatically constructed ones, see blue box plots. As can be seen, for 0D results, the errors we calculate for the SeqSeg models fall in the range of modeler variability for flow for the worst and best cases, and for pressure for the median and best cases. For 1D results, they only fall in the range for flow and pressure for the best case.

Inspection of Example Cases

Besides quantitative evaluation, we also qualitatively inspected the worst performing branch in the the least accurate simulation, case ID 7 in Tables 4.3, 4.4. For this branch, the relative errors of flow and pressure in 1D simulation are 73.1% and 8.35%, and the respected errors for 0D simulation are 1.21% and 2.27%. Shown in 4.5, we found that SeqSeg is under-predicting the cross-sectional area while the manual created VMR appears to over-estimate and approximated the branch entirely using circular contours.

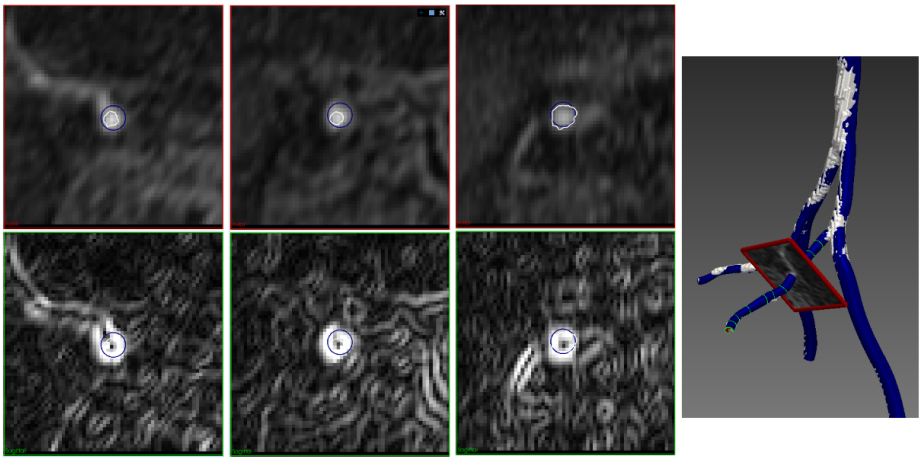


Figure 4.5: Qualitative Inspection of the Worst performing branch; White contour: SeqSeg surface; Blue contour: VMR surface

4.5 Discussion

We present an automatic method to get vascular blood flow reduced order model simulation results, only requiring a seed point placement and boundary condition setting as inputs from a user. The method takes in a medical image scan of patient, CT or MR scan, and a seed point in the vasculature of interest and returns a geometric model using SeqSeg [113], and once boundary conditions have been set at inflow and outlet caps, returns reduced order model simulation results. We demonstrate both 0D and 1D simulation results, for both CT and MR image data as input.

We demonstrate the method by training and testing on CT and MR datasets of aortic and aortofemoral models. We evaluate the geometric models using Dice score, Hausdorff distance and centerline overlap score. All segmentations achieved a Dice score over 0.88 and centerline overlap score over 0.916, meaning over 91.6% of the ground truth centerline was captured. After running 0D and 1D simulations on these automatically constructed models, we compare the results to “ground truth“ results from the ground truth manually constructed

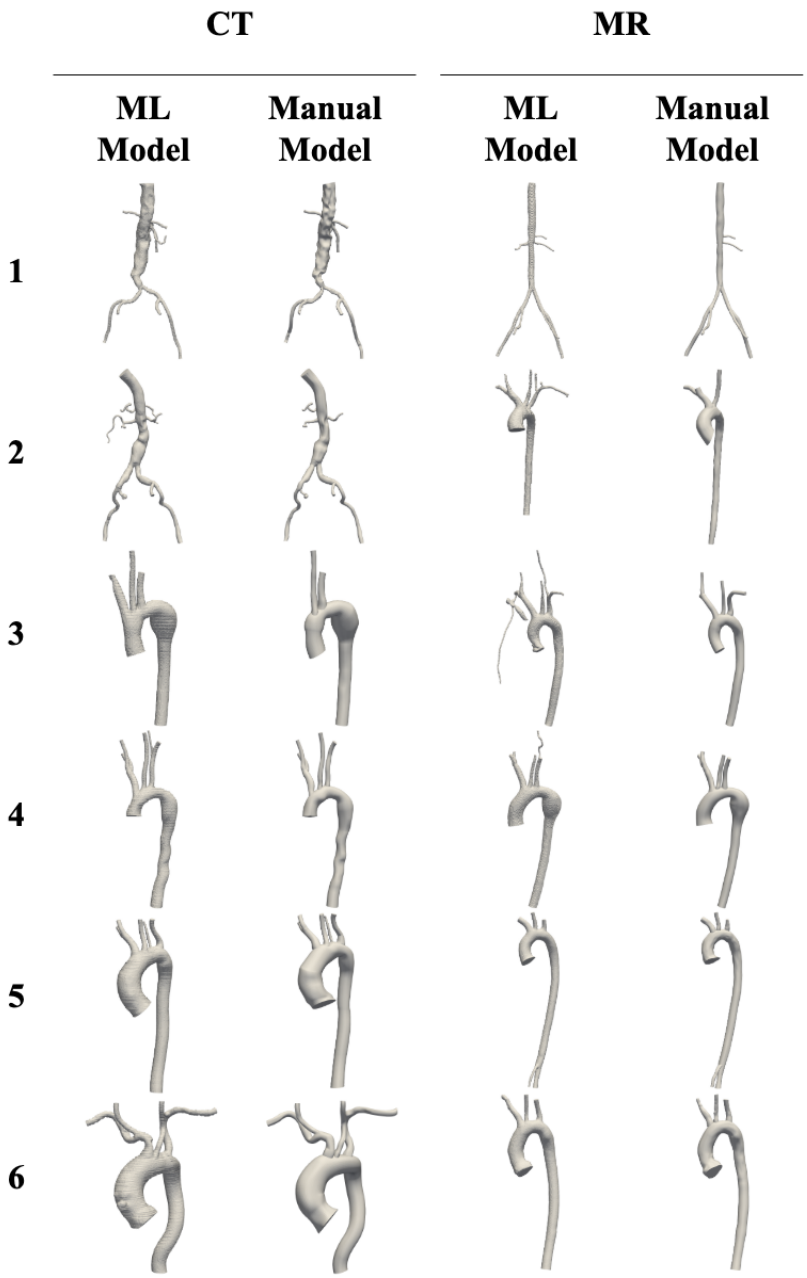


Figure 4.6: Qualitative comparison of the geometric models constructed using machine learning method, SeqSeg, (left) and the manual ground truth model (right), for both CT and MR test data. Numbering is consistent with Tables 4.3,4.4.

Table 4.3: Quantitative error results for 0-D simulation. Error is calculated per branch as time-averaged difference between ground truth and our method at cap; $\mathcal{E}_Q$: flow error, $\mathcal{E}_p$: pressure error. *max* is maximum error, *min* is minimum error and *avg* is the average across all branches.

Mod.	Case	Name	$\mathcal{E}_Q$			$\mathcal{E}_p$		
			min	*max*	*avg*	*min*	*max*	*avg*
CT	1	0174_0000	0.0002	0.0006	0.0003	0.0001	0.0012	0.0006
	2	0176_0000	0.0002	0.0017	0.0010	0.0001	0.0013	0.0007
	3	0188_0001_aorta	0.0154	0.0990	0.0248	0.0167	0.0187	0.0179
	4	O150323_2009_aorta	0.0002	0.0050	0.0015	0.0001	0.0031	0.0014
	5	0139_1001	0.0032	0.0120	0.0059	0.0004	0.0102	0.0048
	6	0146_1001	0.0135	0.0846	0.0308	0.0216	0.0373	0.0333
	Ave.	-	**0.0054**	**0.0338**	**0.0107**	**0.0065**	**0.0120**	**0.0098**
MR	7	0006_0001	0.0063	0.0450	0.0199	0.0071	0.0347	0.0275
	8	0063_1001	0.0103	0.0434	0.0160	0.0136	0.0274	0.0214
	9	0090_0001	0.0324	0.0909	0.0553	0.0020	0.0434	0.0195
	10	0131_0000	0.0006	0.0159	0.0036	0.0002	0.0053	0.0016
	11	KDR12_aorta	0.0374	0.1657	0.0555	0.0346	0.0449	0.0410
	12	KDR33_aorta	0.0008	0.0190	0.0079	0.0097	0.0123	0.0114
	Ave.	-	**0.0147**	**0.0620**	**0.0264**	**0.0112**	**0.0279**	**0.0204**

models. On average, 0D simulation results from machine learning based models differed less than 2.7% in pressure and flow values compared to results from manually constructed "ground truth" models. Similarly for 1D simulation results, calculated flow and pressure values differed, on average, less than 0.87% and 3.6%, respectively.

Treating a manually constructed model as "ground truth" adds analysis uncertainty because of known modeler dependent variability[84, 93]. To investigate the variance in manually constructed geometric models, we compare results from the original test set with ones from three reconstructed models from a "second observer". This second expert constructed new models for the best, median and worst performing cases for comparison. We show that the differences in 0D simulation results between two manually constructed models range up to 6% in flow and 4% in pressure. A similar analysis for 1D simulations shows the differences up to 8% in flow and 3% in pressure. Regarding the worst performing 1D simulation that showed a 73.1% error in flow and 8.35%, upon closer inspection, we found that SeqSeg under-predicts vessel cross-sectional area while VMR over-estimates and only uses circular contours. This would lead to discrepancies in 1D simulation because the 1D solver is highly dependent on cross-sectional area and the sampling locations and methods [94]. Overall, for the best case simulations, both pressure and flow values, and worst case flow values from 0D simulation, the differences between our ML based models and the original test set were within the ranges of the differences between the two manually constructed models.

Table 4.4: Quantitative error results for 1-D simulation. Error is calculated per branch as time-averaged difference between ground truth and our method at cap; $\mathcal{E}_Q$: flow error, $\mathcal{E}_p$: pressure error. *max* is maximum error, *min* is minimum error and *avg* is the average across all branches.

			$\mathcal{E}_Q$			$\mathcal{E}_p$		
Mod.	Case	Name	*min*	*max*	*avg*	*min*	*max*	*avg*
CT	1	0174_0000	0.0002	0.0018	0.0006	0.0004	0.0012	0.0008
	2	0176_0000	0.0010	0.0041	0.0027	0.0003	0.0033	0.0021
	3	0188_0001_aorta	0.0208	0.1442	0.0360	0.0230	0.0465	0.0271
	4	O150323_2009_aorta	0.0003	0.0030	0.0011	0.0008	0.0026	0.0015
	5	0139_1001	0.0008	0.0137	0.0072	0.0068	0.0144	0.0108
	6	0146_1001	0.0010	0.0069	0.0027	0.0008	0.0091	0.0036
	Ave.	-	**0.0046**	**0.0296**	**0.0087**	**0.0056**	**0.0139**	**0.0080**
MR	7	0006_0001	0.0130	0.7314	0.2073	0.0210	0.1137	0.0672
	8	0090_0001	0.0431	0.1255	0.0736	0.0332	0.0488	0.0403
	9	0131_0000	0.0010	0.0219	0.0064	0.0004	0.0062	0.0031
	10	KDR12_aorta	0.0667	0.2990	0.1006	0.0028	0.0797	0.0654
	11	KDR33_aorta	0.0025	0.0299	0.0119	0.0005	0.0072	0.0037
	Ave.	-	**0.0253**	**0.2415**	**0.0800**	**0.0116**	**0.0505**	**0.0359**

By combining the low computation cost of reduced order models with rapid geometric model construction using machine learning methods, we hope this method facilitates entry of hemodynamic simulations into the clinical setting. The lack of automated model construction methods is one of main factors hindering large patient cohort research studies and clinical trials necessary for patient applications [3]. Furthermore, the lack of fully integrated workflows going from medical image scan to simulation results further impedes research and clinical applications because of their non trivial inconvenient setup. This work aims to address this issue by an almost fully automated pipeline for hemodynamic reduced order modeling.

Future directions of this work include further testing and validation. A main limitation of the work presented is the small test set. Additionally, further comparison with other manually constructed models is needed. In general, the field of patient-specific hemodynamic modeling lacks a comprehensive study of modeler variability in geometric model construction and its effect on downstream simulation results. This would allow for insightful analysis of automated model construction methods with clear benchmarks of manual "gold standard" model construction. Furthermore, we present a general comparison of flow and pressure values calculated from patient-specific reduced order models but for clinical applications, more specific investigations must be done. Many hemodynamic simulation applications focus on specific mechanical properties at particular locations, e.g. pressure drop across coronary artery stenosis or wall shear stress inside aneurysms, making validation of automated work-

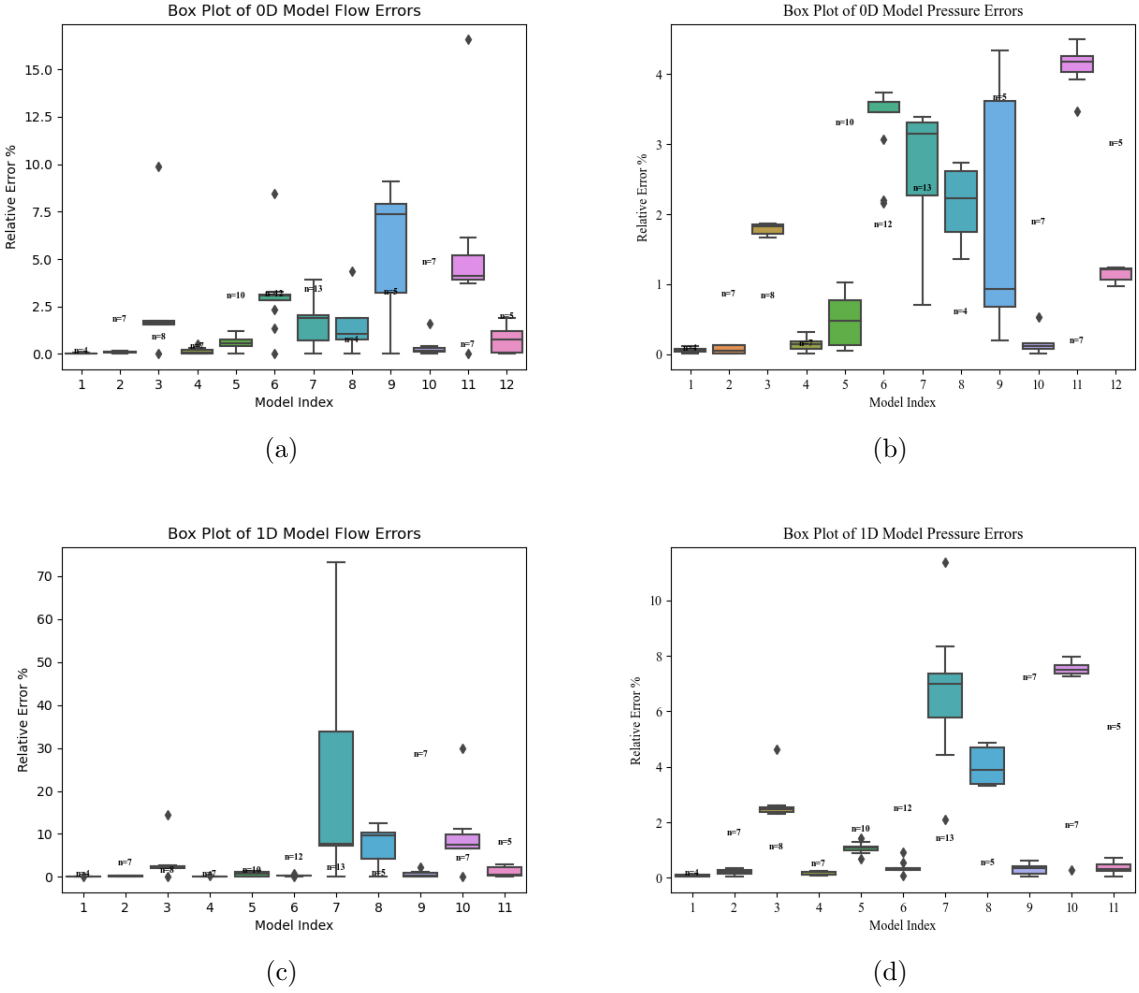


Figure 4.7: Error results for test data; 0D reduced order model errors (top row) and 1D reduced order model error (bottom row). (a), (c) show error in calculated flow and (b), (d) show error in calculated pressure. Model indices are consistent with Tables 4.3,4.4 and Figure 4.6.

flows, such as the one presented, for those specifics necessary.

Automated boundary condition setting was not explored in this work but is a potential area of open research. In the presented workflow, the user must input seed points for initialization and define boundary conditions at inflow and outlet boundaries. Development of methods to facilitate boundary condition choice and tuning would enable further automation of the workflow.

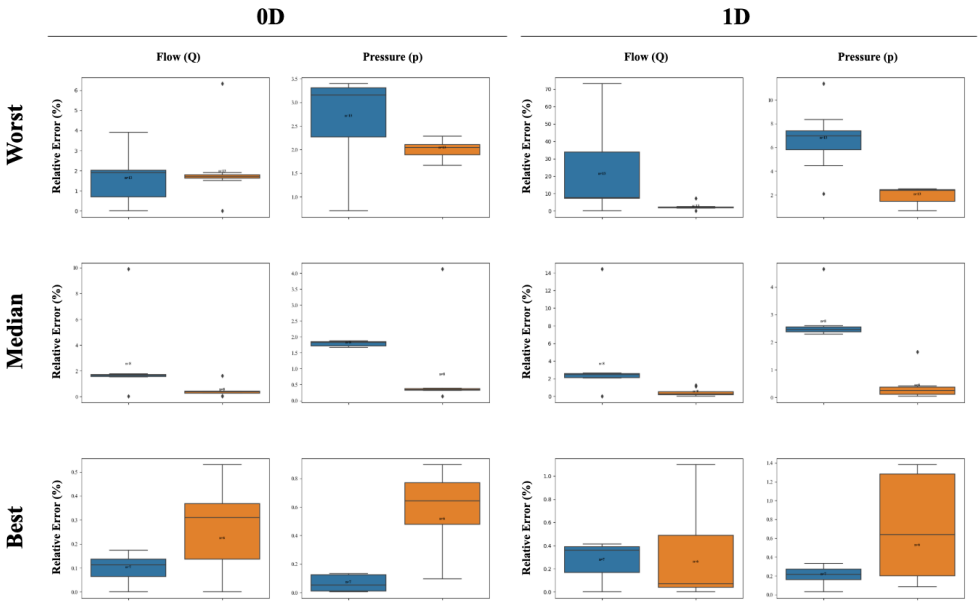
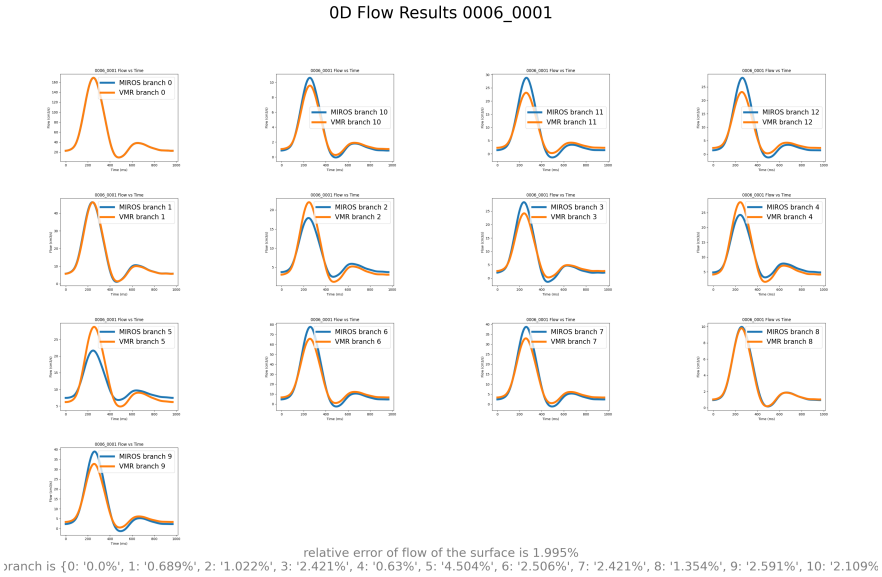
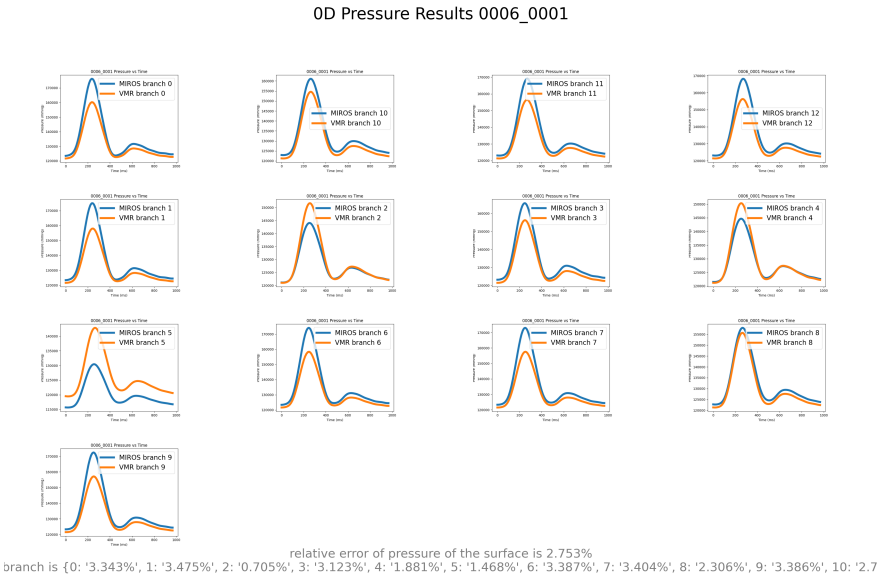


Figure 4.8: Comparing resulting errors of machine learning based constructed models (blue) with errors between two manually constructed models (orange) for the cases that had worst (MR case 1), median (CT case 3), and best (CT case 2) errors. Resulting errors are shown for 0D (left) and 1D (right) simulations, for both flow and pressure values.

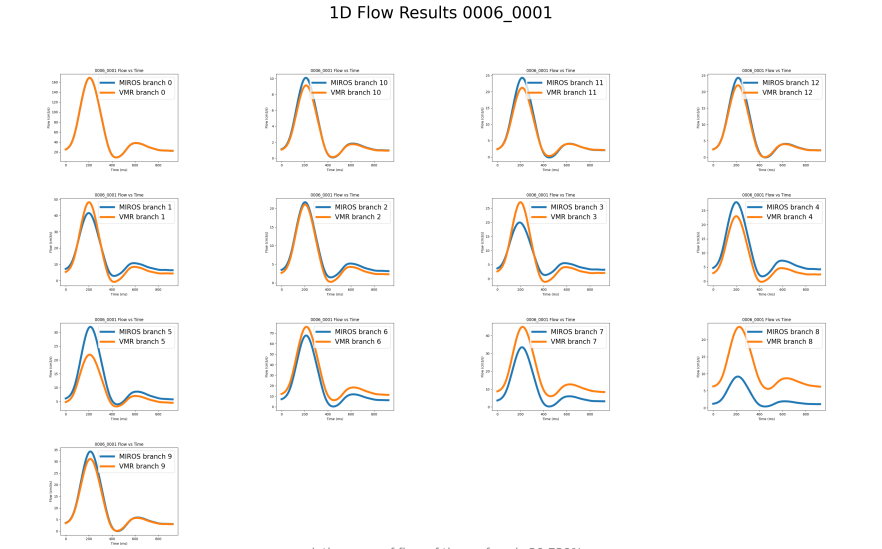


(a)

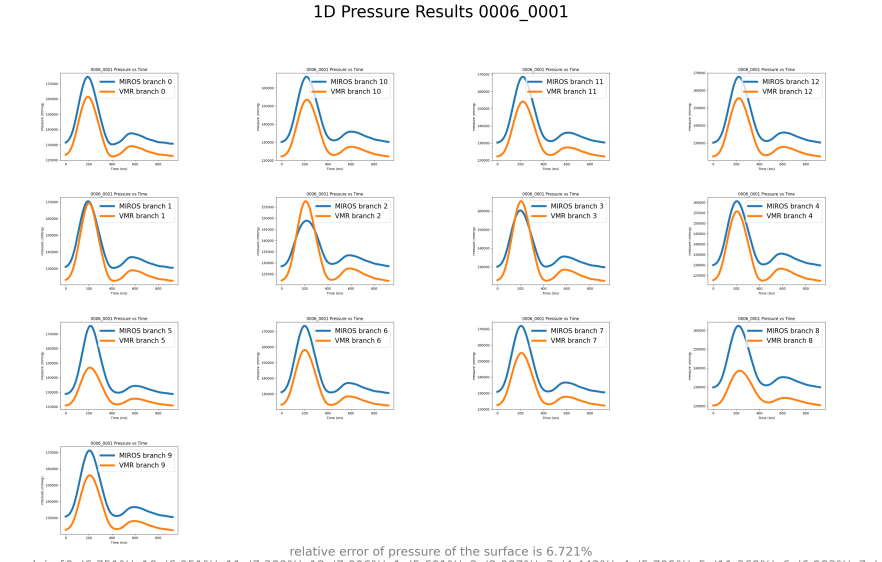


(b)

Figure 4.9: 0D flow and pressure plots at all outlets for model 0006_0001, index 1 in Tables 4.3, 4.4, for which errors where the highest. Comparing the machine learning based constructed model with a manually constructed one (ground truth).

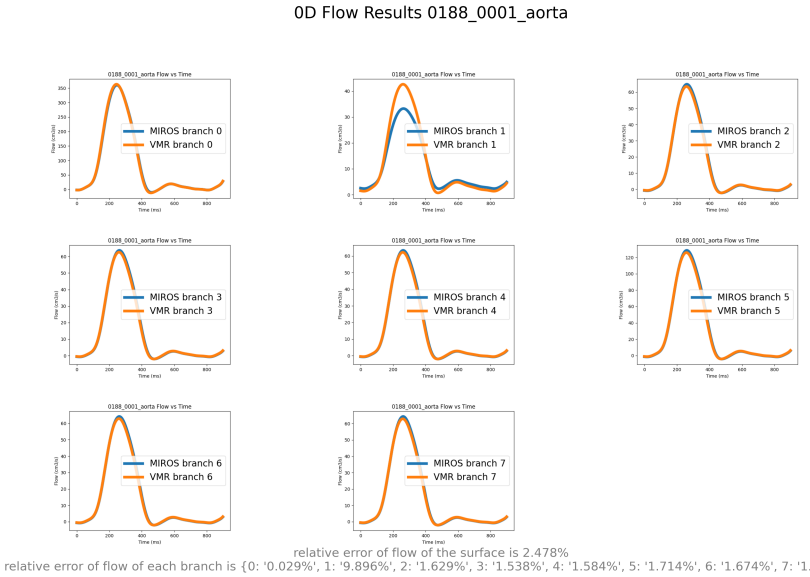


(a)

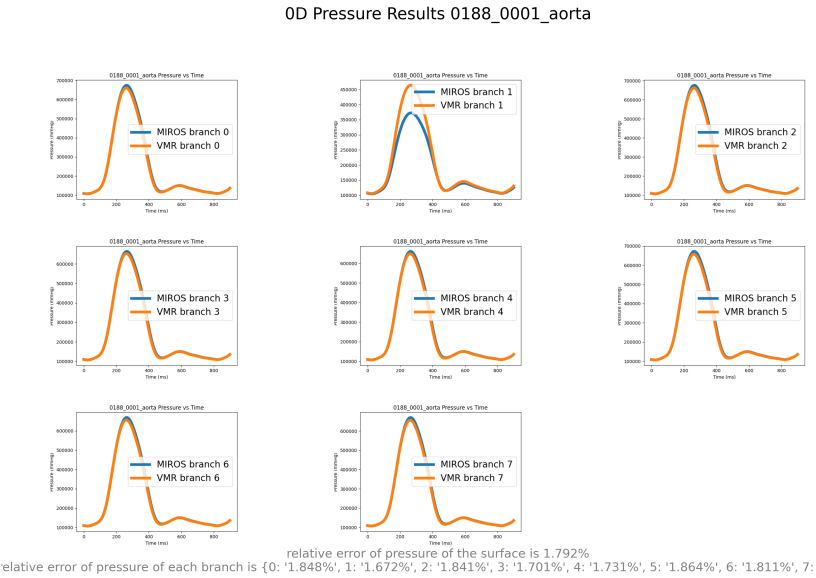


(b)

Figure 4.10: 1D flow and pressure plots at all outlets for model 0006_0001, index 1 in Tables 4.3, 4.4, for which errors where the highest. Comparing the machine learning based constructed model with a manually constructed one (ground truth).

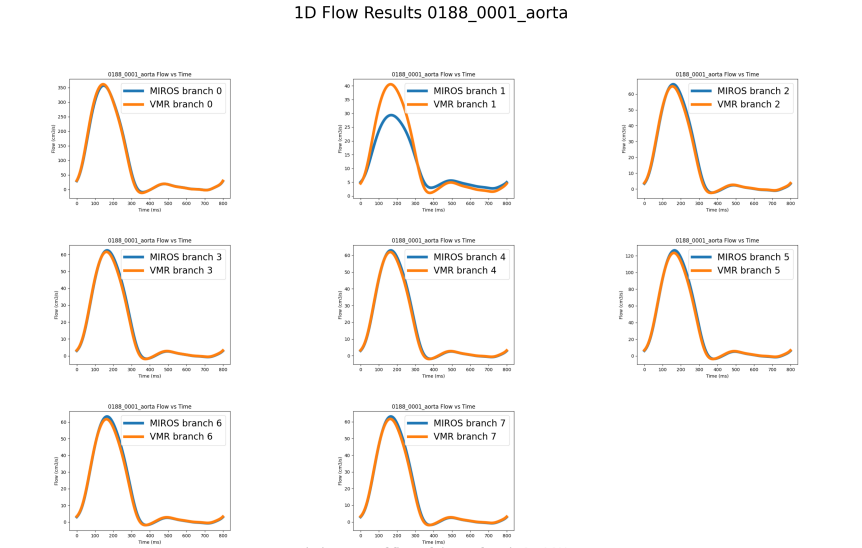


(a)

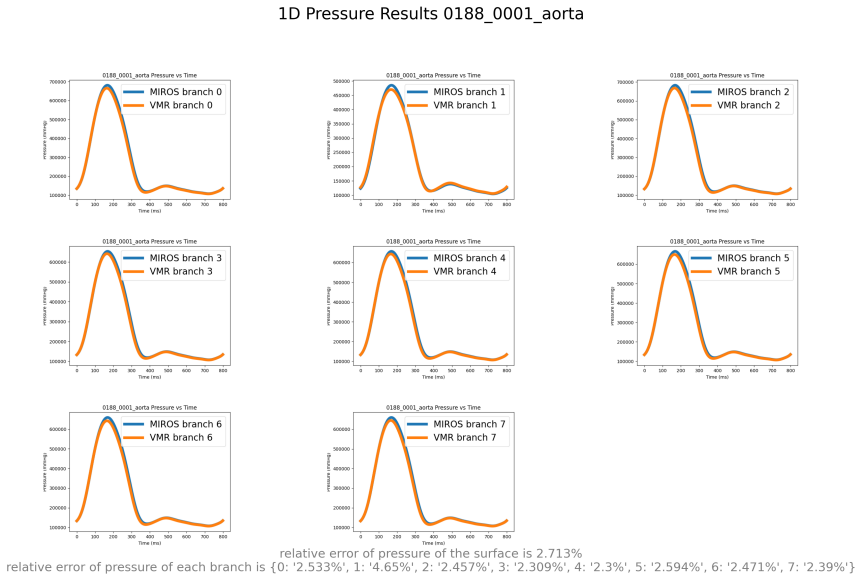


(b)

Figure 4.11: 0D flow and pressure plots at all outlets for model 0188_0001_aorta, index 3 in Tables 4.3, 4.4, for which errors where the median for all models. Comparing the machine learning based constructed model with a manually constructed one (ground truth).



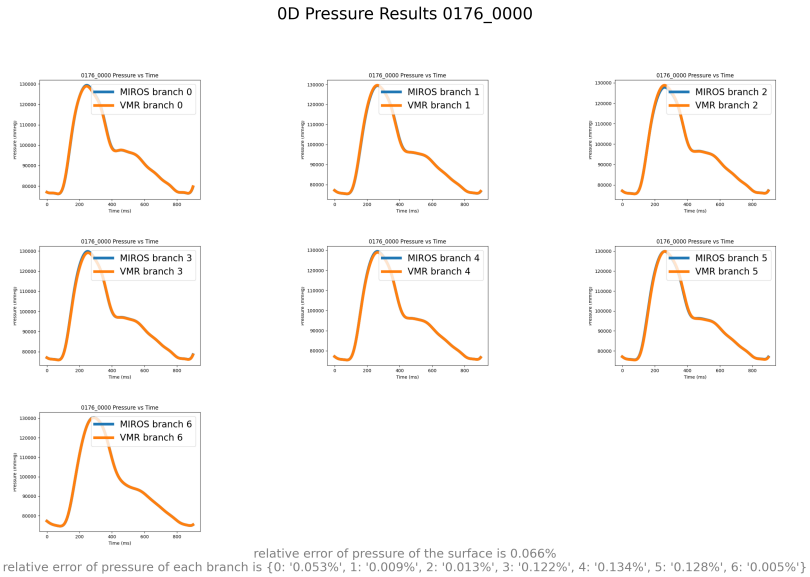
(a)



(b)

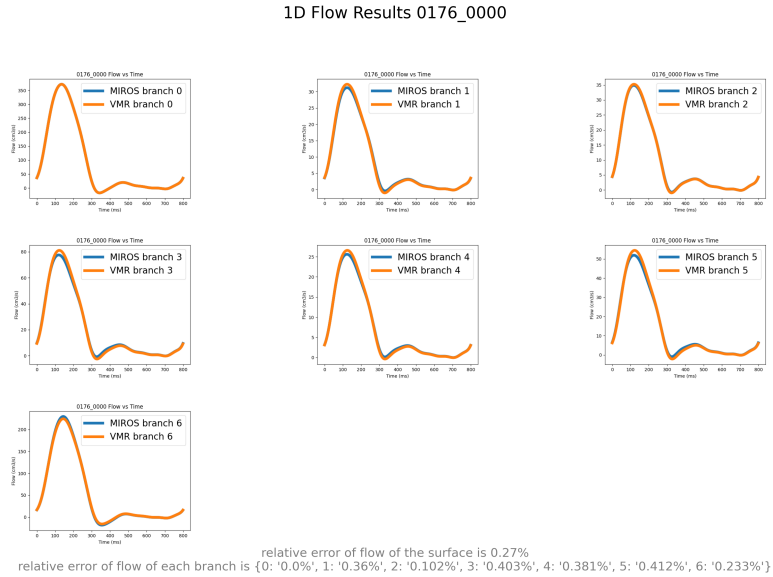
Figure 4.12: 1D flow and pressure plots at all outlets for model 0188_0001_aorta, index 3 in Tables 4.3, 4.4, for which errors where the median for all models. Comparing the machine learning based constructed model with VMR, a manually constructed one (ground truth).

(a)

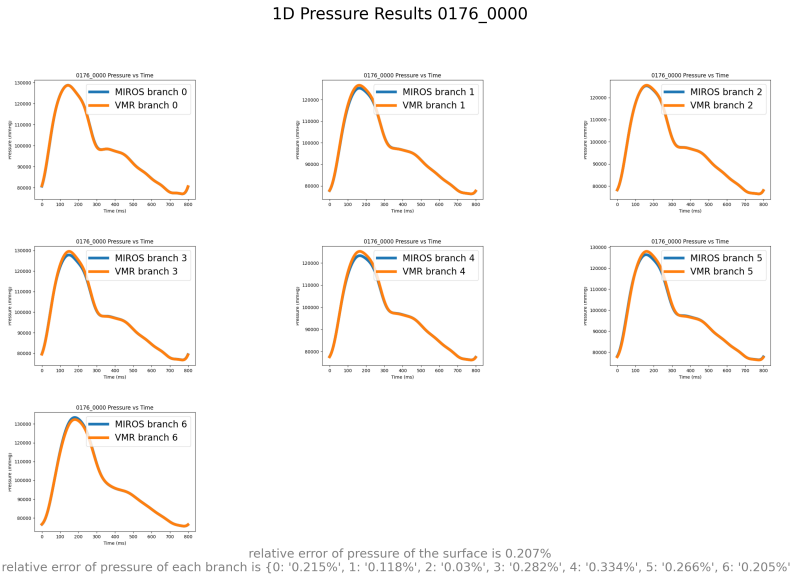


(b)

Figure 4.13: 0D flow and pressure plots at all outlets for model 0176_0000, index 2 in Tables 4.3, 4.4, for which errors where the lowest for all models. Comparing the machine learning based constructed model with a manually constructed one (ground truth).



(a)



(b)

Figure 4.14: 1D flow and pressure plots at all outlets for model 0176_0000, index 2 in Tables 4.3, 4.4, for which errors where the lowest for all models. Comparing the machine learning based constructed model with VMR, a manually constructed one (ground truth).

Chapter 5

VesselTrajNet: A Goal-Driven Approach for Vasculature Prediction in Medical Imaging

5.1 Abstract

Vasculature tracking is a fundamental problem in medical image analysis with critical applications in disease diagnosis, surgical planning, and interventional procedures. This task shares conceptual similarities with human trajectory prediction, where motion history and environmental context inform future path estimation. Inspired by advancements in trajectory forecasting, we propose a novel approach that adapts a U-Net-based Gaussian heat map encoder-decoder architecture, originally developed for human motion prediction, to the problem of vessel tracking in medical images. By predicting multiple goals and trajectories, the approach naturally captures vessel branching patterns without requiring explicit bifurcation detection. We evaluate our method, VesselTrajNet, on datasets of coronary artery CT scans, demonstrating its effectiveness in accurately reconstructing vascular structures. Our results suggest that trajectory-based modeling provides a promising framework for automated vasculature tracking, with potential applications in diagnostic and interventional imaging.

5.2 Introduction

Blood vessel reconstruction from volumetric medical imaging is crucial for cardiovascular healthcare, with applications in diagnostics, treatment planning, and research [55]. A key aspect is path line construction (or centerline extraction), which outlines a graph or "skeletonization" of the vascular network. This path line often captures essential geometric and topological information and is typically a prerequisite for full 3D vascular modeling. Despite

its significance, automated and accurate vascular path line construction remains an open challenge in medical imaging [83].

The challenges in vascular path line construction from volumetric image data stem from several factors. Low image resolution, unclear lumen boundaries, medical image artifacts and inconsistent dye distribution in angiography all contribute to making this a difficult task.

Automated methods developed for path line extraction generally fall into two categories: (1) 3D volumetric segmentation or centerline prediction and (2) iterative tracking. The former analyzes the entire image volume but is computationally expensive and often produces disconnected segments [127]. In contrast, iterative tracking predicts vessel progression locally, using subvolume image features to determine direction and bifurcations. Machine learning methods, such as CNN-based orientation classification [127] and graph neural networks [2], have shown promise. However, these approaches often struggle with bifurcation detection, as they rely on sequential "next direction" predictions.

Vessel tracking shares key similarities with trajectory prediction in other fields. Human motion prediction, widely studied in computer vision and robotics [103], is essential for intelligent agents like autonomous vehicles to navigate safely. These algorithms predict future human movements by analyzing past trajectories and optional environmental inputs like RGB images, segmentation, or encoded features such as lanes and traffic lights. Over the past decades, data-driven machine learning approaches have tackled the challenge of forecasting accurate trajectories while accounting for interactions, surroundings, and the multimodal nature of human behavior—where multiple future paths are possible. Recent advancements have significantly improved performance, including the approach of predicting multiple goals first and conditioning trajectory predictions on these goals [70], and using a heat map representation of trajectories to help neural networks reason about trajectory history within an environment [69].

Building on advances in human motion prediction, we propose a goal-oriented tracking approach for vascular structures. Just as humans may follow a single path or choose from multiple possible paths to different destinations based on their environment, vessels can continue as a single path or bifurcate into multiple branches. Instead of predicting the next direction, we first predict multiple likely "goals" or endpoints for the trajectory. Based on these endpoints, we generate one trajectory per goal. This framework naturally handles bifurcations and more complex branching structures.

We present VesselTrajNet, a novel goal-oriented automated vascular tracking method. We validate our approach using a dataset of coronary CT scans, demonstrating superior performance compared to baseline methods. We also show our method's ability to automatically predict multiple trajectories for sub-volumes containing bifurcations. Most importantly, this work, which to our knowledge is the first one to apply insights in human trajectory prediction to vessel tracking, bridges two distinct research domains and reveals their underlying synergies.

5.3 Related Work

Vasculature Tracking Methods Recursive vessel tracking typically combines seed point placement with a growth method that analyzes local vessel features to determine direction. Early model-based approaches assume vessels as cylindrical segments fitted to image data [12][29]. Other methods use minimal cost path optimization such as [19] though these often require knowledge of start and end points. More recently, convolutional neural networks (CNNs) have shown promise for vessel segmentation and centerline prediction [23][83][46]. [113] proposed tracking vessels using local 3D segmentations with post-processing for path extraction, while [127] trained a CNN-based orientation classifier on local image volumes. Extending this, [2] introduced a graph neural network for orientation prediction.

Computer Vision Trajectory Prediction For autonomous vehicles, forecasting the future positions of road agents is a critical problem for safe and successful operation [40],[17]. Many of these approaches incorporate environmental context but rely highly structured nature of the driving problem and incorporate parameterized scene elements such as lanes and traffic lights [18],[106],[88],[108]. In contrast, human trajectory predictors [103] cannot rely on structured environments with explicit lanes and traffic lights. Earlier approaches did not incorporate the environment and relied on constant velocity models [8], or the past motion history alone [44] [70]. A critical innovation from [70] proposed first predicting a distribution of trajectory endpoints, or goals, and then conditioning the full trajectory prediction on the goals. More recent work incorporates the environment through RGB scene images, originally as neural network features [49], [71]. A more recent approach, YNet [69], proposes that neural networks can better combine image features with trajectory coordinates when projecting the trajectories into Gausian heat maps of the same dimensions as the scene segmentation. This approach, which also uses goal-conditioned trajectory prediction [70], results in successful long-range predictions, and recent works have built on it to tackle even more challenging, unstructured environments [120].

5.4 Methodology

The vascular tracking task is formulated as a trajectory forecasting problem; given a medical image volume and a history of past n_p locations (in a sequential order), predict the n_f future locations. Therefore, the prediction is conditioned on two factors: the scene information (i.e. image data values) and the history of prior points (i.e. the x, y, z coordinates). We take a goal-driven approach to the problem. A goal is predicted first and then a trajectory conditioned on the goal is produced. Both the goal and trajectory predictions have past trajectory and scene information encoded.

Our method takes in a 3D medical image sub volume containing a blood vessel segment and n_p past path points as an ordered list of 3D coordinates. However, we do future trajectory prediction in 2D and therefore require to extract 2D slices of the image volume. We define

the slicing planes using the tangent of the past path line points because we assume the next points in the trajectory will be in (or close to) that plane. We define the planes such that they intersect the last past path line point and have normals orthogonal to the tangent. Because there exist infinitely many vectors orthogonal to the tangent in 3D, we set a parameter n_s for the number of planes to define and run prediction on. We take slices defined by different angle θ in the cross sectional plane, see Figure 5.1 (a). Sufficient number of slices must be extracted to ensure bifurcations are detected during prediction.

Past path line points are projected onto slicing planes and resampled to obtain n_p past points. Figure 5.1 (b) illustrates vessel segmentation sliced along tangent, normal, and bi-normal vectors computed from path line points, while (c) shows the corresponding raw image slices with projected points. The image volume is also rotated so that the x-axis in 2D slices aligns with the tangent (Figure 5.1 (c)), simplifying learning as trajectories consistently move from "left" to "right." All 2D slices are resampled to a fixed 400×400 size, with x, y coordinates scaled and shifted to a local coordinate system ranging from 0 to 400.

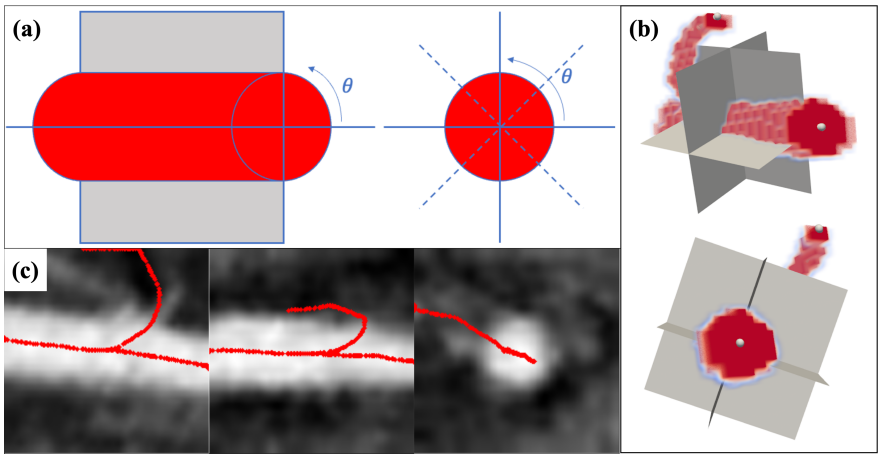


Figure 5.1: (a) 2D image slices taken using the tangent and center point for plane definition, multiple planes defined by varying θ. (b) 3D vascular segmentation (red), ground truth path line points and the image planes defined for 2D slicing. (c) The resulting 2D raw images with path line points projected onto the planes, plane defined by path line normal (left), plane defined by path line bi-normal (center), and cross sectional plane defined by the path line tangent (right).

The vascular segmentation and past points are fed into our model, which uses the YNet architecture proposed in [69]. The past trajectory points are represented as n_p separate Gaussian heat maps. The heat maps are concatenated with the vascular segmentation and fed into a U-Net style encoder, $E_{S,P}$. The scene and history encoded features are fed into two decoder networks a goal decoder network D_G and future trajectory decoder network D_F. Features at different resolutions are fed into the goal decoder network, which then

outputs a goal probability map. The probability distribution is sampled and represented as a Gaussian heat map (similar to the history trajectory). This heat map is then resampled and fed into the trajectory decoder at different stages along with the output features from encoder $E_{S,P}$. The output of the trajectory decoder is a n_f channel tensor containing probability maps of corresponding future point locations. The point coordinates are determined using a `softargmax` operation. The overall model architecture is shown in Figure 5.2. Multiple goals can be sampled, such as for bifurcation detection. A parameter n_g is set for the number of goals to be predicted. In that case, a trajectory is determined independently for each goal using D_F. Furthermore, we deploy a `Test Time Sampling Trick (TTST)` which clusters the goal probability distribution and returns their centers as the n_g goals. Additionally during inference, we deploy `Conditional Waypoint Sampling (CWS)` which favors trajectory predictions close to waypoint samples, for further details see [69].

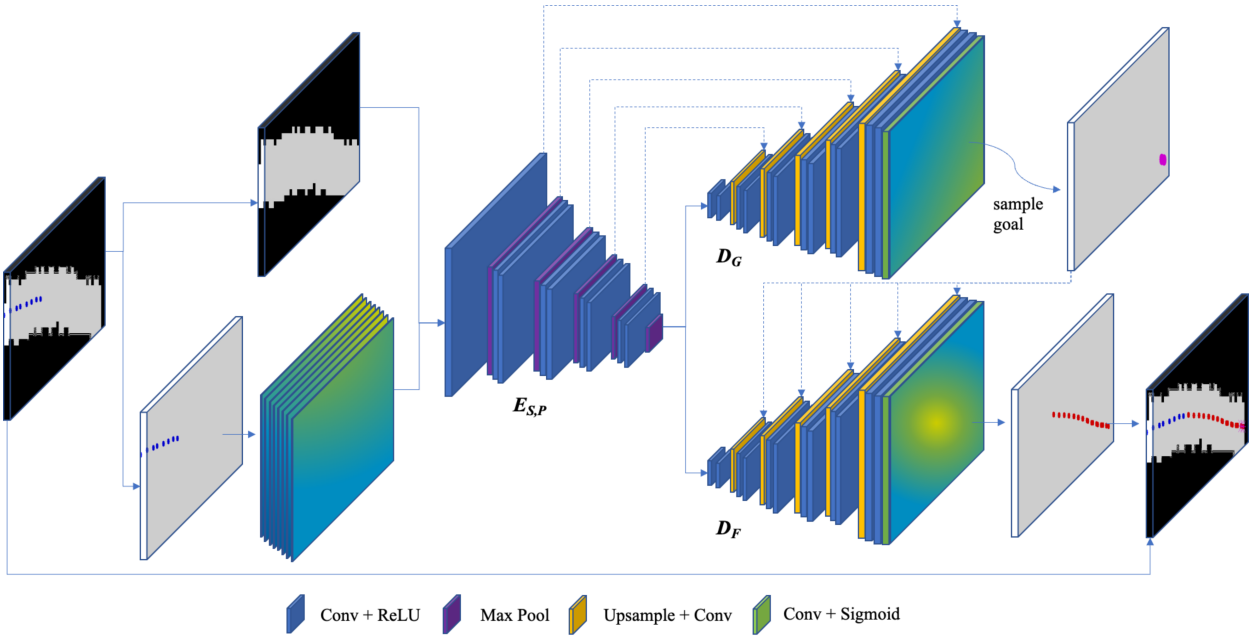


Figure 5.2: Model architecture. The input is a sequence of n_p past points and a vessel segmentation, both of which are encoded using the $E_{S,P}$ encoder. The encoded features are passed to the goal decoder, D_G, at different resolutions. The output goal probability map is sampled and fed into the trajectory decoder, D_F, which produces n_f heat maps that are converted to points by a `softargmax` operation.

During inference, we predict $n_g = 2$ trajectories on $n_s = 10$ planes defined by the path line tangent, resulting in $n_g \times n_s$trajectories per subvolume. Predictions are merged if their goals overlap within a threshold $t = 50$; otherwise, a bifurcation is detected, and the trajectory from the bifurcation plane is used. If no bifurcation is found, trajectories are averaged in 3D space, returning a single path.

For training we require a dataset of annotated 3D image volumes and corresponding path lines. We extract 2D slices using the ground truth tangent in the center of the volume. We resample the path line points to be a total of 20 and project them onto the image planes, 8 are used for history and 12 for future prediction. If there is a bifurcation, each trajectory is processed independently. By taking multiple planes per volume, we get a larger dataset of 2D images with corresponding trajectory points. We use a binary cross entropy loss with Adam optimizer and learning rate of 0.0001. Training involves 300 epochs using an NVIDIA Geforce RTX 2080ti GPU (11 GB GPU memory) on the Savio High Performance Computing cluster at the University of California, Berkeley. Validation metrics (described below) are used for checkpoint choice where training/validation split was 80/20% of trajectories.

5.5 Experimental Setup

We test our method on the ASOCA dataset of coronary artery CT scans with annotated path lines and segmentation masks [31]. A total of 40 cases are used, 30 for training and 10 of which are kept aside for testing.

For evaluation, we use Average Displacement Error (ADE) and Final Displacement Error (FDE), measured on future predicted trajectories against ground truth. ADE is the L_2 distance between all predicted and ground truth trajectory points, while FDE is the L_2 distance between the final predicted and ground truth points. Min k ADE/FDE is computed by sampling k trajectories and reporting the best ADE/FDE.

We compare to several baseline methods:

Random prediction Sample n_f random points in the image coordinate system ($x, y \in [0, 400)$).

K **nearest neighbor ($K \in$ {1,3,10})** Find the nearest neighbor(s) in the training set for the past trajectories and return the corresponding future, for $K > 1$ we average the future trajectories.

Constant velocity Calculate the velocity vector between the last two points in history and recursively add to predict next n_f points.

Multilayer perceptron (MLP) neural network Train a 2 hidden layer MLP (dimensions $(100, 100)$) on the training dataset trajectories.

All of our baseline methods only take the past history as input, and are not conditioned on the image. We use ADE and FDE to evaluate our method and compare it to baseline methods.

5.6 Results

Quantitative metric results with corresponding baseline values for ASOCA datset are shown in Table 5.1. Random prediction gets an ADE score of 140.6, which can be viewed as an upper bound. A constant velocity prediction achieves 35.7, much better than random generation,

Table 5.1: Results from quantitative comparison between our method and several baselines on ASOCA dataset test trajectories. For ADE_{min}, FDE_{min}: $n_g = 5$

Method	ADE (pixels)	FDE (pixels)	ADE_{min} (5)	FDE_{min} (5)
Random	140.60	162.72	-	-
Constant Velocity	35.74	58.90	-	-
1 Nearest Neighbor	32.82	55.07	-	-
3 Nearest Neighbor	26.78	44.46	-	-
10 Nearest Neighbor	24.62	40.40	-	-
MLP (2 HL, HD=100)	23.21	38.72	-	-
Our w/o TTST, CWS	18.63	32.84	12.64	14.93
Our w/ TTST, CWS	**16.13**	**24.09**	**11.48**	**10.74**

and K-NN further outperforms with a minimum ADE of 24.6, for $K = 10$. Finally, a two hidden layer MLP network obtains the best metric results of all baselines, a resulting ADE value of 23.2. On average, our method obtains ADE metric scores of 18.6 with a single goal prediction. However, when we allow for $n_g = 5$ goals, we get a minimum ADE score of 12.6. Furthermore by using `TTST`,`CWS` during inference, we get 16.1 for ADE for single goal prediction and a minimum of 11.5 for $n_g = 5$ goals. The FDE results follow a similar trend and in particular improves by approximately 8 when adding `TTST` and `CWS` during inference. When compared to our baselines, our method outperforms all of them, improving on the best baseline by approximately 30.5% in ADE and 37.8% in FDE score.

Figure 5.3 shows examples of predicted vascular trajectories using two goal prediction, with corresponding ground truth for comparison. We show examples for high, median and low metric scores. Figure 5.3 (top row, second from right) shows that sometimes when the score computed with these metrics indicates a high error, the trajectory may still be consistent with the image.

To evaluate bifurcation detection, we present qualitative examples in Figure 5.4. Each case includes six 2D image planes (raw and segmentation masks) with corresponding past and predicted future trajectories, highlighting where bifurcations were detected. For 3D assessment, Figure 5.4 also displays the predicted 3D trajectory overlaid on the raw volumetric image data as an isosurface. During training, the model learns a goal probability map capable of representing multiple paths simultaneously, effectively capturing bifurcations

A limitation of the 2D image slice approach described here is the dependence on 1) a well defined tangent and 2) multiple image plane slices for bifurcation detection. This leads to projected points and information loss which necessitates further post processing to aggregate the multiple image plane predictions into one. This is particularly difficult when discerning bifurcations from faulty trajectory predictions. To tackle highly branched vasculature, we aim to investigate a 3D prediction framework instead. This would allow for direct bifurcation prediction using a single forward pass on the 3D image volume directly, using multiple goals similar the proposed 2D method.

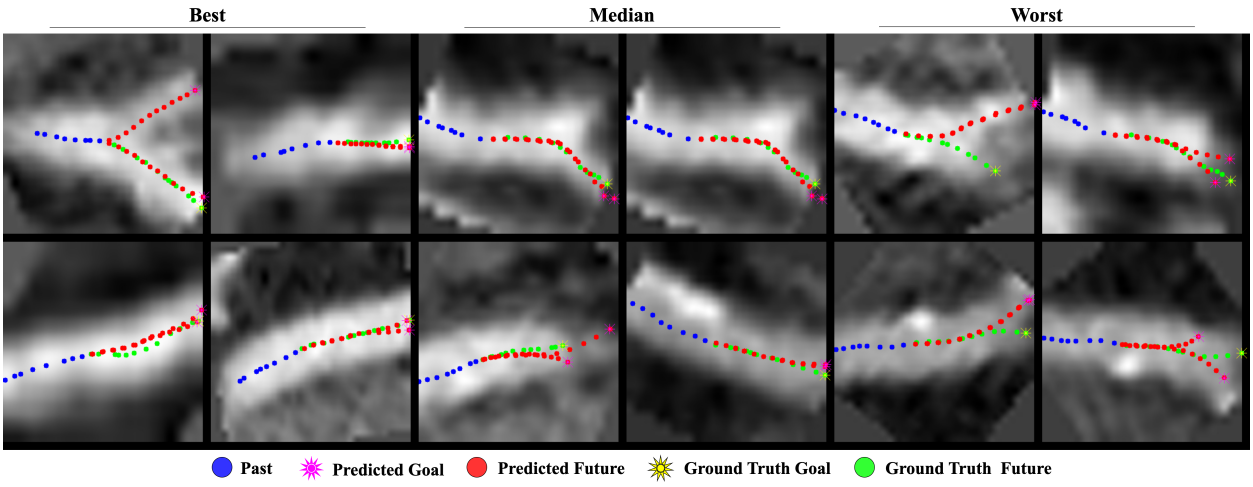


Figure 5.3: Two Goal Prediction. 2D trajectory prediction results overlaid in raw image data, comparing with ground truth future and goals.

5.7 Conclusion

Despite its broad applicability in cardiovascular healthcare, automated vascular path line construction remains unsolved. We introduce a goal-oriented deep learning tracking approach, adapted from human motion prediction, and validate it on coronary angiography data. By leveraging similarities between these tasks, we highlight the potential for cross-domain technique transfer.

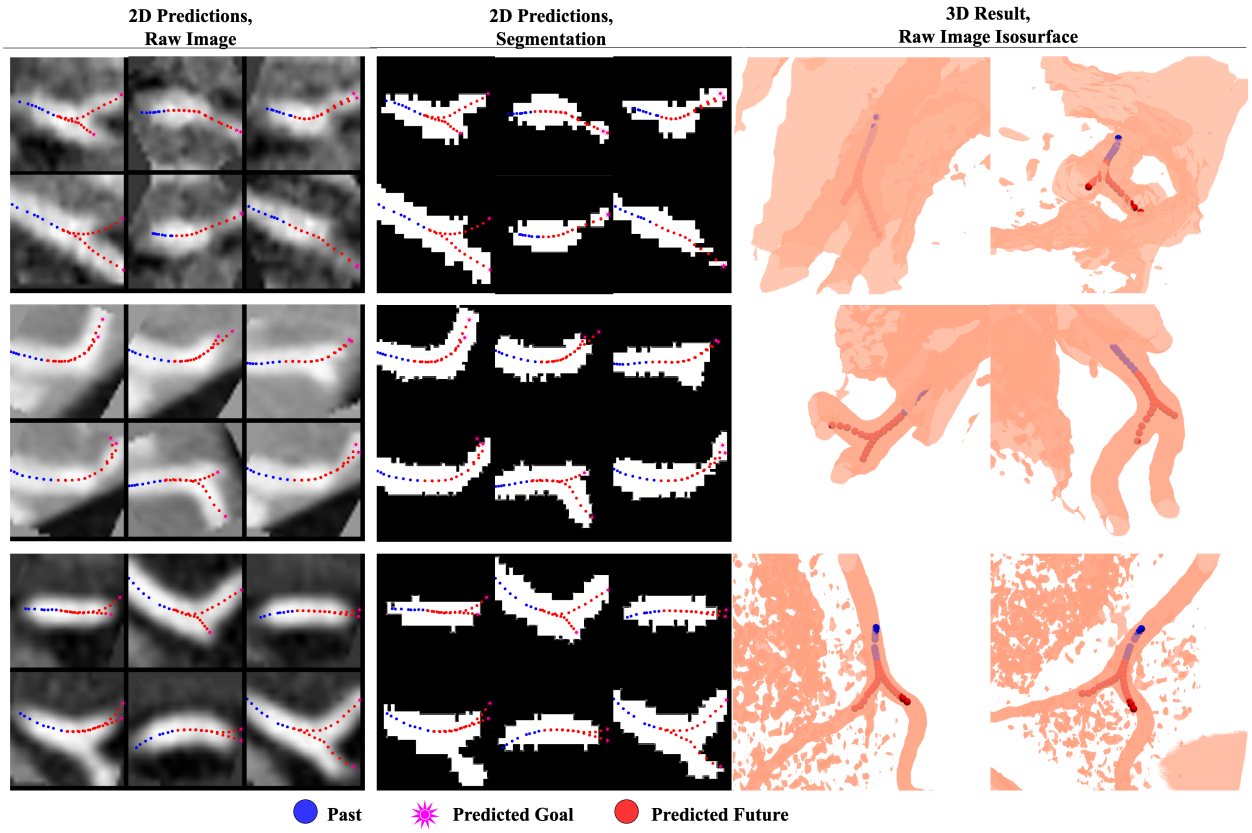


Figure 5.4: Bifurcation prediction. History (blue) with predicted future trajectories (red) using two goals (pink). Results are overlaid on raw image data (left) and vascular segmentations (center), and after being post processed and visualized in 3D (right).

Chapter 6

Conclusion

Patient-specific computational modeling of cardiovascular function is a powerful tool for the diagnosis, understanding, and treatment of cardiovascular disease. However, current workflows are hindered by bottlenecks in constructing patient-specific geometries from medical imaging, which limits scalability to large patient cohorts and delays clinical translation.

In this thesis, we developed deep learning–based tools to automate the construction of patient-specific cardiovascular models. Namely, in Chapter 2 we introduce SeqSeg, an automatic vascular tracking and segmentation method designed to address the time-consuming and resource-intensive workflows that currently constrain large-scale studies and broader clinical adoption. Requiring only a single seed point for initialization, SeqSeg enables rapid, fully automated vascular model construction from CT or MR imaging data. Within minutes, the method can produce a segmentation, surface mesh, and vessel centerline of the targeted vasculature. We further demonstrate that SeqSeg not only outperforms existing state-of-the-art vessel segmentation methods but also generalizes to vascular structures not present in the training data. To facilitate accessibility and adoption, SeqSeg is released as open-source software, with support for all major operating systems, including macOS, Windows, and Ubuntu.

Many computational cardiovascular applications require integrated geometric models that include both the cardiac chambers and major vascular branches, along with clearly defined valve interfaces. In Chapter 3, we build upon the automated vascular modeling framework by adding LinFlo-Net, a method for fully automated construction of comprehensive cardiovascular models from medical imaging. To our knowledge, this represents the first fully automatic approach capable of generating such models. We evaluate the method on a dataset of CT images, demonstrating its effectiveness. By employing a dual modeling strategy, template-based for cardiac structures and growth-based for vascular branches, we highlight the value of anatomy-specific modeling approaches tailored to different components of the cardiovascular system.

Workflows for patient-specific, physics-based vascular modeling typically face two major bottlenecks: the construction of geometric models from medical imaging and the computational cost of running simulations with numerical solvers. In Chapter 4, we introduce MIROS

(Medical Image to Reduced-Order Simulation), a framework that combines the automated, rapid model construction of SeqSeg with the computational efficiency of reduced-order hemodynamic models in SimVascular to enable patient-specific vascular simulations in minutes. We demonstrate that, in most cases considered, simulation results based on SeqSeg-derived models fall within the inter-observer variability of simulations based on manually constructed geometries. MIROS is evaluated on both MR and CT datasets of the aorta and aortofemoral vasculature. By significantly reducing both modeling and simulation time, MIROS represents a step toward integrating hemodynamic modeling into clinical workflows for personalized diagnosis and treatment planning.

Finally, in Chapter 5, we build upon SeqSeg to explore a novel approach for vascular tracking that focuses on predicting vessel direction and identifying bifurcations. While SeqSeg relies on pixel-wise classification for segmentation, its performance in detecting bifurcations can degrade in regions of low contrast or image artifacts. To address this limitation, we draw inspiration from recent advances in human trajectory prediction and introduce VesselTrajNet, a goal-oriented vascular tracking method. By framing tracking as a multi-goal prediction task, VesselTrajNet inherently handles bifurcations without requiring explicit detection, offering potential improved robustness in challenging imaging scenarios.

Patient-specific, physics-based computational modeling of the cardiovascular system holds significant promise for advancing the diagnosis, investigation, and treatment of cardiovascular disease. What was once confined to academic research and theoretical modeling is now beginning to enter clinical practice, with demonstrated benefits for patient care[30]. Still, a major bottleneck preventing broader clinical adoption is the time-consuming process of constructing anatomical models from medical imaging. This thesis addresses this challenge by developing deep learning–based methods to automate its pipeline.

6.1 Future directions

This work represents a significant step toward the automatic construction of cardiovascular geometries using deep learning. However, the problem remains far from fully solved. In this final section, I outline several future directions that have the potential to build upon the contributions presented here and further advance the integration of cardiovascular simulations into clinical practice.

Improving segmentation performance The primary contribution of this thesis, SeqSeg, relies on neural network–based vascular segmentation. The current implementation employs a U-Net architecture trained from randomly initialized parameters. However, there is considerable room for improvement in the local segmentation component of SeqSeg.

First, alternative network architectures can be explored. Deep learning–based image segmentation has advanced significantly in recent years, resulting in a range of architectures that may outperform the current model. These include Feature Pyramid Networks (FPN)[60], Pyramid Scene Parsing Network (PSPNet)[132], and nnU-Net[41], as well as attention-based

models such as UNETR[90], TransUNet[14], Swin-UNETR[36], MISSFormer[39], and Attention U-Net[91]. These architectures remain to be evaluated for improving segmentation accuracy on local vascular segments.

Second, segmentation performance could potentially benefit from transfer learning, in which model training is initialized with pretrained weights obtained from related tasks or datasets. Several studies have demonstrated that transfer learning can yield substantial performance gains in medical image segmentation[114, 61, 15, 98, 129, 75].

Finally, current segmentation outputs are constrained by the resolution of the input imaging data. However, this limitation is not intrinsic to the task. Future work could explore predicting segmentations at a higher resolution than the input, which may reduce voxel-level artifacts and yield more precise vessel boundary predictions.

Mesh post processing The primary goal of this work is to generate anatomically accurate, patient-specific cardiovascular meshes suitable for physics-based simulations. However, the current approach produces meshes that exhibit staircase artifacts, a consequence of the underlying voxel-based pixel classification task. Developing improved post-processing methods for mesh extraction, such as refining the outputs of algorithms like marching cubes[64], remains an open area for enhancement.

Going beyond pixel classification: signed distance fields, point clouds and meshes Reformulating the task to move beyond voxel-wise classification presents a promising direction. Future work could explore training deep learning models to directly predict surface meshes[59, 100, 124], point clouds[131, 62], or implicit representations such as signed distance fields[27, 10, 128]. These alternative representations have the potential to eliminate voxel-induced artifacts and yield smoother, more geometrically faithful vascular models.

Integration with manual editing tools One key advantage of manual vascular model construction is the flexibility it offers for making targeted edits. Given that automated methods are unlikely to achieve perfect anatomical accuracy in all cases, particularly in challenging regions, post hoc manual correction remains an important part of the modeling pipeline. Integrating the automated methods developed in this thesis with existing manual editing tools, such as those available in SimVascular, would significantly enhance their practical utility. Such integration would not only support the creation of more anatomically accurate models but also enable efficient correction of mislabeled training data. In turn, this would lead to improved performance and generalizability of the deep learning models by ensuring higher-quality supervision during training.

Utilizing deep learning to accelerate training data annotation A key limitation of data-driven methods for vascular segmentation is the reliance on high-quality annotated training data. For vascular model construction, this typically involves manual delineation using tools such as SimVascular, CRIMSON, or similar software platforms. These manual

workflows are time-intensive, often requiring hours or even days per case, which limits the scale of most training datasets to just a few dozen patients. In addition, the acquisition of medical imaging data is often restricted due to privacy regulations and institutional barriers, further complicating the curation of robust training datasets.

However, the automated methods presented in this thesis may themselves serve as tools to accelerate the generation of annotated data. By integrating these methods into existing manual editing workflows, users could start from automated segmentations and perform only minor corrections, rather than segmenting from scratch. This semi-automated approach has been shown to substantially reduce annotation time. For example, recent work using deep learning–assisted labeling frameworks such as MONAI Label, DeepEdit, and HAL-IA has demonstrated annotation time reductions of 50–80% for pixel-wise segmentation tasks[21, 20, 56]. These tools could be extended to vascular model annotation, facilitating the creation of larger and more diverse training datasets.

Automated vascular tracking In Chapter 5, we introduce VesselTrajNet, a deep learning–based model capable of automatically tracking vasculature. While its performance was evaluated on standalone trajectory prediction tasks, its integration into SeqSeg for local trajectory prediction has not yet been explored. Due to its goal-oriented formulation, VesselTrajNet has the potential to outperform SeqSeg in challenging regions characterized by low contrast, imaging artifacts, or uneven contrast distributions.

There are also opportunities to further improve the method. For instance, the current implementation operates in 2D, but extending it to 3D volumetric prediction remains an open direction for future research. Additionally, SeqSeg could benefit from incorporating existing automatic tracking techniques, which have not yet been systematically evaluated in this context. Methods such as [127, 105, 119, 34] could be adapted to enhance vessel trajectory inference and model robustness.

Other medical imaging domains: ultrasound and time-series data This thesis has primarily focused on static CT and MR imaging data. However, vascular geometry is often dynamic, changing over time due to physiological processes such as cardiac motion (e.g., deformation of the coronary arteries) or pulsatile blood flow (e.g., expansion and contraction of the aorta). Modeling vasculature across time using time-series imaging data remains a largely open research area, though some work has been done for cardiac modeling[51]. This task is significantly more complex due to the requirement for temporal consistency of mesh indices across frames, which is critical for simulations and longitudinal analysis.

In addition, CT and MR imaging are resource-intensive and not universally accessible. These modalities are predominantly available in well-resourced, high-income settings. To extend the benefits of cardiovascular modeling to lower-cost and under-resourced regions, it is essential to explore more accessible imaging modalities, such as ultrasound. However, 3D geometric modeling from ultrasound presents substantial challenges due to its primarily 2D acquisition, low signal-to-noise ratio, and operator variability. Emerging deep learning

methods and recent advances in 2D-to-3D reconstruction from the field of computer vision may offer promising solutions for generating anatomically accurate vascular models from ultrasound data[130, 26].

Bibliography

[1] Samaneh Abbasi-Sureshjani et al. "Automatic detection of vascular bifurcations and crossings in retinal images using orientation scores". In: *Proceedings - International Symposium on Biomedical Imaging*. Vol. 2016-June. Washington D.C.: IEEE Computer Society, June 2016, pp. 189–192. ISBN: 9781479923502. DOI: 10.1109/ISBI.2016.7493241.

[2] Dieuwertje Alblas et al. "SIRE: Scale-invariant, rotation-equivariant estimation of artery orientations using graph neural networks". In: *Medical Image Analysis* 101 (Apr. 2025), p. 103467. ISSN: 13618415. DOI: 10.1016/j.media.2025.103467. URL: https://linkinghub.elsevier.com/retrieve/pii/S1361841525000155.

[3] Luca Antiga et al. "An image-based modeling framework for patient-specific computational hemodynamics". In: *Medical & Biological Engineering & Computing* 46.11 (2008), pp. 1097–1112. ISSN: 1741-0444. DOI: 10.1007/s11517-008-0420-1. URL: https://doi.org/10.1007/s11517-008-0420-1.

[4] Michela Antonelli et al. "The Medical Segmentation Decathlon". In: *Nature Communications* 13.1 (Dec. 2022). ISSN: 20411723. DOI: 10.1038/s41467-022-30695-9.

[5] Christopher J. Arthurs et al. "CRIMSON: An open-source software framework for cardiovascular integrated modelling and simulation". In: *PLoS Computational Biology* 17.5 (May 2021). ISSN: 15537358. DOI: 10.1371/journal.pcbi.1008881.

[6] Amirhossein Arzani et al. "Effect of exercise on patient specific abdominal aortic aneurysm flow topology and mixing". In: *International Journal for Numerical Methods in Biomedical Engineering* 30.2 (Feb. 2014), pp. 280–295. ISSN: 20407939. DOI: 10.1002/cnm.2601.

[7] Kathrin Bäumler et al. "Fluid–structure interaction simulations of patient-specific aortic dissection". In: *Biomechanics and Modeling in Mechanobiology* 19.5 (Oct. 2020), pp. 1607–1628. ISSN: 16177940. DOI: 10.1007/s10237-020-01294-8.

[8] R A Best and J P Norton. "A new model and efficient tracker for a target with curvilinear motion". In: *IEEE Transactions on Aerospace and Electronic Systems* 33.3 (1997), pp. 1030–1037. DOI: 10.1109/7.599328.

[9] Pranav Bhagirath et al. "From bits to bedside: entering the age of digital twins in cardiac electrophysiology". In: *Europace* 26.12 (Dec. 2024). ISSN: 1099-5129. DOI: 10.1093/europace/euae295.

[10] Lea Bogensperger et al. "FlowSDF: Flow Matching for Medical Image Segmentation Using Distance Transforms". In: *arXiv preprint arXiv:2405.18087* (Feb. 2025). URL: http://arxiv.org/abs/2405.18087.

[11] Patrick M Boyle, Juan Carlos del Álamo, and Nazem Akoum. "Fibrosis, atrial fibrillation and stroke: clinical updates and emerging mechanistic models". In: *Heart* 107.2 (Jan. 2021), pp. 99–105. ISSN: 1355-6037. DOI: 10.1136/heartjnl-2020-317455.

[12] Juan F. Carrillo et al. "Recursive tracking of vascular tree axes in 3D medical images". In: *International Journal of Computer Assisted Radiology and Surgery* 1.6 (2007), pp. 331–339. ISSN: 18616429. DOI: 10.1007/s11548-007-0068-6.

[13] Bin Chen et al. "Automatic segmentation of pulmonary blood vessels and nodules based on local intensity structure analysis and surface propagation in 3D chest CT images". In: *International Journal of Computer Assisted Radiology and Surgery* 7.3 (2012), pp. 465–482. ISSN: 18616429. DOI: 10.1007/s11548-011-0638-5.

[14] Jieneng Chen et al. "TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation". In: *arXiv preprint arXiv: 2102.04306* (). URL: https://github.com/Beckschen/.

[15] Sihong Chen, Kai Ma, and Yefeng Zheng. "Med3D: Transfer Learning for 3D Medical Image Analysis". In: *arXiv preprint arXiv:1904.00625* (Apr. 2019). URL: http://arxiv.org/abs/1904.00625.

[16] Paolo Cignoni et al. "MeshLab: an Open-Source Mesh Processing Tool". In: *Eurographics Italian Chapter Conference*. Ed. by Vittorio Scarano, Rosario De Chiara, and Ugo Erra. The Eurographics Association, 2008. ISBN: 978-3-905673-68-5.

[17] Can Cui et al. "A survey on multimodal large language models for autonomous driving". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024, pp. 958–979.

[18] Henggang Cui et al. "Multimodal Trajectory Predictions for Autonomous Driving using Deep Convolutional Networks". In: *2019 International Conference on Robotics and Automation (ICRA)*. Montreal, QC, Canada, 2019, pp. 2090–2096. DOI: 10.1109/ICRA.2019.8793868.

[19] Thomas Deschamps and Laurent D Cohen. "Fast extraction of minimal paths in 3D images and applications to virtual endoscopy". In: *Medical Image Analysis* 5 (2001), pp. 281–299. DOI: 10.1016/S1361-8415(01)00046-9.

[20] Andres Diaz-Pinto et al. "DeepEdit: Deep Editable Learning for Interactive Segmentation of 3D Medical Images". In: *Data Augmentation, Labelling, and Imperfections*. Springer Nature Switzerland, 2022, pp. 11–21. DOI: 10.1007/978-3-031-17027-0{_}2.

[21] Andres Diaz-Pinto et al. "MONAI Label: A framework for AI-assisted interactive labeling of 3D medical images". In: *Medical Image Analysis* 95 (July 2024). ISSN: 13618423. DOI: 10.1016/j.media.2024.103207.

[22] Siamak N. Doost et al. "Heart blood flow simulation: A perspective review". In: *BioMedical Engineering Online* 15.1 (Aug. 2016). ISSN: 1475925X. DOI: 10.1186/s12938-016-0224-8.

[23] Alexandru Dorobantiu, Valentin Ogrean, and Remus Brad. "Coronary centerline extraction from ccta using 3d-unet". In: *Future Internet* 13.4 (2021). ISSN: 19995903. DOI: 10.3390/fi13040101.

[24] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *arXiv preprint arXiv: 2010.11929* (2021). URL: https://arxiv.org/abs/2010.11929.

[25] Pamela S Douglas et al. "1-Year Outcomes of FFR CT-Guided Care in Patients With Suspected Coronary Disease The PLATFORM Study". In: 68.5 (Aug. 2016).

[26] Mark C. Eid, Ana I. L. Namburete, and João F. Henriques. "UltraGauss: Ultrafast Gaussian Reconstruction of 3D Ultrasound Volumes". In: *arXiv preprint arXiv: 2505.05643* (May 2025). URL: http://arxiv.org/abs/2505.05643.

[27] Salvatore Esposito et al. "VesselSDF: Distance Field Priors for Vascular Network Reconstruction". In: *arXiv preprint arXiv:2506.16556* (June 2025). URL: http://arxiv.org/abs/2506.16556.

[28] Andriy Fedorov et al. "3D Slicer as an image computing platform for the Quantitative Imaging Network". In: *Magnetic Resonance Imaging* 30.9 (Nov. 2012), pp. 1323–1341. ISSN: 0730725X. DOI: 10.1016/j.mri.2012.05.001.

[29] Ola Friman et al. "Multiple hypothesis template tracking of small 3D vessel structures". In: *Medical Image Analysis* 14.2 (Apr. 2010), pp. 160–171. ISSN: 13618415. DOI: 10.1016/j.media.2009.12.003.

[30] Sara Gaur et al. "Rationale and design of the HeartFlowNXT (HeartFlow analysis of coronary blood flow using CT angiography: NeXt sTeps) study". In: *Journal of Cardiovascular Computed Tomography* 7.5 (Sept. 2013), pp. 279–288. ISSN: 19345925. DOI: 10.1016/j.jcct.2013.09.003.

[31] Ramtin Gharleghi et al. "Automated segmentation of normal and diseased coronary arteries – The ASOCA challenge". In: *Computerized Medical Imaging and Graphics* 97 (Apr. 2022). ISSN: 18790771. DOI: 10.1016/j.compmedimag.2022.102049.

[32] Richard A. Gray and Pras Pathmanathan. "Patient-Specific Cardiovascular Computational Modeling: Diversity of Personalization and Challenges". In: *Journal of Cardiovascular Translational Research* 11.2 (Apr. 2018), pp. 80–88. ISSN: 1937-5387. DOI: 10.1007/s12265-018-9792-2.

[33] Viatcheslav Gurev et al. "Models of cardiac electromechanics based on individual hearts imaging data: Image-based electromechanical models of the heart". In: *Biomechanics and Modeling in Mechanobiology* 10.3 (2011), pp. 295–306. ISSN: 16822978. DOI: 10.1007/s10237-010-0235-5.

[34] Tao Han et al. "Ordered multi-path propagation for vessel centerline extraction". In: *Physics in Medicine and Biology* 66.15 (Aug. 2021). ISSN: 13616560. DOI: 10.1088/1361-6560/ac0d8e.

[35] M S Hassouna and A A Farag. "Robust centerline extraction framework using level sets". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. 2005, pp. 458–465. DOI: 10.1109/CVPR.2005.306.

[36] Ali Hatamizadeh et al. "Swin UNETR: Swin Transformers for Semantic Segmentation of Brain Tumors in MRI Images". In: *arXiv preprint arXiv:2201.01266* (Jan. 2022). URL: http://arxiv.org/abs/2201.01266.

[37] Ali Hatamizadeh et al. "UNetFormer: A Unified Vision Transformer Model and Pre-Training Framework for 3D Medical Image Segmentation". In: *arXiv preprint arXiv:2204.00631* (Apr. 2022). URL: http://arxiv.org/abs/2204.00631.

[38] Xiaolong Hu, Liejun Wang, and Yongming Li. "HT-Net: A Hybrid Transformer Network for Fundus Vessel Segmentation". In: *Sensors* 22.18 (Sept. 2022). ISSN: 14248220. DOI: 10.3390/s22186782.

[39] Xiaohong Huang et al. "MISSFormer: An Effective Transformer for 2D Medical Image Segmentation". In: *IEEE Transactions on Medical Imaging* 42.5 (May 2023), pp. 1484–1494. ISSN: 1558254X. DOI: 10.1109/TMI.2022.3230943.

[40] Yanjun Huang et al. "A survey on trajectory-prediction methods for autonomous driving". In: *IEEE Transactions on Intelligent Vehicles* 7.3 (2022), pp. 652–674.

[41] Fabian Isensee and Klaus H. Maier-Hein. "An attempt at beating the 3D U-Net". In: *arXiv preprint arXiv:1908.02182* (Aug. 2019). URL: http://arxiv.org/abs/1908.02182.

[42] Fabian Isensee et al. "nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation". In: *Nature Methods* 18.2 (Feb. 2021), pp. 203–211. ISSN: 15487105. DOI: 10.1038/s41592-020-01008-z.

[43] Lucian Itu et al. "Non-invasive hemodynamic assessment of aortic coarctation: Validation with in vivo measurements". In: *Annals of Biomedical Engineering* 41.4 (Apr. 2013), pp. 669–681. ISSN: 00906964. DOI: 10.1007/s10439-012-0715-0.

[44] Boris Ivanovic and Marco Pavone. "The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 2375–2384.

[45] Richard Izzo et al. "The Vascular Modeling Toolkit: A Python Library for the Analysis of Tubular Structures in Medical Images". In: *Journal of Open Source Software* 3.25 (May 2018), p. 745. DOI: 10.21105/joss.00745.

[46] Dengqiang Jia and Xiahai Zhuang. "Learning-based algorithms for vessel tracking: A review". In: *Computerized Medical Imaging and Graphics* 89 (Apr. 2021). ISSN: 18790771. DOI: 10.1016/j.compmedimag.2020.101840.

[47] Mithun Kumar Kar, Debanga Raj Neog, and Malaya Kumar Nath. "Retinal Vessel Segmentation Using Multi-Scale Residual Convolutional Neural Network (MSR-Net) Combined with Generative Adversarial Networks". In: *Circuits, Systems, and Signal Processing* (2022). ISSN: 15315878. DOI: 10.1007/s00034-022-02190-5.

[48] Rashed Karim et al. "Algorithms for left atrial wall segmentation and thickness – Evaluation on an open-source CT and MRI image database". In: *Medical Image Analysis* 50 (Dec. 2018), pp. 36–53. ISSN: 13618423. DOI: 10.1016/j.media.2018.08.004.

[49] Kris M Kitani et al. "Activity forecasting". In: *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part IV 12*. 2012, pp. 201–214.

[50] Fanwei Kong and Shawn C. Shadden. "Automating Model Generation for Image-Based Cardiac Flow Simulation". In: *Journal of Biomechanical Engineering* 142.11 (Nov. 2020). ISSN: 15288943. DOI: 10.1115/1.4048032.

[51] Fanwei Kong and Shawn C. Shadden. "Learning Whole Heart Mesh Generation From Patient Images for Computational Simulations". In: *IEEE Transactions on Medical Imaging* 42.2 (Feb. 2023), pp. 533–545. ISSN: 1558254X. DOI: 10.1109/TMI.2022.3219284.

[52] Fanwei Kong, Nathan Wilson, and Shawn Shadden. "A deep-learning approach for direct whole-heart mesh reconstruction". In: *Medical Image Analysis* 74 (Dec. 2021). ISSN: 13618423. DOI: 10.1016/j.media.2021.102222.

[53] Hongzhi Lan et al. "A Re-Engineered Software Interface and Workflow for the Open-Source SimVascular Cardiovascular Modeling Package". In: *Journal of Biomechanical Engineering* 140.2 (2018), pp. 1–11. ISSN: 15288951. DOI: 10.1115/1.4038751.

[54] Byoung Kwon Lee. "Computational fluid dynamics in cardiovascular disease". In: *Korean Circulation Journal* 41.8 (2011), pp. 423–430. ISSN: 17385555. DOI: 10.4070/kcj.2011.41.8.423.

[55] David Lesage et al. "A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes". In: *Medical Image Analysis* 13.6 (Dec. 2009), pp. 819–845. ISSN: 13618415. DOI: 10.1016/j.media.2009.07.011.

[56] Xiaokang Li et al. "HAL-IA: A Hybrid Active Learning framework using Interactive Annotation for medical image segmentation". In: *Medical Image Analysis* 88 (Aug. 2023). ISSN: 13618423. DOI: 10.1016/j.media.2023.102862.

[57] Yuxin Li et al. "VBNet: An end-to-end 3D neural network for vessel bifurcation point detection in mesoscopic brain images". In: *Computer Methods and Programs in Biomedicine* 214 (Feb. 2022). ISSN: 18727565. DOI: 10.1016/j.cmpb.2021.106567.

[58] Zhuowei Li et al. "A Deep Reinforced Tree-Traversal Agent for Coronary Artery Centerline Extraction". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 12905 LNCS. Springer Science and Business Media Deutschland GmbH, 2021, pp. 418–428. ISBN: 9783030872397. DOI: 10.1007/978-3-030-87240-3{_}40.

[59] Fengming Lin et al. "From Pixels to Polygons: A Survey of Deep Learning Approaches for Medical Image-to-Mesh Reconstruction". In: *arXiv preprint arXiv:2505.03599* (May 2025). URL: http://arxiv.org/abs/2505.03599.

[60] Tsung Yi Lin et al. "Feature pyramid networks for object detection". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2017-January. IEEE, Nov. 2017, pp. 936–944. ISBN: 9781538604571. DOI: 10.1109/CVPR.2017.106.

[61] Geert Litjens et al. "A survey on deep learning in medical image analysis". In: *Medical Image Analysis* 42 (Dec. 2017), pp. 60–88. ISSN: 13618423. DOI: 10.1016/j.media.2017.07.005.

[62] Lu Liu et al. "Anatomy-aided deep learning for medical image segmentation: A review". In: *Physics in Medicine and Biology* 66.11 (June 2021). ISSN: 13616560. DOI: 10.1088/1361-6560/abfbf4.

[63] Alejandro Lopez-Perez, Rafael Sebastian, and Jose M. Ferrero. "Three-dimensional cardiac computational modelling: methods, features and applications". In: *BioMedical Engineering Online* 14.1 (Apr. 2015). ISSN: 1475925X. DOI: 10.1186/s12938-015-0033-5.

[64] William E Lorensen and Harvey E Cline. "Marching cubes: A high resolution 3D surface construction algorithm". In: *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '87 21.4 (Aug. 1987), pp. 163–169. DOI: 10.1145/37401.37422.

[65] Lukas Radl et al. "AVT: Multicenter aortic vessel tree CTA dataset collection with ground truth segmentation masks". In: *SPIE Medical Imaging, Proceedings* 40 (2022), p. 116001. DOI: 10.6084/m9.figshare.14806362. URL: http://creativecommons.org/licenses/by/4.0/.

[66] Jun Ma et al. "Segment Anything in Medical Images and Videos: Benchmark and Deployment". In: (Aug. 2024). URL: http://arxiv.org/abs/2408.03322.

[67] Gabriel Maher, Nathan Wilson, and Alison Marsden. "Accelerating cardiovascular model building with convolutional neural networks". In: *Medical and Biological Engineering and Computing* 57.10 (2019), pp. 2319–2335. ISSN: 17410444. DOI: 10.1007/s11517-019-02029-3.

[68] Gabriel Maher et al. "Neural Network Vessel Lumen Regression for Automated Lumen Cross-Section Segmentation in Cardiovascular Image-Based Modeling". In: *Cardiovascular Engineering and Technology* 11.6 (Dec. 2020), pp. 621–635. ISSN: 18694098. DOI: 10.1007/s13239-020-00497-5.

[69] Karttikeya Mangalam et al. "From Goals, Waypoints & Paths To Long Term Human Trajectory Forecasting". In: *Proceedings of the IEEE International Conference on Computer Vision*. Institute of Electrical and Electronics Engineers Inc., 2021, pp. 15213–15222. ISBN: 9781665428125. DOI: 10.1109/ICCV48922.2021.01495.

[70] Karttikeya Mangalam et al. "It is not the journey but the destination: Endpoint conditioned trajectory prediction". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. 2020, pp. 759–776.

[71] Francesco Marchetti et al. "Mantra: Memory augmented networks for multiple trajectory prediction". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 7143–7152.

[72] Alison L. Marsden. "Optimization in cardiovascular modeling". In: *Annual Review of Fluid Mechanics* 46 (Jan. 2014), pp. 519–546. ISSN: 00664189. DOI: 10.1146/annurev-fluid-010313-141341.

[73] Alison L. Marsden et al. "A New Multiparameter Approach to Computational Simulation for Fontan Assessment and Redesign". In: *Congenital Heart Disease* 5.2 (Mar. 2010), pp. 104–117. ISSN: 1747079X. DOI: 10.1111/j.1747-0803.2010.00383.x.

[74] Alison L. Marsden et al. "Evaluation of a novel Y-shaped extracardiac Fontan baffle using computational fluid dynamics". In: *Journal of Thoracic and Cardiovascular Surgery* 137.2 (2009). ISSN: 00225223. DOI: 10.1016/j.jtcvs.2008.06.043.

[75] Hicham Messaoudi et al. "Cross-dimensional transfer learning in medical image segmentation with deep learning". In: *Medical Image Analysis* 88 (Aug. 2023). ISSN: 13618423. DOI: 10.1016/j.media.2023.102868.

[76] Vuk Milišić and Alfio Quarteroni. "Analysis of lumped parameter models for blood flow simulations and their relation with 1D models". In: *Mathematical Modelling and Numerical Analysis* 38.4 (July 2004), pp. 613–632. ISSN: 0764583X. DOI: 10.1051/m2an:2004036.

[77] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation". In: *2016 Fourth International Conference on 3D Vision (3DV)*. 2016, pp. 565–571. DOI: 10.1109/3DV.2016.79.

[78] Milner JS et al. "Hemodynamics of human carotid artery bifurcations: Computational studies with models reconstructed from magnetic resonance imaging of normal subjects". In: *Journal of vascular surgery* 28.1 (1998), pp. 143–156. DOI: 10.1016/s0741-5214(98)70210-1.

[79] Mehran Mirramezani and Shawn C. Shadden. "A Distributed Lumped Parameter Model of Blood Flow". In: *Annals of Biomedical Engineering* 48.12 (Dec. 2020), pp. 2870–2886. ISSN: 15739686. DOI: 10.1007/s10439-020-02545-6.

[80] Mehran Mirramezani and Shawn C. Shadden. "Distributed lumped parameter modeling of blood flow in compliant vessels". In: *Journal of Biomechanics* 140 (July 2022), p. 111161. ISSN: 0021-9290. DOI: 10.1016/J.JBIOMECH.2022.111161.

[81] Mehran Mirramezani et al. "Reduced order models for transstenotic pressure drop in the coronary arteries". In: *Journal of Biomechanical Engineering* 141.3 (Mar. 2019). ISSN: 15288951. DOI: 10.1115/1.4042184.

[82] Rajat Mittal et al. "Computational modeling of cardiac hemodynamics: Current status and future outlook". In: *Journal of Computational Physics* 305 (Jan. 2016), pp. 1065–1082. ISSN: 10902716. DOI: 10.1016/j.jcp.2015.11.022.

[83] Sara Moccia et al. "Blood vessel segmentation algorithms — Review of methods, datasets and evaluation metrics". In: *Computer Methods and Programs in Biomedicine* 158 (May 2018), pp. 71–91. ISSN: 18727565. DOI: 10.1016/j.cmpb.2018.02.001.

[84] Javier Montalt-Tordera et al. "Automatic segmentation of the great arteries for computational hemodynamic assessment". In: *Journal of Cardiovascular Magnetic Resonance* 24.1 (Dec. 2022). ISSN: 1532429X. DOI: 10.1186/s12968-022-00891-z.

[85] Paul D Morris et al. "Computational fluid dynamics modelling in cardiovascular medicine". In: *Heart* 102.1 (Jan. 2016), pp. 18–28. DOI: 10.1136/heartjnl-2015-308044. URL: http://heart.bmj.com/.

[86] Debanjan Mukherjee et al. "The Role of Circle of Willis Anatomy Variations in Cardio-embolic Stroke: A Patient-Specific Simulation Based Study". In: *Annals of Biomedical Engineering* 46.8 (Aug. 2018), pp. 1128–1145. ISSN: 15739686. DOI: 10.1007/s10439-018-2027-5.

[87] Arjun Narayanan, Fanwei Kong, and Shawn Shadden. "LinFlo-Net: A Two-Stage Deep Learning Method to Generate Simulation Ready Meshes of the Heart". In: *Journal of Biomechanical Engineering* 146.7 (July 2024). ISSN: 15288951. DOI: 10.1115/1.4064527.

[88] Nigamaa Nayakanti et al. "Wayformer: Motion forecasting via simple & efficient attention networks". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 2980–2987.

[89] Anam Nazir et al. "OFF-eNET: An Optimally Fused Fully End-to-End Network for Automatic Dense Volumetric 3D Intracranial Blood Vessels Segmentation". In: *IEEE Transactions on Image Processing* 29 (2020), pp. 7192–7202. ISSN: 19410042. DOI: 10.1109/TIP.2020.2999854.

[90] Ali Hatamizadeh Nvidia et al. *UNETR: Transformers for 3D Medical Image Segmentation*. Tech. rep. URL: https://monai.io/research/unetr.

[91] Ozan Oktay et al. "Attention U-Net: Learning Where to Look for the Pancreas". In: *arXiv preprint arXiv:1804.03999* (May 2018). URL: http://arxiv.org/abs/1804.03999.

[92] Mette S Olufsen. "Structured tree outflow condition for blood flow in larger systemic arteries". In: *Am J Physiol* 1.276 (Jan. 1999), pp. 257–268. DOI: 10.1152/ajpheart.1999.276.1.H257.

[93] Phani Kumari Paritala et al. "Reproducibility of the computational fluid dynamic analysis of a cerebral aneurysm monitored over a decade". In: *Scientific Reports* 13.1 (Dec. 2023). ISSN: 20452322. DOI: 10.1038/s41598-022-27354-w.

[94] Martin R. Pfaller et al. "Automated generation of 0D and 1D reduced-order models of patient-specific blood flow". In: *International Journal for Numerical Methods in Biomedical Engineering* 38.10 (Oct. 2022). ISSN: 20407947. DOI: 10.1002/cnm.3639.

[95] Romina Plitman Mayo et al. "Numerical models for assessing the risk of leaflet thrombosis post-transcatheter aortic valve-in-valve implantation: Risk of Leaflet Thrombosis post ViV TAVI". In: *Royal Society Open Science* 7.12 (Dec. 2020). ISSN: 20545703. DOI: 10.1098/rsos.201838.

[96] Michal Plotkowiak et al. *High Performance Computer Simulations of Cardiac Electrical Function Based on High Resolution MRI Datasets*. Tech. rep. URL: www.itk.org.

[97] Harry Pratt et al. "Automatic detection and distinction of retinal vessel bifurcations and crossings in colour fundus photography". In: *Journal of Imaging* 4.1 (2018). ISSN: 2313433X. DOI: 10.3390/jimaging4010004.

[98] Maithra Raghu et al. "Transfusion: Understanding Transfer Learning for Medical Imaging". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 3342–3352. URL: https://proceedings.neurips.cc/paper/2019/hash/eb1e78328c46506b46a4ac4a1e378b91-Abstract.html.

[99] Oona Rainio, Jarmo Teuho, and Riku Klén. "Evaluation metrics and statistical tests for machine learning". In: *Scientific Reports* 14.1 (Dec. 2024). ISSN: 20452322. DOI: 10.1038/s41598-024-56706-x.

[100] Edoardo Remelli et al. "MeshSDF: Differentiable Iso-Surface Extraction". In: *Advances in Neural Information Processing Systems*. Vancouver, Canada, 2020.

[101] Mohammad Rezaeimoghaddam and Frans N. van de Vosse. "Continuum modeling of thrombus formation and growth under different shear rates". In: *Journal of Biomechanics* 132 (Feb. 2022). ISSN: 18732380. DOI: 10.1016/j.jbiomech.2021.110915.

[102] Olafand Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab et al. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4.

[103] Andrey Rudenko et al. "Human motion trajectory prediction: A survey". In: *The International Journal of Robotics Research* 39.8 (2020), pp. 895–935.

[104] Tomas Sakinis et al. "Interactive segmentation of medical images through fully convolutional neural networks". In: (Mar. 2019). URL: http://arxiv.org/abs/1903.08205.

[105] Zohaib Salahuddin, Matthias Lenga, and Hannes Nickisch. "Multi-Resolution 3D Convolutional Neural Networks for Automatic Coronary Centerline Extraction in Cardiac CT Angiography Scans". In: *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*. IEEE, Apr. 2021, pp. 91–95. DOI: 10.1109/ISBI48211.2021.9434002. URL: http://arxiv.org/abs/2010.00925%20http://dx.doi.org/10.1109/ISBI48211.2021.9434002.

[106] Tim Salzmann et al. "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*. 2020, pp. 683–700.

[107] Erica L. Schwarz et al. "Beyond CFD: Emerging methodologies for predictive simulation in cardiovascular health and disease". In: *Biophysics Reviews* 4.1 (Mar. 2023). ISSN: 26884089. DOI: 10.1063/5.0109400.

[108] Ari Seff et al. "Motionlm: Multi-agent motion forecasting as language modeling". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 8579–8590.

[109] Dibyendu Sengupta et al. "Image-based modeling of hemodynamics in coronary artery aneurysms caused by Kawasaki disease". In: *Biomechanics and Modeling in Mechanobiology* 11.6 (July 2012), pp. 915–932. ISSN: 16177959. DOI: 10.1007/s10237-011-0361-8.

[110] M. Shojima. "Magnitude and Role of Wall Shear Stress on Cerebral Aneurysm. Computational Fluid Dynamic Study of 20 Middle Cerebral Artery Aneurysms". In: *Stroke* (Oct. 2004). ISSN: 0039-2499. DOI: 10.1161/01.str.0000144648.89172.of.

[111] Brooke N. Steele et al. "In vivo validation of a one-dimensional finite-element method for predicting blood flow in cardiovascular bypass grafts". In: *IEEE Transactions on Biomedical Engineering* 50.6 (June 2003), pp. 649–656. ISSN: 00189294. DOI: 10.1109/TBME.2003.812201.

[112] Qi Sun et al. "LIVE-Net: Comprehensive 3D vessel extraction framework in CT angiography". In: *Computers in Biology and Medicine* 159 (June 2023), p. 106886. ISSN: 00104825. DOI: 10.1016/j.compbiomed.2023.106886.

[113] Numi Sveinsson Cepero and Shawn C Shadden. "SeqSeg: Learning Local Segments for Automatic Vascular Model Construction". In: *Annals of Biomedical Engineering* 53.1 (2025), pp. 158–179. ISSN: 1573-9686. DOI: 10.1007/s10439-024-03611-z. URL: https://doi.org/10.1007/s10439-024-03611-z.

[114] Nima Tajbakhsh et al. "Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?" In: *IEEE Transactions on Medical Imaging* 35.5 (May 2016), pp. 1299–1312. ISSN: 1558254X. DOI: 10.1109/TMI.2016.2535302.

[115] Gabriel Taubin, Tong Zhang, and Gene Golub. "Optimal surface smoothing as filter design". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 1064. Heidelberg: Springer Verlag, 1996, pp. 283–292. ISBN: 978-3-540-61122-6. DOI: 10.1007/bfb0015544.

[116] C. A. Taylor and C. A. Figueroa. "Patient-specific Modeling of Cardiovascular Mechanics". In: *Annual review of biomedical engineering* 11 (Aug. 2009), p. 109. ISSN: 15239829. DOI: 10.1146/annurev.bioeng.10.061807.160521. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC4581431/.

[117] Charles A Taylor, Thomas J R Hughes, and Christopher K Zarins. "Finite element modeling of blood flow in arteries". In: *Computer Methods in Applied Mechanics and Engineering* 158.1 (1998), pp. 155–196. ISSN: 0045-7825. DOI: https://doi.org/10.1016/S0045-7825(98)80008-X. URL: https://www.sciencedirect.com/science/article/pii/S004578259880008X.

[118] Chayut Teeraratkul et al. "Computational investigation of blood flow and flow-mediated transport in arterial thrombus neighborhood". In: *Biomechanics and Modeling in Mechanobiology* 20.2 (Apr. 2021), pp. 701–715. ISSN: 16177940. DOI: 10.1007/s10237-020-01411-7.

[119] Giles Tetteh et al. "DeepVesselNet: Vessel Segmentation, Centerline Prediction, and Bifurcation Detection in 3-D Angiographic Volumes". In: *Frontiers in Neuroscience* 14 (Dec. 2020). ISSN: 1662453X. DOI: 10.3389/fnins.2020.592352.

[120] Neerja Thakkar et al. "Adaptive Human Trajectory Prediction via Latent Corridors". In: *Computer Vision – ECCV 2024*. Dec. 2025, pp. 297–314. DOI: 10.1007/978-3-031-72920-1{_}17.

[121] Fangzheng Tian et al. "Automatic coronary artery segmentation algorithm based on deep learning and digital image processing". In: *Applied Intelligence* 51.12 (Dec. 2021), pp. 8881–8895. ISSN: 15737497. DOI: 10.1007/s10489-021-02197-6.

[122] Catalina Tobon-Gomez et al. "Benchmark for Algorithms Segmenting the Left Atrium From 3D CT and MRI Datasets". In: *IEEE Transactions on Medical Imaging* 34.7 (July 2015), pp. 1460–1473. ISSN: 1558254X. DOI: 10.1109/TMI.2015.2398818.

[123] Adam Updegrove et al. "SimVascular: An Open Source Pipeline for Cardiovascular Simulation". In: *Annals of Biomedical Engineering* 45.3 (Mar. 2017), pp. 525–541. ISSN: 15739686. DOI: 10.1007/s10439-016-1762-8.

[124] Weiyue Wang et al. "DISN: Deep Implicit Surface Network for High-quality Single-view 3D Reconstruction". In: *arXiv preprint arXiv:1905.10711* (May 2019). URL: https://github.com/laughtervv/DISN..

[125] Jakob Wasserthal et al. "TotalSegmentator: Robust Segmentation of 104 Anatomic Structures in CT Images". In: *Radiology: Artificial Intelligence* 5.5 (2023). DOI: 10.5281/zenodo.6802613. URL: https://www.github.com/wasserth/TotalSegmentator.

[126] Jelmer M. Wolterink et al. "An evaluation of automatic coronary artery calcium scoring methods with cardiac CT using the orCaScore framework". In: *Medical Physics* 43.5 (May 2016), pp. 2361–2373. ISSN: 24734209. DOI: 10.1118/1.4945696.

[127] Jelmer M. Wolterink et al. "Coronary artery centerline extraction in cardiac CT angiography using a CNN-based orientation classifier". In: *Medical Image Analysis* 51 (Jan. 2019), pp. 46–60. ISSN: 13618423. DOI: 10.1016/j.media.2018.10.005.

[128] Yuan Xue et al. "Shape-Aware Organ Segmentation by Predicting Signed Distance Maps". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.07 (Apr. 2020), pp. 12565–12572. ISSN: 2374-3468. DOI: 10.1609/aaai.v34i07.6946.

[129] Wenjian Yao et al. "From CNN to Transformer: A Review of Medical Image Segmentation Models". In: *Journal of Imaging Informatics in Medicine* 37.4 (Mar. 2024), pp. 1529–1547. ISSN: 29482933. DOI: 10.1007/s10278-024-00981-7.

[130] Pak-Hei Yeung et al. "ImplicitVol: Sensorless 3D Ultrasound Reconstruction with Deep Implicit Representation". In: *arXiv preprint arXiv:2109.12108* (Sept. 2021). URL: http://arxiv.org/abs/2109.12108.

[131] Rui Zhang et al. "Deep-Learning-Based Point Cloud Semantic Segmentation: A Survey". In: *Electronics* 12.17 (Aug. 2023), p. 3642. ISSN: 2079-9292. DOI: 10.3390/electronics12173642.

[132] Hengshuang Zhao et al. "Pyramid Scene Parsing Network". In: *arXiv:1612.01105* (Apr. 2017). URL: http://arxiv.org/abs/1612.01105.

[133] Xiahai Zhuang et al. "Evaluation of algorithms for Multi-Modality Whole Heart Segmentation: An open-access grand challenge". In: *Medical Image Analysis* 58 (Dec. 2019). ISSN: 13618423. DOI: 10.1016/j.media.2019.101537.

[134] Christos Zohios, Georgios Kossioris, and Yannis Papaharilaou. "Geometrical methods for level set based abdominal aortic aneurysm thrombus and outer wall 2D image segmentation". In: *Computer Methods and Programs in Biomedicine* 107.2 (Aug. 2012), pp. 202–217. ISSN: 01692607. DOI: 10.1016/j.cmpb.2011.06.009.