

Detection of Coronary Arteries in Contrast-Enhanced CT Images

Master's thesis in Complex Adaptive Systems

FREDRIK ELOFSSON

MASTER'S THESIS 2016:EX035

Detection of Coronary Arteries in Contrast-Enhanced CT images

Fredrik Elofsson



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Signals and Systems
Image Analysis and Computer Vision
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2016

Detection of Coronary Arteries in Contrast-Enhanced CT Images

Fredrik Eloffsson

© Fredrik Eloffsson, 2016.

Examiner: Fredrik Kahl, Department of Signals and Systems

Master's Thesis 2016:EX035
Department of Signals and Systems
Image Analysis and Computer Vision
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Visualization of the spherical assessment of shortest paths to track the right coronary artery (RCA). The origin of the RCA is marked as a red circle, and the starting point in the shortest-path algorithm is the black star. Shortest paths from this starting point to all points on a spherical surface are assessed, and the best are plotted with thick black lines, showing a majority preferring the true artery (red line). The remainder of the computed shortest paths are shown in grey or hidden to make the result visible.

Typeset in L^AT_EX.
Gothenburg, Sweden 2016.

FREDRIK ELOFSSON

Department of Signals and Systems
Chalmers University of Technology

Abstract

Within cardiology, assessing the function of coronary arteries is of vital importance as these arteries supply the heart with oxygenated blood. A method routinely used in this assessment is x-ray computer tomography (CT), where a 3D image of the heart is analyzed by an expert. However, this is a laborious task, and thus it is tempting to automate parts of this process. In this thesis, an automated coronary artery segmentation method is presented. Firstly, the origins of the right coronary artery (RCA) and the left anterior descending artery (LAD) are estimated via a feature-based image registration method. Secondly, an incremental shortest-path algorithm is presented to track the vessels.

The segmentation method is evaluated on the public data set of the Rotterdam Coronary Artery Algorithm Evaluation Framework (RCAAEF). Using the training set of the RCAAEF, the average error of estimating the origin of the RCA and LAD is found to be 5 mm. Furthermore, the vessel segmentation is evaluated by tracking the RCA in the training set. The tracking performance is found to be on average 53 % in the framework's standardized overlap measure (0-100 %, where "100 %" means that the entire segmented vessel is within the radius of the true vessel). The results are worse than that of other methods submitted to the RCAAEF, and potential improvements of the proposed method, such as choice of data-cost function and post-processing options, are discussed.

Keywords: medical image analysis, feature-based image registration, Rotterdam Coronary Artery Algorithm Evaluation Framework, coronary arteries, image segmentation

Acknowledgements

This Master's Thesis was carried out at MedTech West* at Sahlgrenska University Hospital. I owe my deepest gratitude to my past and present colleagues there, in particular Jennifer Alvéén, Måns Larsson, Moa Peter and Agnes Tegen, without you this thesis would not have been possible.

For the original idea to this project, and for support and great patience, I want to sincerely thank my examiner, Fredrik Kahl. For helpful input to the project, I want to acknowledge Alexander Norlén. I also would like to show my gratitude to Rolf Heckemann and to my opponent Oskar Lilja for helpful feedback on the written report. Finally, I want to thank my supervisor Behrooz Nasihatkon.

Fredrik Elofsson
Gothenburg, May 2016

* MedTech West is a network and collaborative platform for research, education, development and evaluation of new biomedical concepts and technologies. It is a joint venture and was founded in 2009 by Chalmers University of Technology, Gothenburg University, University of Borås, Västra Götalandsregionen, and Sahlgrenska University Hospital. www.medtechwest.se

Contents

List of Figures	xi
List of Tables	xiii
List of Algorithms	xv
1 Introduction	1
1.1 Background	1
1.2 Problem formulation and scope	1
1.3 The Rotterdam Coronary Artery Algorithm Evaluation Framework	2
1.3.1 Data set and reference standard	2
1.3.2 Evaluation framework	4
1.4 Related work and contributions	6
1.5 Structure of the report	6
2 Theory	7
2.1 The heart and the coronary arteries	8
2.2 Feature-based image registration	9
2.3 The Frangi vesselness measure	10
2.4 k -means cluster analysis	11
2.5 Dijkstra’s algorithm	12
3 Method	15
3.1 Preprocessing	16
3.2 Pericardium segmentation	16
3.3 Estimation of starting points	18
3.4 Applying the Frangi vesselness measure	20
3.5 Tracking the vessel	22
3.5.1 Dijkstra’s algorithm	22
3.5.2 Assessment of shortest paths	23
3.5.3 Termination	24
3.5.4 Summary	25
3.6 Evaluation	26
3.7 Implementation	27
3.7.1 Run times	27

4	Results	29
4.1	Estimation of starting points	29
4.2	Tracking the vessel	31
5	Discussion	37
5.1	Estimation of starting positions	37
5.2	Tracking the vessel	37
6	Conclusion	39
	Bibliography	43

List of Figures

1.1	A drawing of a heart and the coronary arteries	2
1.2	The reference standard for one image in the training set	3
1.3	The radius of the RCA and LAD in all images in the training set as as function of their length	4
2.1	Example of the shortest-path problem	13
3.1	Flowchart of proposed method.	15
3.2	Pericardium segmentation in one of the CT images	17
3.3	Distance to pericardium for the reference standard	17
3.4	Comparison of cluster center estimates.	19
3.5	The Frangi vesselness measure as a function of the length of a vessel	21
3.6	Illustration of 4-connectivity	23
4.1	Average error for coronary origin estimation for the RCA and LAD using the two estimators	30
4.2	Visual result of the tracking algorithm	33
4.3	Deviation from gold standard for the estimated vessels.	34
4.4	Distance to pericardium for the estimated vessel positions.	35

List of Tables

1.1	Meta data of the images in RCAAEF	2
3.1	Chosen parameters for the Frangi filter.	20
3.2	Algorithm parameters.	25
3.3	Run times of key parts of the algorithm.	27
4.1	Minimum error and parameters for the feature-based image registration of coronary origins	29
4.2	Evaluation scores for all reported automatic methods in the RCAAEF challenge	32

List of Algorithms

1	The k -means clustering algorithm	11
2	Dijkstra's algorithm	13
3	Main algorithm	25

1

Introduction

1.1 Background

According to the WHO, ischemic heart disease, also known as coronary artery disease, is the leading cause of death in the developed part of the world [1]. Supplying the heart with oxygenated blood, the coronary arteries are vital as a restriction of blood flow to the heart itself may lead to e.g. myocardial infarction (heart attack) [2, p. 334]. See Figure 1.1 for an illustration of the heart and the coronary arteries.

In cardiology, the function of the coronary arteries is routinely assessed using non-invasive x-ray computer tomography (CT). In that process, the patient is injected with a contrast agent to enhance the appearance of blood vessels. This method generates a 3D image which is then assessed by an expert. However, this is a laborious task and could benefit from automisation. For instance, by finding the relevant vessels and presenting them to the expert for direct assessment.

There is also a specific demand for an automated coronary artery segmentation in CT images within the project of SCAPIS (Swedish CardioPulmonary bioImage Study), a national project where thousands of individuals are assessed and tracked to estimate and detect risk factors for cardiovascular diseases. The project includes more images than realistically possible for a single expert analyse [3] and by having a computer algorithm that automatically segments and possibly also assesses the condition of the coronary arteries, it would be possible to generate knowledge of risk indicators of cardiovascular diseases from this data set.

1.2 Problem formulation and scope

The purpose of this thesis is to develop a method that automates the process of detection and segmentation of coronary arteries' centerlines in contrast-enhanced CT images. To “detect” means to enhance structures that are vessels. To “segment” means to find a set of positions $\{\mathbf{y}_j\}_{j=1}^n$ in \mathbb{R}^3 , that together form an estimate for the position of the centerline in the true arteries, which may be represented as parametric curves $\mathcal{C}_i : I \mapsto \mathbb{R}^3$, where I is an interval $[0, L_i]$ analogous to the total running length of the vessel i .

This thesis considers estimation of the origins of both the right coronary artery (RCA) and left anterior descending artery (LAD). However, it will only consider tracking one of the main coronary arteries: the RCA, as it has less major bifurcations than the LAD. Further, it will only consider the position of the vessel's centerline, not the vessel's radius.

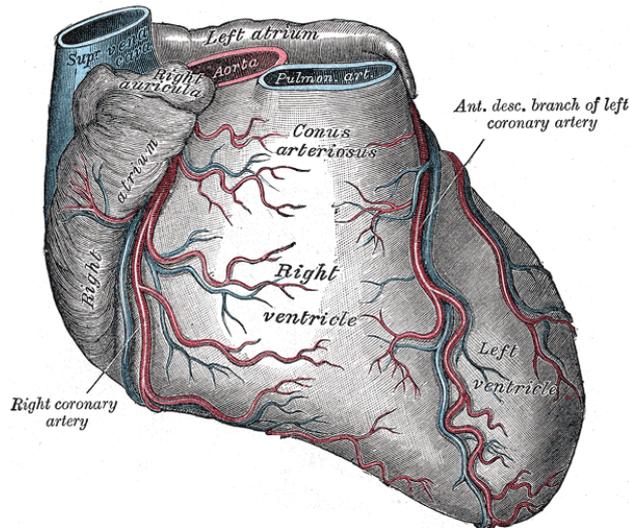


Figure 1.1: A drawing of a heart and the coronary arteries. (Gray’s anatomy [4]).

1.3 The Rotterdam Coronary Artery Algorithm Evaluation Framework

The Rotterdam Coronary Artery Algorithm Evaluation Framework (RCAAEF) is a framework for evaluation and benchmarking of algorithms segmenting the coronary arteries. It contains publicly available data and an evaluation framework. As of this date (May 2016), some 40 methods have reported their results. The methods compete in different categories, depending on how much human interaction they require. The framework is presented in [5] and the remainder of this section presents the most relevant parts for this thesis: the data set, the reference standard and the evaluation framework.

1.3.1 Data set and reference standard

There are in total 31 CT images of the heart available through the RCAAEF. The images are divided into two sets: a training set (8 images) and a test set (23 images). For the training set, a reference standard, or “gold standard”, for the positions of the coronary arteries are given. See Table 1.1 for summary of the data set’s properties.

Table 1.1: Meta data of the images in the Rotterdam Coronary Artery Algorithm Evaluation Framework.

Property	Value
No. of images	31
– of which in test set	23
– of which in training set	8
File format	.mhd (header-file), .raw (image)
File size	ca. 140 MB
Image dimensions	Varies, on average: $512 \times 512 \times 287$
Voxel Size	Varies, on average: $0.32 \times 0.32 \times 0.40$ mm
Data precision	Single

The gold standard of the coronary arteries in the CT images are a weighted average of four independent experts' annotations [5]. It is available for four coronary arteries: the right coronary artery (RCA), the left anterior descending artery (LAD), the left circumflex artery (LCX) and the fourth being the biggest of one of the side branches to the three previously mentioned arteries. The gold standard of the vessels are paths starting from the aorta, and follows the vessel until it no longer can be discriminated compared to the grey-level background. They are numerically represented as a set of coordinates, where the distances between subsequent points are roughly 0.3 mm. See Figure 1.2 for an example.

Besides providing the positions for these arteries, the radius of the vessels at every position is also provided. See Figure 1.3 for how the radius of the RCA and LAD varies with the length of the vessels. Extracted from this figure, the average starting radius of vessels are 2.5 and 2.6 mm for the RCA and LAD respectively.

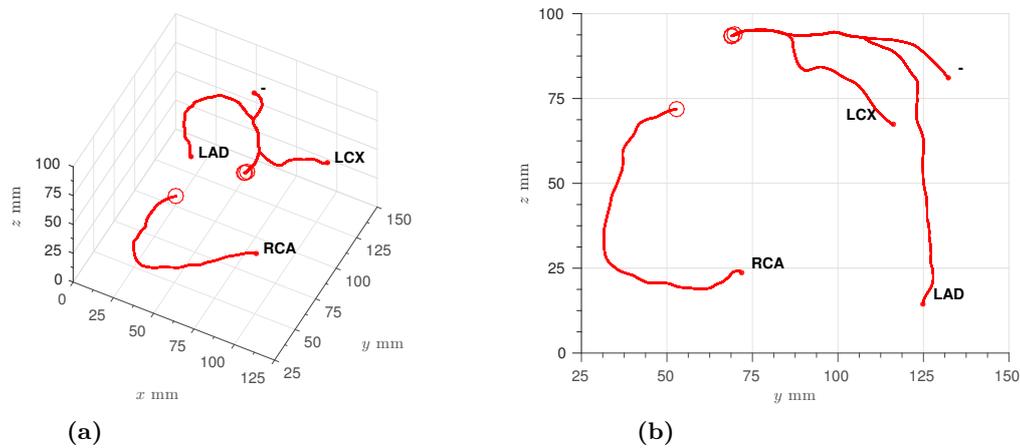
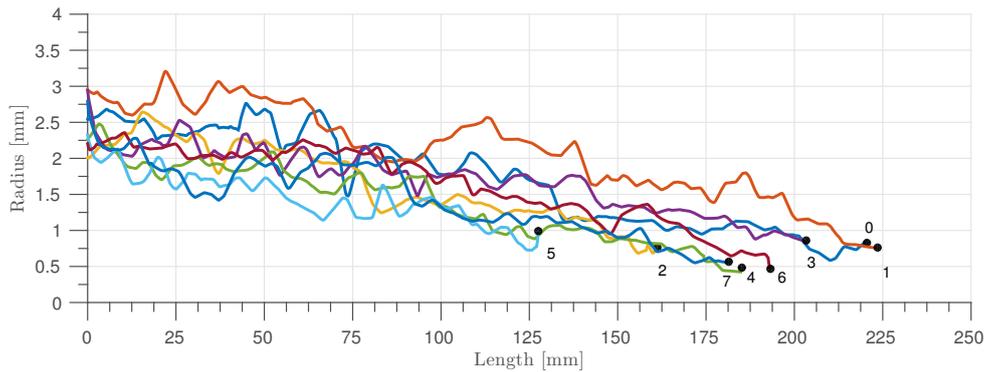
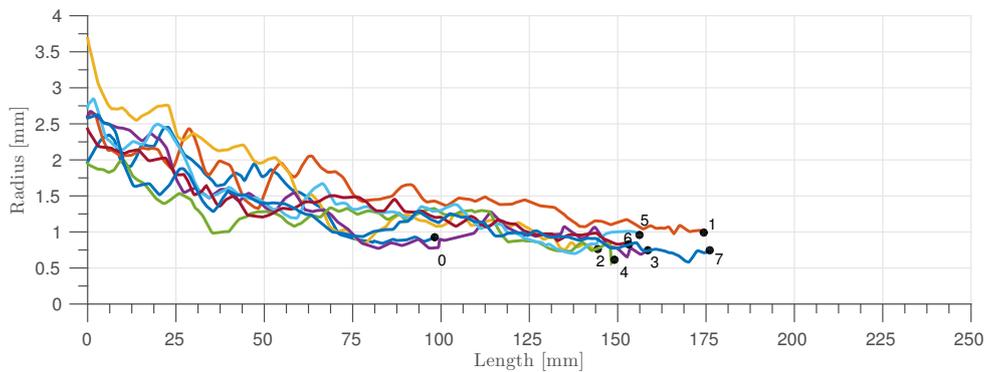


Figure 1.2: The reference standard for one image in the training set from two different angles. Origins are marked with a red circle (o). The annotations are RCA, LAD, LCX and a fourth artery “-”. Notable, there are two major bifurcations along the LAD vessel.



(a) RCA



(b) LAD

Figure 1.3: The radius of the RCA and LAD in all images in the training set as function of their length. The small black digits represent the image number in the training set (0-7).

1.3.2 Evaluation framework

The main contribution of [5] is its evaluation framework for an estimated position of coronary artery centerlines. To evaluate an estimated centerline in the RCAAEF, it is first resampled equidistantly using a distance of 0.3 mm. It is then clipped at the origin at the aorta, using a disc perpendicular to the vessel’s direction in the start. Due to this, all evaluated centerlines per definition start at the same point.

Next, correspondences between the gold standard and the evaluated centerline are created. Let the pair of points with the shortest Euclidean distance be that pair’s respective correspondence. If a point on the *gold standard* has a connection to a point on the estimated centerline within the gold standard’s radius, it is marked as a true positive, TPR, otherwise as a false negative, FN. If a point on the *estimated centerline* has a correspondence within its radius it is marked true positive, TPM, and false positive, FP, otherwise.

The following ratio is defined, and is a general overlap measure of how well the estimated centerline follows the gold standard:

$$OV = \frac{\|TPM\| + \|TPR\|}{\|TPM\| + \|TPR\| + \|FN\| + \|FP\|}, \quad (1.1)$$

where $\|\cdot\|$ denotes the number of elements. This ratio is exactly 1 if the entire evaluated

centerline is within the true radius of the true vessel, and less otherwise.

As a side note, two other ratios are defined in [5], called “Overlap until first error (OF)” and “Overlap with most clinically relevant part of the vessel (OT)”, where the former is very similar to the ratio above, only calculated from the start of the vessel up until the point where the first false positive occurs. The latter one is identical to Equation (1.1), only that it is evaluated at a distance from the start up to the point where the vessel’s radius goes below 1.5 mm, a size lower of which is stated to be less clinically important.

Further, [5] defines an accuracy measure for the evaluated centerline, only applied to regions where the tracking succeeded, i.e. where the distance between the evaluated centerline and gold standard is less than the vessel’s radius. It is calculated as the average of the distances between the reference standard and evaluated centerline. This accuracy measure is later weighed together with the accuracies of the four independent experts, producing a score. If it equals 50, the performance is similar to one the experts. If its 100, it is “perfect”. Finally, a rank is obtained based on the performance of all the overlap measures on all vessels compared to other participating methods in the RCAAEF-challenge.

1.4 Related work and contributions

Within the RCAAEF challenge (see Section 1.3), 40 methods have made contributions [5, 6]. Ten of the 40 methods extract the coronary artery centerlines in an automatic way. Their methods of choice range from shape/feature-based models adapted to the coronary arteries, e.g. [7, 8, 9], to more data-driven approaches using cost-functions, e.g. [8, 10, 11]. Outside of the RCAAEF, good results have been reported by machine-learning-based methods to classify voxels as vessels or not [12].

In 1998, Frangi et al. presented a vesselness measure that is still used today by many methods [13]. In [14], the measure is used in an example to demonstrate an alternative version of the shortest-path problem. Supplied with *both* starting and end point, their algorithm successfully tracks a centerline in an image from the RCAAEF.

This thesis presents a method to optimize the parameters used in the feature-based image registration of [15], in order to create as good estimates for the origins of the right coronary artery, RCA, and the left anterior descending artery, LAD, as possible. The advantages of using feature-based image registration is that it is generally faster than intensity-based methods. Further, it presents an incremental shortest-path vessel tracking method using a combination of the classical and rather simple Frangi vesselness measure and the perpendicular distance to the pericardium as data cost.

1.5 Structure of the report

The structure of this thesis is the following:

In Chapter 2 (p. 7), the reader may find the theoretical framework for the concepts used in the method, found in Chapter 3 (p. 15). The results are presented in Chapter 4 (p. 29).

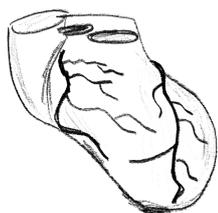
Finally, the report is ended by a discussion (Chapter 5, p. 37), with thoughts concerning future work and a conclusion (Chapter 6, p. 39).

2

Theory

In this chapter the theoretical background for the concepts used in this thesis will be covered.

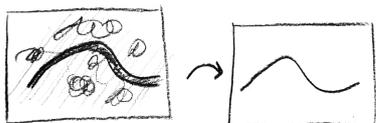
The chapter is structured as follows: first some additional terminology concerning the heart and the coronary arteries will be introduced, followed by an introduction of feature-based image registration and a vessel-enhancement filter used in this thesis. Lastly, the chapter is concluded by a presentation of two algorithms: one concerning data clustering and another concerning the shortest path in a graph. Please refer to the table below for page references.



Section 2.1: The heart and the coronary arteries (p. 8)



Section 2.2: Feature-based image registration (p. 9)



Section 2.3: The Frangi vesselness measure (p. 10)



Section 2.4: The k -means clustering algorithm (p. 11)



Section 2.5: Dijkstra's algorithm (p. 12)

2.1 The heart and the coronary arteries

The coronary arteries are the first arteries to exit the aortic valve. From there, they follow the shape of the heart, bifurcating and gradually decreasing in size, like branches in a tree. The word “coronary” is derived from the Latin “corona”, meaning ‘wreath, crown’, and is used in a medical context to describe e.g. vessels and nerves that encircle a part of an organ like a crown [16]. Together with the heart they are contained in a double-walled sack, the pericardium [2, p. 282, 333].

The coronary arteries are designated according to their position and which area of the heart they mainly supply with blood. These are, for example, the right coronary artery (RCA), the left coronary artery (LCA), the left anterior descending artery (LAD) and the left circumflex artery (LCX).

See Figure 1.1 (p. 2) for a drawing of the heart and the coronary arteries.

2.2 Feature-based image registration

Image registration is the process of spatially aligning two images with each other [17]. The ultimate goal of image registration is to have a mapping that transfers any pixel or voxel in a so-called *source image*, I_s , to a corresponding position in a so-called *target image*, I_t . Mathematically, it may be formulated as the problem

$$\arg \min_{\mathbf{T}} \rho(I_t, \mathbf{T} \circ I_s), \quad (2.1)$$

where ρ is a cost-function used when comparing the target image to the image mapped by the transformation \mathbf{T} . \mathbf{T} may represent a translation, scaling, rotation, shearing or a more complicated deformation, “warping”, of the source image and $\mathbf{T} \circ I_s$ is the rendered image by applying \mathbf{T} to I_s .

The choice of cost-function ρ depends on the application and what type of transformations one allows [18]. However, it is typically a comparison of the values of the voxels, intensities, of the target image and the mapped image, a comparison between “features” in the images (such as corners, contours or rapid gradients), or a combination of the two.

The method of image registration in this thesis will be focusing on features, hence the choice of cost-function is feature-based. In feature-based image registration, one starts with finding “interest points”, which are characterized by means of their local neighbourhood and made scale and rotationally invariant. The next step is to match the detected features in the source image to the target image. Finally, one may estimate a transformation \mathbf{T} to map the source image into the target image using the matched features. The main advantages of feature-based image registration is that it is generally faster than intensity-based registration methods.

More specifically, this thesis will be using the feature-detection method presented in [15] called `ff3d`. Based on the methods SIFT[19] and SURF[20], it detects and describes features which are matched by comparing their local neighbourhood. The matches are then used to estimate \mathbf{T} as an affine transformation, i.e. a mapping $\mathbf{T} : \mathbf{x} \mapsto \mathbf{A}\mathbf{x} + \mathbf{t}$, where the matrix multiplication of A represents a scaling, rotation or shearing, and \mathbf{t} a translation. This affine transformation may also be represented as a single matrix multiplication:

$$\begin{pmatrix} \mathbf{y} \\ 1 \end{pmatrix} = \begin{bmatrix} A & \mathbf{t} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} = \mathbf{T}^A \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}, \quad (2.2)$$

where this new matrix \mathbf{T}^A is referred to as an *augmented* matrix.

The method of which [15] estimates \mathbf{T}^A is by using RANSAC, an algorithm suitable for large and noisy data sets [21]. In this case, a random selection of features in the source image are mapped using an estimated \mathbf{T}^A . Depending on the choice of a threshold t_{aff} , features are designated as being either “inliers” or “outliers”, depending on their distance to their matched feature in the target image. The RANSAC algorithm iteratively selects random features and updates \mathbf{T}^A accordingly, stopping when it has reached a predetermined maximum number of iterations. \mathbf{T}^A is finally chosen as the transformation that minimizes either the L_1 - or L_2 -norm of the distances between target features and corresponding mapped source features.

2.3 The Frangi vesselness measure

The Frangi vesselness measure is a multi-scale filter created to enhance vessel-like structures in an image [13]. The filter works in both 2D and 3D, and takes as input a whole image and returns, in the best case, an image with values close to 1 where there are vessels and 0's elsewhere. The remainder of this section will be spent detailing this filter and explaining its parameters.

The central part of the Frangi vesselness measure (or “Frangi filter”), is its analysis of the second order derivatives: the Hessian. First the image is smoothed using a Gaussian kernel at scale s . Next, the Hessian H and its eigenvalues are computed at every voxel in the image. The authors of [13] find that the value of the eigenvalues and direction of eigenvectors contain valuable information regarding the specific voxel's tubularity or “vesselness”.

Arranged in absolute size, let $|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$ be the eigenvalues of H . For an ideal tubular structure in 3D, Frangi et al find

$$\begin{aligned} |\lambda_1| &\approx 0, \\ |\lambda_1| &\ll |\lambda_2|, \\ \lambda_2 &\approx \lambda_3. \end{aligned} \tag{2.3}$$

By geometrical analogies, they define two ratios:

$$\begin{aligned} \mathcal{R}_A &= \frac{|\lambda_2|}{|\lambda_3|}, \\ \mathcal{R}_B &= \frac{|\lambda_1|}{\sqrt{|\lambda_2\lambda_3|}}, \end{aligned} \tag{2.4}$$

A third quantity is defined as

$$\mathcal{S} = \text{“second order structureness”} = \|H\|_F = \sqrt{\sum_{j=1}^3 \lambda_j^2}, \tag{2.5}$$

which provides the filter with a measure that is small in a uniform background, but larger when there are variations (potential vessels).

For an ideal voxel in a vessel-like structure, \mathcal{R}_A and \mathcal{S} should be large and \mathcal{R}_B should be close to zero. They are combined as factors into a single formula which reaches its maximum when this is true:

$$\mathcal{V}_o(s) = \begin{cases} 0 & \text{if } \lambda_2 > 0 \quad \text{or } \lambda_3 > 0 \\ \left(1 - \exp\left(-\frac{\mathcal{R}_A^2}{2\alpha^2}\right) \exp\left(-\frac{\mathcal{R}_B^2}{2\beta^2}\right) \left(1 - \exp\left(-\frac{\mathcal{S}^2}{2c^2}\right)\right)\right) & \text{otherwise} \end{cases}, \tag{2.6}$$

where $\mathcal{V}_o(s)$ denotes the Frangi vesselness measure at scale s . Now, the calculations are redone using a different scale s , and the Frangi filter of a voxel is the maximum value of $\mathcal{V}_o(s)$ found for different scales.

The parameters of the filter are α , β , c and they individually control the impact of each respective term. Authors of [13] use $\alpha = \beta = 0.5$ in their examples, with c taken to be “half the value of the maximum Hessian norm”, and note that it “has proven to work in most cases”.

2.4 k -means cluster analysis

The k -means cluster analysis [22, p. 442] is a simple, iterative clustering algorithm that converges quickly with reasonable clustering. It is in this thesis used as a method to reject false bifurcations. It assumes isotropic, Gaussian distributions with a diagonal covariance matrix, i.e. the scattering in clusters is identical in all coordinate directions and there should be no dependence between clusters.

Given a set of n data points $\{\mathbf{x}_i\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^d$, and a presumed amount of clusters k , the k -means clustering problem may be formulated in the following way: what are the positions of the k clusters' centers $\{\boldsymbol{\mu}_j\}_{j=1}^k$ that “best” describe the clustering of the data?

The k -means clustering algorithm works in three steps: first it randomly initializes random positions of $\boldsymbol{\mu}_j$ and assigns each data point \mathbf{x}_i to the *closest* $\boldsymbol{\mu}_j$. It then updates the positions of $\boldsymbol{\mu}_j$ by letting it be the average (centroid) of the points that have been assigned to cluster j :

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^n z_{ij} \mathbf{x}_i}{\sum_{i=1}^n z_{ij}}, \quad (2.7)$$

where z_{ij} is a “marker”, either 0 or 1, being the latter if \mathbf{x}_i belongs to cluster j and vice versa. Finally, the algorithm iteratively assigns the data to presumably updated cluster positions, and does so until no change in assignment occurs [22, p. 442]. The procedure may be found summarized in algorithm 1.

As a remark, if k is taken to be unreasonably high the algorithm copes with this well, as sooner or later either one cluster will “take over”, or the redundant cluster will have identical assignments as another one: thus the output of the k -means clustering algorithm may *not* need to be exactly k clusters. MATLAB features the k -means clustering algorithm, but a faster version is available here¹.

Algorithm 1: The k -means clustering algorithm [22, p. 442].

Input: k , the assumed number of clusters, data points $\{\mathbf{x}_i\}_{i=1}^n$.

Result: A labeling z_{ij} , where $z_{ij} = 1$ if \mathbf{x}_i belongs to cluster j and updated cluster positions $\{\boldsymbol{\mu}_j\}_{j=1}^k$.

```

1 Initialization: Guess positions of  $\{\boldsymbol{\mu}_j\}_{j=1}^k$  ;
2 while assignments still changing do
3   | Assign  $\mathbf{x}_n$  to its closest  $\boldsymbol{\mu}_j$  ;
4   | if  $\mathbf{x}_i$  assigned to  $\boldsymbol{\mu}_j$  then
5   |   | Set  $z_{ij} = 1$ ;
6   | else
7   |   |  $z_{ij} = 0$ ;
8   | end
9   | Update  $\{\boldsymbol{\mu}_j\}_{j=1}^k$  cluster centers to be the average of the  $\mathbf{x}_i$  that belong to that center according to equation (2.7).
10 end

```

¹Kmeans Clustering by Mo Chen. Updated: 13 Mar 2016.

Available at: <http://www.mathworks.com/matlabcentral/fileexchange/24616-kmeans-clustering>

2.5 Dijkstra’s algorithm

Dijkstra’s algorithm is a fast algorithm for finding the shortest path between any two nodes in a graph, used in this thesis as a way to segment vessels.

Formally, consider the graph $G = (V, E)$ consisting of vertices (or nodes) V connected with edges E . Each edge has a cost (representing a length, a time etc.) that is greater or equal to zero. A path P is a series of connected vertices, and the cost or length of that path is the sum of all edges in that path. Assume for simplicity that there is a way between all vertices in the graph. The *shortest path problem* may be formulated in the following way: given a starting node s , what is the shortest path to a node t ?

The algorithm works in the way that it incrementally determines the minimum distance from s to more and more nodes in V . By keeping track of the set of already “explored” nodes, the algorithm eventually finds $d(t)$, the wanted minimum distance between the start s and destination t . From this known set of minimum distances from s , a path between s and t may be constructed, starting from t , moving backwards.

Formally, the set of nodes that have been explored may be denoted as S . Initially $S = \{s\}$ and $d(s) = 0$. Now, the algorithm considers $v \notin S$, i.e. an unexplored node, and tries to construct a path within S , finally ending in a single edge (u, v) , where $u \in S$. To determine $d(v)$, we consider the quantity $d'(v) = \min_{e=(e,v), u \in S} d(u) + \ell_e$, where ℓ_e denotes the cost of traversing between u to v . We choose the node v for which this is minimized and we assign $d(v)$ this value. The set of explored nodes S is updated to include v and the algorithm continues incrementally until all nodes in V have been explored. Perhaps the algorithm could be visualized as an expanding “cloud” that incrementally covers more and more nodes in V . It finally ends when it covers the whole graph [23, p. 180].

The procedure may be found summarized in Algorithm 2 on page 13. See also the graph in Figure 2.1, illustrating a simple version of the shortest-path problem.

Presented in 1959 by E. Dijkstra [24], the algorithm speed has been improved since. The straight-forward implementation has a time complexity of $\mathcal{O}(|V| \cdot |E|)$, where $|\cdot|$ denotes the number of elements in the set of vertices and edges respectively and \mathcal{O} is the asymptotic upper bound², but its speed may be significantly improved using so-called heap data structures. As a side note, the algorithm speed may be further improved if one knows the cost of traversing an edge to be positive integers [23, p. 179–184]. Dijkstra’s algorithm is implemented in MATLAB in a function called `graphshortestpath`. It has a time complexity of $\mathcal{O}(|E| \cdot \log |V|)$. It also has a built-in functionality for sparse matrices [25].

² $\mathcal{O}(f(n))$ denotes the asymptotic upper bound for the algorithm speed, a frequently used notation in algorithm analysis, roughly meaning that the algorithm speed is limited by this function $f(n)$, times a constant [23, p. 36].

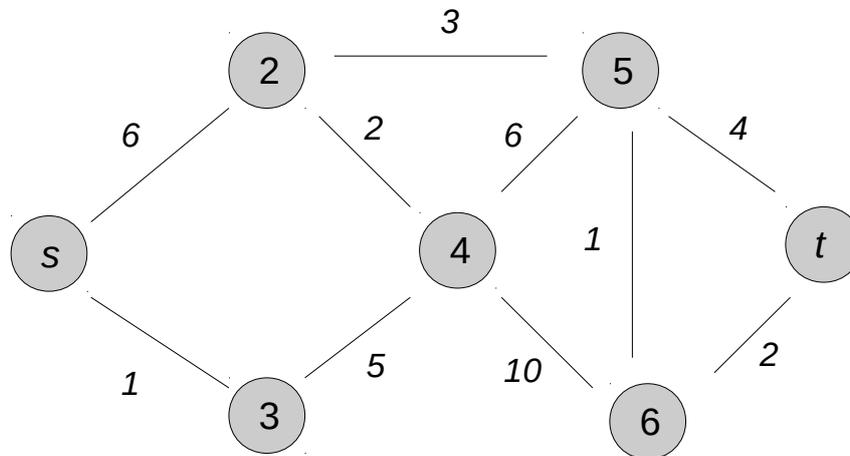


Figure 2.1: An example of the shortest-path problem, where a network of connected nodes with edge weights, represents the cost of traversing from one node to another. The shortest path problem is to find the path with least cost from e.g. s to t . In this case, from s through to node 2, 5, 6 and then to t is the shortest path, generating a total path cost of $6 + 3 + 1 + 2 = 12$.

Algorithm 2: Dijkstra's algorithm [23, p. 180].

Input: A connected graph $G = (V, E)$ with positive edge weights. Starting node s .

Result: The minimum distance $d(v)$ from s to any $v \in V$

- 1 Initialization: Let S be the set of explored nodes. ;
- 2 For every $u \in S$, store a distance $d(u)$;
- 3 Initially, $S = \{s\}$ and $d(u) = 0$;
- 4 **while** $S \neq V$ **do**
- 5 Select a node $v \notin S$ with at least one edge from S for which
- 6 $d'(v) = \min_{e=(e,v), u \in S} d(u) + \ell_e$ is as small as possible ;
- 7 Add s to S and define $d(v) = d'(v)$;
- 8 **end**

3

Method

In this chapter the method of detection and segmentation of coronary arteries will be described. The outline of the chapter is the following: firstly, a preprocessing step of the CT images will be explained. Next, details regarding a pericardium segmentation will be shown, followed by an account of a feature-based image registration of starting points, concluded by the Frangi filter applied to the whole image. These steps provide the necessary information and data for the vessel tracking algorithm. Finally, the chapter ends with listing the ways in which the results will be evaluated.

The method may also be found summarized in Figure 3.1.

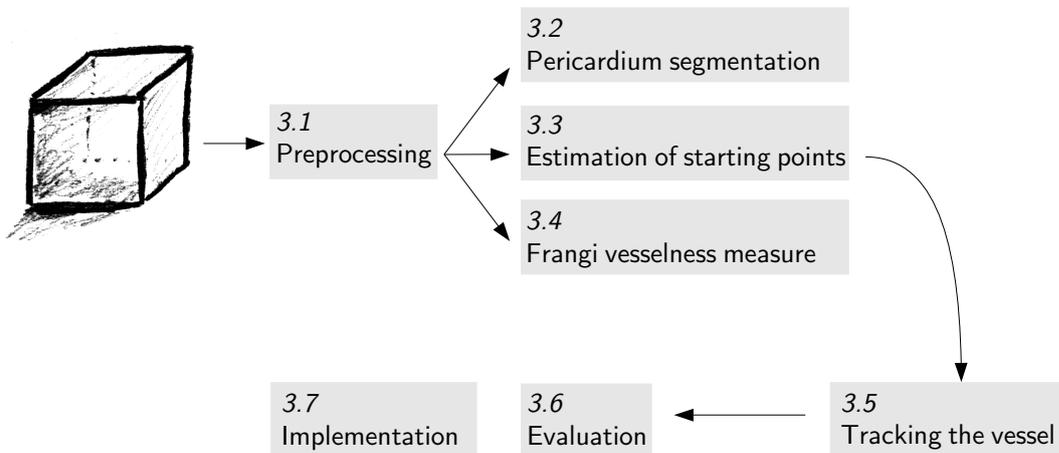


Figure 3.1: Flowchart of proposed method.

3.1 Preprocessing

CT images containing the heart volume were retrieved from the Rotterdam Coronary Artery Algorithm Evaluation Framework (see section 1.3 for details). Frequency analysis of these images has shown that they have an artificially high resolution and can be downsampled by a factor of 2 [26]. Thus, from an approximate size of $[512 \times 512 \times \sim 287]$, all images are downsampled to a size of $[256 \times 256 \times \sim 144]$. Due to their great original size, linear interpolation was used.

As a second preprocessing step, all images are spline-interpolated to have identical voxel dimensions. This makes implementation of later parts of the main algorithm easier, as we now can treat distances between voxels in all directions in the same way. As can be seen in Table 1.1 (p. 2), the voxel dimension in the z -direction is higher. With this interpolation taken into account, the original average voxel dimensions of $[0.32 \times 0.32 \times 0.40]$ mm become $[0.64 \times 0.64 \times 0.64]$ mm.

3.2 Pericardium segmentation

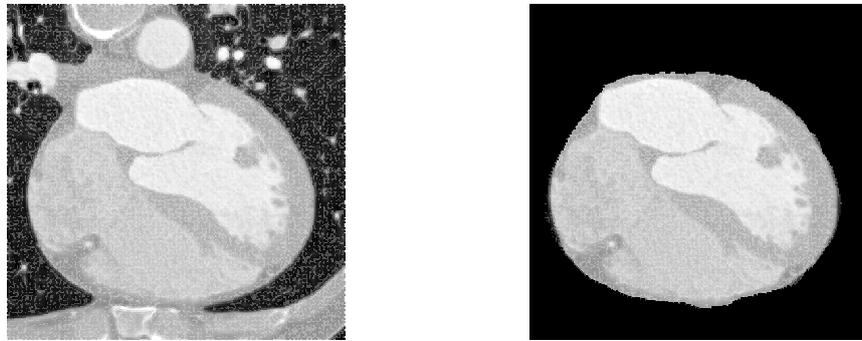
All preprocessed images undergo a pericardium segmentation to be used in a later stage of the proposed method.

The segmentation is done using a multi-atlas feature-based image registration [27]. The output of this method returns a 3D image $D(\mathbf{x})$, with identical size as the original CT image, where each voxel contains a floating number of its estimated signed perpendicular distance in millimeters to the pericardium. Thus, the surface spanned by voxels in $D(\mathbf{x})$ with value close to zero represents the estimate of the pericardium. Consequently, voxels inside the pericardium have a negative value, and those outside carry a positive value.

This segmentation is powerful: as the whole CT volume containing the heart may also include parts of the lungs, bones, all of which may contain other vessels or structures similar to the coronary arteries. By using this segmentation, we confine ourselves to remain within the heart volume.

Visual inspection of the pericardium segmentation in the eight images in the training set showed reasonable pericardium outlines. See Figure 3.2 for a slice where the pericardium has been segmented.

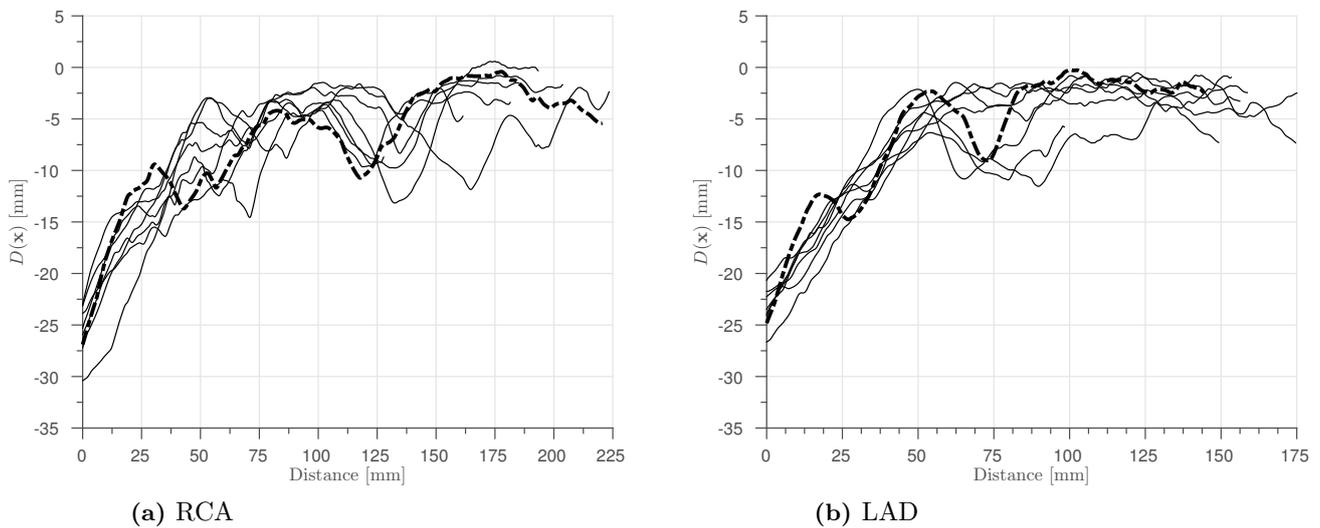
Lastly, the gold standard of the coronary arteries in the training set were assessed in terms of distance to the pericardium, see Figure 3.3, providing helpful information in later stages of the algorithm.



(a)

(b)

Figure 3.2: Pericardium segmentation in one of the CT images.



(a) RCA

(b) LAD

Figure 3.3: Distance to the pericardium $D(\mathbf{x})$ as a function of traversed distance in the RCA and the LAD from the starting point for all images in the training set. For visualization purposes, one of the vessels are highlighted using a thicker dashed line.

3.3 Estimation of starting points

The method in this section will describe how estimates for the starting point or origins of the RCA and LAD arteries are retrieved. In basic terms, the method utilizes the known gold standard in the eight images of the training set to perform a feature-based image registration. To make the registration robust, seven images act as sources images I_s to be registered to the eighth target image I_t , and the resulting origin estimate in I_t is retrieved by “merging” the seven individual estimates. To optimize the parameters of this registration procedure, the “eighth image” is inter-changed and thus all images are at some point acting as the target image.

More formally, let \mathbf{y}_i denote the true starting point of a vessel in image i . To simplify notation, we omit notation regarding if it is the RCA or LAD vessel: the method is the same. We seek to find an estimate for this starting point, denoted as $\hat{\mathbf{y}}_i$. To calculate $\hat{\mathbf{y}}_i$, a large set of features are computed in another image, image j , using the framework presented in Section 2.2. As we are interested solely in finding the position of \mathbf{y}_i , and not mapping the entire image j to image i , a selection of the N closest features to \mathbf{y}_j are chosen.

Using these N features, an affine transformation to corresponding features in image i is estimated. This transformation \mathbf{T}_j^A minimizes the distances between the N features and matched features in image i , where \mathbf{T}^A is the augmented matrix form as presented in Section 2.2. Our great hope is now that also the mapping of \mathbf{y}_j also is as close as possible to \mathbf{y}_i . This is our estimate for the starting point in image i , using image j . Mathematically,

$$\mathbf{T}_j^A(\mathbf{y}_j) = \hat{\mathbf{y}}_i. \quad (3.1)$$

Since we have more images, more estimates of \mathbf{y}_i can be computed. Using the training set of eight images, seven estimates can be computed for each image. These estimates can then be averaged to retrieve a total estimate. Mathematically,

$$\text{Centroid estimate:} \quad \hat{\mathbf{y}}_i = \frac{1}{7} \sum_{\substack{j=0, \\ j \neq i}}^7 \hat{\mathbf{y}}_i^j. \quad (3.2a)$$

Upon inspection of these averaged estimates, it was found that the estimates $\hat{\mathbf{y}}_i^j$ at times were very scattered. As illustrated by Figure 3.4, the centroid can sometimes be a bad estimate for a cluster’s center if the outliers are few and far away. The centroid, or average, minimizes the squared Euclidean distance to all data points. This means that outliers heavily influence the result. To compensate and reduce the influence of the mappings lying far away, the point which minimized the L_1 -norm was instead computed:

$$\text{L}_1\text{-norm estimate:} \quad \hat{\mathbf{y}}_i = \arg \min_{\mathbf{y}} \frac{1}{7} \sum_{\substack{j=0, \\ j \neq i}}^7 |\mathbf{y} - \hat{\mathbf{y}}_i^j|, \quad (3.2b)$$

where $|\cdot|$ denotes a summation of the absolute values, e.g: $|\mathbf{a}| = |a_1| + \dots + |a_n|$. This minimization problem was numerically solved with MATLAB’s intrinsic solver `fminunc`, which finds minimums to unconstrained multivariate functions. The quasi-newton method was used with the centroid estimate as starting guess.

Depending of which estimate of $\hat{\mathbf{y}}_i$ we choose, this allows us to define an estimate error:

$$e_i = \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|, \quad (3.3)$$

where $\|\cdot\|$ denotes the Euclidian norm.

Both of these estimators can now be optimized in terms of two parameters: the number of closest features N and the threshold for the affine transformation, t_{aff} , in the feature-based image registration. A grid-based search was conducted, to find parameters to minimize the error e_i and thus find the best estimates for \mathbf{y}_i . The maximum number of iterations in the RANSAC-algorithm was set to 100000 and the distances between matched features were minimized in terms of the L_2 -norm.

Finally, given optimal parameters, we can estimate the starting points in one of images using the others as source images. These estimated origins are then fed as starting points to the vessel tracking algorithm later described in this chapter.

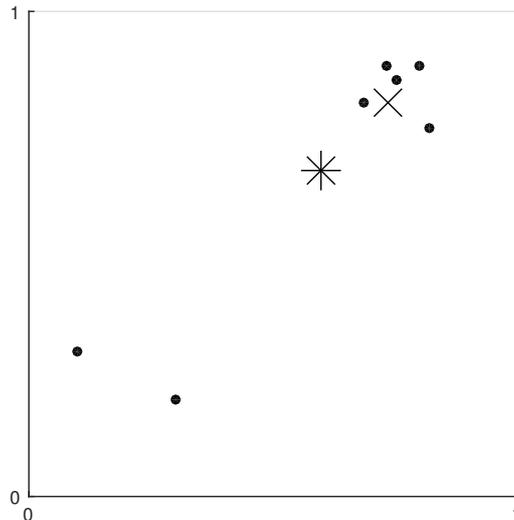


Figure 3.4: Illustration of how the centroid (average), marked as a star, can make a bad estimation for a cluster's center. In this figure, the two points far from the cluster centre influence the centroid greatly. In this figure, the numerically computed L_1 -estimator, marked as a cross, is preferred as an estimation of the cluster's center.

3.4 Applying the Frangi vesselness measure

The Frangi vesselness measure enhances tubular-like structures, as described in Section 2.3 and is the method chosen to enhance vessel structures in this thesis. Its main advantages are that it remains relatively simple compared to other vessel enhancement methods and as it has been implemented and optimized for MATLAB [28].

The Frangi vesselness measure has three parameters: α , β and c . It may be useful to note what values of the parameters other authors have used. In an example of tracking coronary arteries in the same data set as in this thesis, [14] uses $\alpha = \beta = 0.5$ and $c = 230$. Further, within the function documentation of [28], the value of c is suggested to be that of “dividing the greyvalues of the vessels by 4 [to] 6”.

It is important to note that we do not seek just an increase of the value of a vessel’s corresponding voxels; we seek to enhance the vessel structures, meaning an increase of local contrast around the vessels. Therefore, in order to measure the performance of a setting of Frangi filter parameters, it is not enough to simply look at the voxel values of the vessel (available through the gold standard).

Instead, using cubic interpolations, the values of “vessels” running parallel to the actual vessel were computed and finally averaged together, creating an average of the values around the vessel. The distances from the actual vessel to the interpolated parallel vessels was chosen to be that of the radius at the current point. In Figure 3.5, an example of how the Frangi vesselness measure varies at a vessel’s coordinates compared to its immediate neighborhood can be seen. Optimally, one would like the neighborhood (dashed blue) very close to zero and the values of the true vessel to be high and stable. Despite many attempts with different parameter settings, this was hard to accomplish.

Eventually, Frangi filter parameters were chosen by manually tweaking the original suggestion of [14] and by visually comparing the resulting vesselness along a true vessel in the training set with its immediate neighborhood, for instance by comparing the area beneath the graphs. The parameters can be found in Table 3.1.

Table 3.1: Chosen parameters for the Frangi filter.

Parameter	Value
α	0.5
β	0.6
c	60

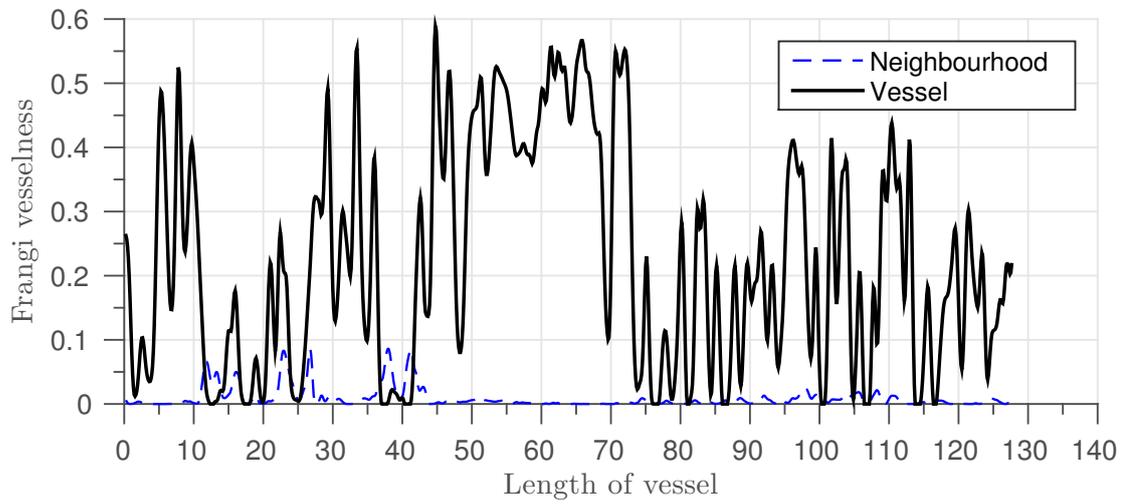


Figure 3.5: The Frangi vesselness measure for the positions of the right coronary artery in one of the images (black), and the vesselness in its immediate neighbourhood (dashed blue). The neighbourhood value is calculated as the average of 50 fictional vessels, running in parallel with the true vessel at the distance of its current radius.

3.5 Tracking the vessel

We now have estimates for the starting points of the coronary arteries (Section 3.3), and from the previous section a measure of every voxel’s “vesselness”. This section describes the algorithm by which the vessel is tracked. In short, it may be explained in four steps:

1. Performing Dijkstra’s algorithm in a local sphere, using a cost function based on the Frangi vesselness measure and pericardium distance.
2. Assessing the shortest paths generated by above algorithm, detecting bifurcations and choosing which path(s) to proceed with.
3. Incrementally performing step 1–2 again with a new starting point, until the tracked vessel(s) reaches a predetermined maximum distance.
4. Finally, selecting the best path as the estimated RCA centerline.

3.5.1 Dijkstra’s algorithm

We are now almost ready to track the vessel: Section 3.3 has provided us with estimated starting points, Section 3.4 a measure of how much vessel a voxel is and Section 3.2 serves us with the distance $D(\mathbf{x})$ to the pericardium, where \mathbf{x} is a position in the image. To continue, and to apply Dijkstra’s algorithm to this problem we only need two more things: a connected graph with edge weights (combining the information from the vesselness measure and the pericardium distance) and an end point to supply to the shortest-path algorithm.

Starting with the former of the two, let $G = (V, E)$ be the graph consisting of all voxels, each and everyone connected to their immediate neighbours. I.e. a voxel at index position (i, j, k) has neighbours at indices $(i \pm 1, j \pm 1, k \pm 1)$. This means that each voxel has six neighbours if it is not on a boundary, see Figure 3.6 for an illustration.

Further, let the edges E be defined as a matrix E where each element e_{ij} represents the cost of traversing from node i to j , taken as the average of the “cost” of being at either \mathbf{x}_i or \mathbf{x}_j : $e_{ij} = \frac{C(\mathbf{x}_i) + C(\mathbf{x}_j)}{2}$.

The choice of function $C(\cdot)$ to represent the cost deserves some consideration. It should include the vesselness measure, \mathcal{V}_o , as we in particular want vessels to have a low cost. Also, it should incorporate the information present in the performed pericardium segmentation – being outside of the pericardium should have a high cost. The following function was eventually defined:

$$C(\mathbf{x}) = \begin{cases} \log\left(\frac{1}{\mathcal{V}_o(\mathbf{x}) + \varepsilon_{\text{machine}}}\right) \cdot \left(1 + \left(\frac{D(\mathbf{x})}{10}\right)^5\right) & \text{if } D(\mathbf{x}) > 0 \\ \log\left(\frac{1}{\mathcal{V}_o(\mathbf{x}) + \varepsilon_{\text{machine}}}\right) & \text{otherwise.} \end{cases} \quad (3.4)$$

Explained from “the inside-out”, the ratio $\frac{1}{\mathcal{V}_o}$ is easy to motivate: a vessel has a high value of \mathcal{V}_o and thus the inverse is a lower number. To prevent numerical issues with values of \mathcal{V}_o being 0, the smallest floating number $\varepsilon_{\text{machine}}$ was added in the denominator. This ratio was then used as argument to the log-function. Being a monotonically increasing function, it maps extremely high values to what was perceived as a more manageable range of numbers. It should be noted that the argument to the log-function always was strictly greater than 1, as the values of \mathcal{V}_o never exceeded 1. Thus the costs $C(\mathbf{x})$ were always

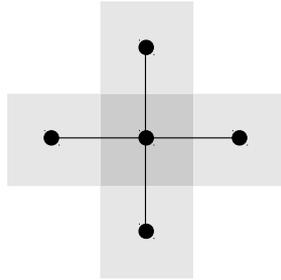


Figure 3.6: A voxel with 4-connectivity in 2D. For the 3D analogy, imagine another voxel on top and below, resulting in 6-connectivity.

positive. Finally, the distance to the pericardium was used as a penalty factor. If \mathbf{x} was estimated to be outside of the pericardium, i.e. $D(\mathbf{x}) > 0$, an extra factor was multiplied with. To compensate for inaccuracies in both gold standard and the estimation of $D(\mathbf{x})$, the value of $D(\mathbf{x})$ is treated through a simple filter: dividing by 10 and the resulting ratio taken to the power of 5. This means that voxels with $D(\mathbf{x}) < 10$ mm will have a very small penalty, but those outside of that will be heavily penalized. These numbers, 10 and 5, were derived by iteratively evaluating the performance of the algorithm, and adapting the numbers so that the algorithm did not track vessels outside of the pericardium.

Now, we need to address the issue with supplying an end point to Dijkstra’s algorithm. If one would know the initial direction of the vessel, or even the region of where it terminates, this would be ideal end points. However, this information is not incorporated in the proposed method as it would make the algorithm less automatic. Instead, we propose to consider *all* points on the spherical surface with radius $r = 20$ mm around the starting point $\hat{\mathbf{x}}_i$.

Using this “spherical approach”, a large set of shortest paths will be retrieved $\{p_i\}_{i=1}^m$: one for every end point on the sphere’s surface. Roughly speaking, a sphere with radius 20 mm has a surface area of approximately 5000 mm^2 . Assuming voxel dimensions of $[1 \times 1 \times 1] \text{ mm}^3$ and a neat 1-to-1 correspondence between every single square-millimeter and voxels, we would have 5000 shortest paths to consider. One can thus safely assume that this proposed implementation of the Dijkstra’s algorithm generates thousands of shortest paths. It should be noted that Dijkstra’s algorithm copes well with more than end point: in fact, it utilizes the information gained when finding the shortest path to a node t and a neighbouring node t' .

In the following section, the method in which these paths are assessed is presented.

3.5.2 Assessment of shortest paths

The method in the previous section provides us with a set of shortest paths from the starting point to all other voxels on the surface of a surrounding sphere. To assess which of these paths that are potential vessels and candidates for our the real coronary arteries, the paths undergo an assessment:

Firstly, paths crossing the surface of the sphere are rejected. To cross the sphere one needs to, per definition, pass a node that is on the surface. Then that node is already included as being one of the end points in another path. Thus, paths overriding this criteria are paths that try to reach one point on the surface of the sphere but “cheat” and arrive at it from the wrong direction.

Secondly, paths ending close to the previously used starting point are rejected. This is not relevant in the very first stage of the algorithm, but in later stages when a vessel is tracked this prevents backtracking.

Lastly, in the very first step of the vessel tracking, the paths are evaluated by means of assessing their perpendicular distance to the pericardium. Paths that go “into the heart” are rejected. In practice, the paths were limited to be at least within 15 mm from the pericardium after 40 mm, numbers derived by inspection of Figure 3.3 (p. 17). This prevents the algorithm from picking up a false lead in the beginning (perhaps due to an inaccurate starting point estimate), and helps it find vessels leading out to the pericardium.

The remaining paths are then sorted according to a normalized score:

$$\text{Path Score for path } q = \frac{\text{total cost of path } q}{\text{length of path } q}, \quad (3.5)$$

where the cost of a path is the sum of all edge costs in that path.

The path with the best normalized score is most likely a vessel, and is thus a very good candidate for our vessel segmentation. However, if we choose the best path with the best score in every iteration, this limits the algorithm as it does not take bifurcations into account. Instead, a k -means cluster analysis (see Section 2.4) with $k = 3$ of the 30 best paths’ end points is performed.

If the cluster analysis only finds one cluster this is a sign that the majority of the 30 best paths lie around or very nearby the same vessel. If instead a substantial part of the 30 end points are within different clusters, this is most likely a bifurcation of the vessel. Performing this cluster analysis of the best paths’ end points allows the algorithm to cope with bifurcations, and pursue new vessels. A threshold of 35 % was set to mark if a cluster is relevant or not. I.e. if 35 % or less of the best paths’ end points are in the same cluster, they are rejected. This number was derived by assuming a presence of three clusters (remember $k = 3$), and in that case guaranteedly rejecting one of them.

Finally, we arrive at the last step of this part of the vessel tracking: having found likely end points to be a continuation of the vessel, it is tempting to use one of these points, or an average of them, as new starting points for another step in the algorithm. However, as these points most likely were in a cluster, the next algorithm start point is instead a point retrieved by backtracking from the cluster’s center, heuristically taken as distance of a third of the original sphere’s radius, i.e. $\frac{20}{3}$ mm. This should make for a more robust vessel tracking.

If a bifurcation was detected, it was noted and pursued recursively after the first path was terminated. Also, a maximum number of allowed bifurcations was set to reduce algorithm run time. For this vessel tracking of the RCA, it was taken to be 5.

3.5.3 Termination

Judging when a tracked vessel terminates using the Frangi vessel measure or the cost function proved to be difficult. As can be seen in Figure 3.5 (p. 21), the measure fluctuates heavily and approaches zero even though when the vessel is not about to terminate. Instead, a hard-coded termination condition was implemented: if the length of the tracked vessel exceeded 300 mm, it was terminated. As of Figure 1.3 (p. 4), it can be seen that no RCA in the training set exceeded 225 mm, thus 300 mm is definitely enough to cover them all. Further, the initial thought of this limit was that it would also be enough to cover the

vessels in the *test set* aswell (where the true lengths are unknown). Also, it is motivated to track longer than shorter, if a vessel has been tracked too long, it could perhaps be assessed retroactively and be cut at the appropriate point (not part of this thesis).

Finally, the segmented vessel (to be evaluated) was taken as the concatenated path with the totally best normalized score, as defined by Equation (3.5).

3.5.4 Summary

The proposed procedure is summarized in Algorithm 3, with parameters defined in Table 3.2.

Algorithm 3: Main vessel-tracking algorithm.	
Input:	Edge costs defined by matrix $C(\mathbf{x})$, starting point s
Output:	Estimated centerline position
1	Initialization: $\{P_i\}$, empty path structure(s), $i = 1, P_1 = s$
2	while $i \leq 5$ do
3	Generate sphere of end points $\{t\}$, radius r from s ;
4	$p_i \leftarrow$ Shortest paths from s to $\{t\}$ using $C(\mathbf{x})$, sorted by normalized score ;
5	Reject illegal paths in p_i ;
6	k -means clustering on the end points of $p_1 - p_{30}$;
7	for all clusters found significant: $P_{i+1} \leftarrow P_{i+1} +$ (best path in p within that cluster (backtracked));
8	if length of $P_i > 300$ mm then
9	$i \leftarrow i + 1$
10	else
11	// do nothing
12	end
13	$s \leftarrow$ current end point of P_i ;
14	end
15	return The path in P_i with best normalized score

Table 3.2: Algorithm parameters.

Parameters	Value
Radius r	20 mm
# of path endpoints used in cluster analysis	30
# of clusters k , initialization	3
Bifurcation threshold	35 %
Maxmimum number of bifurcations, RCA	5
Maxmimum length RCA	300 mm

3.6 Evaluation

As we have eight images in the training set with available gold standard, these can be used when measuring the performance of the feature-based image registration of the origins of the coronary artery using leave-one-out cross validation. We should also note if the proposed incremental shortest-path algorithm tracks the correct artery (at least from the start) given the estimated starting point.

Further, to evaluate the resulting segmentation for the positions of the coronary arteries we have the standardised evaluation framework as presented in Section 1.3.2. Also, we may use the gold standard available to show the deviations from the true position as a function of tracked length.

Only evaluation of the tracking of the RCA is considered. This is due to the more major bifurcations of the LAD, where it bifurcates in e.g. the LCX (which is also in the Rotterdam Coronary Artery Algorithm Evaluation Framework-challenge). The proposed method rejects bifurcations and finds the best total path and is thus not suited for tracking a vessel that bifurcates. Due to this, no evaluation was done on the test set, as one would there require to submit estimated positions to the RCAAEF for not only the RCA, but the LAD, LCX and the fourth biggest coronary artery after these three.

3.7 Implementation

The proposed method was implemented in MATLAB. To save time when doing parameter optimizations and debugging, the preprocessed, intermediate and resized versions of the 3D images were saved and preloaded in a temporary data folder as binary files with single precision. To distinguish same images with different parameters, the file were named through hash-function³ of the image, and parameters themselves.

Implementing the shortest-path algorithm in as of Section 3.5 (p. 22) is a computational challenge. Considering the size of the preprocessed images used in this thesis (Section 1.3, p. 2), enormous edge matrices E need to be generated. The average size of the images is $256 \times 256 \times \sim 144$, rendering approximately 10 million voxels in total. In other words, E would be a matrix with just as many rows *and* columns. Even when represented as a data structure of a sparse matrix, this proved to be way too much for the computer memory. Instead, the problem was addressed by implementing the shortest-path algorithm locally in a “box” containing the starting point and end points scattered on the spherical surface. The shortest paths computed there was then transferred back to the original CT volume.

3.7.1 Run times

Table 3.3 summarizes the run times for the most important parts of the proposed method. All run times are retrieved from a single-core computation with MATLAB R2014b, using a PC with Linux Redhat 6.6 (Santiago) distribution configured with Interl i5 2.8 GHz, 64-bit architecture and 16GB RAM.

Table 3.3: Run times of key parts of the algorithm.

Component	Run time
Preprocessing	2 min
Feature detection of a new image	1 min
Feature-based image registration to predict starting point, one vessel	4 min
Applying the Frangi filter	2 min
Tracking RCA	5 min

³DataHash: MD5 or SHA-1 for array, struct, cell or file. Available at: <http://www.mathworks.com/matlabcentral/fileexchange/31272-datahash>. Updated: 29 Feb 2016

4

Results

In this chapter the results derived from the method are presented.

4.1 Estimation of starting points

A leave-one-out grid-based parameter optimization was performed on the training set to find optimal values of N and t_{aff} in order to minimize the estimation error e_i in the feature-based image registration. The optimization was done in separate for the origins: the starting point of the RCA and LAD. Also, it was performed either using the centroid estimate or the L_1 -estimator. This generated four surface plots, which can be found in Figure 4.1.

From Figure 4.1 a good selection of parameters for the feature-based registration may be derived. The error is minimized for a high number of attempted matched features, N , and a rather low inlier/outlier threshold. The parameters that best optimize respective vessel and estimator can also be found in Table 4.1.

The L_1 -estimator performs better than the centroid-estimate. The average error can be compared with the average vessel radius at the starting point, as shown by Figure 1.3 (p. 4), the radius is approximately 2.5 mm in the start.

Table 4.1: Minimum error for centroid and the L_1 -estimator method respectively, when estimating the starting point of the RCA and LAD respectively.

Vessel	Method	Error [mm]	Parameters
RCA	Centroid	10.7	$N = 275, t_{\text{aff}} = 8$
	L1	4.8	$N = 260, t_{\text{aff}} = 8$
LAD	Centroid	4.8	$N = 245, t_{\text{aff}} = 8$
	L1	4.5	$N = 230, t_{\text{aff}} = 10$

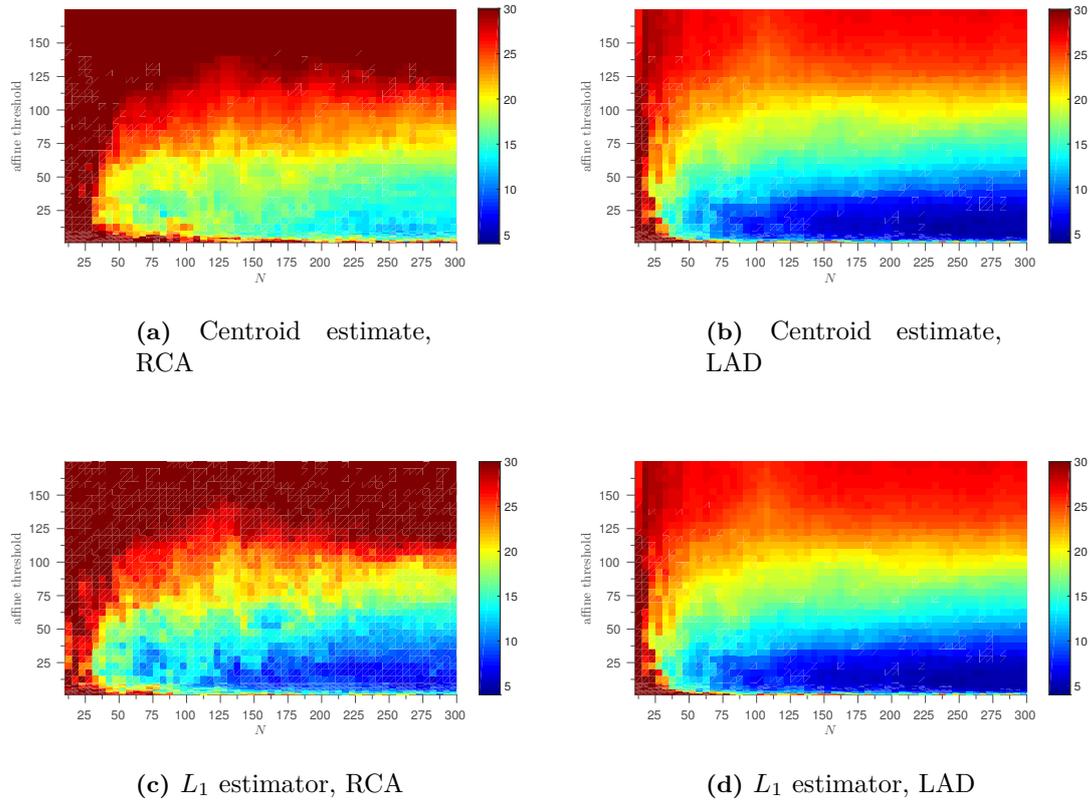


Figure 4.1: Average error of the estimation of coronary artery starting points as a function of the number of used features N and inlier/outlier threshold t_{aff} in RANSAC (less is better). Left column: the RCA. Right column: the LAD. The color map is the same in all sub figures, dark blue corresponding to roughly 5 mm in average error, and red: an error of 30 mm or more.

4.2 Tracking the vessel

The vessel tracking algorithm as of Section 3.5 was applied to all eight training images. Using the starting points generated by the feature-based image registration described in Section 3.3, it was manually verified that the algorithm correctly detected and tracked the beginning of every RCA in the training set. A visual result for one of the trackings of the RCA can be found in Figure 4.2, where the algorithm successfully follows the artery, but separates from the true artery at a turn.

The vessel tracking results can also be viewed in the upper subfigure of Figure 4.3, where the deviations from the gold standard is shown. Optimally, one would to have zero deviation throughout the vessel’s length, but this is not the case here. For images 1, 2, 3 and 5, less than 100 mm of the vessel is tracked. At e.g. 30 mm, there is a “break-away” in image 3 and the estimated position of the coronary artery is no longer the true vessel. Applying the algorithm on image 4 generates the best results: it is successfully tracked within an error of 1.5 mm all the way up up roughly 180 mm, where the estimated position deviates.

In the lower subfigure of Figure 4.3, fluctuations from the true vessel are shown using a narrower y -scale. These fluctuations can be compared to the true radius of the RCA in Figure 1.3 (p 4), which means that most of the estimated vessels deviate outside of the true vessel more than once before finally “breaking away” from the true vessel.

In Figure 4.4, the distance to the pericardium for the estimated centerline positions can be seen. After initially “climbing” up to the surface almost all estimated centerlines remain there before being terminated. Image 2’s vessel tracking has been terminated prematurely, most likely as it went “into the heart” in the start.

Finally, and perhaps most importantly, are the results obtained by the standardised evaluation procedure described in Section 1.3.2. See Table 4.2 and 4.3 for a comparison of the results when evaluating the performance on the training set for the proposed method and the ten other automatic methods presented in the RCAAEF-challenge. The performance of the proposed method is worse than all other submissions in the RCAEFF, both in terms of overlap measure (“OV”), “score” and in accuracy (neglecting the two second last methods in Table 4.2 that seem to be missing scores).

Table 4.2: Evaluation scores for *automatic* methods participating in the RCAAEF challenge and the proposed method on the training set. OV = overlap, OF = overlap until first error, OT = overlap in clinically important regions. “%” = overlap percentage compared to the reference standard, Score = weighed performance compared to experts’ annotations, Rank = comparative number, calculated by the RCAAEF and not by the locally implemented evaluation framework on its own, hence “n/a” for the proposed method. Please refer to Section 1.3 for more precise definitions of these acronyms and measures.

Method	OV			OF			OT		
	%	Score	Rank	%	Score	Rank	%	Score	Rank
ModelDrivenCenterline [7]	92.4	54.3	10.19	80.6	58	7.59	93.4	65.3	8.25
SupervisedExtraction [8]	91.2	53.7	11.28	84.1	62.9	8	93.6	68.1	8.47
COR Analyzer [29]	86.8	55	12.09	76.1	54.2	9.53	88.2	60.8	10.59
DepthFirstModelFit [9]	89.2	52.2	11.44	78.3	60.3	7.88	93	67.3	7.84
CocomoBeach [30]	80.2	44.3	14.63	66.4	41.7	11.38	81.6	48.1	13.97
TwoPointMinCost [31]	93.3	71.2	6.81	67.8	57.8	7.91	93.5	72.5	6.88
GFVCoronaryExtractor [11]	90.3	50.7	11.94	74.6	57.6	8.75	94	64.6	8.53
GVFTube’n’Linkage [10]	89.6	53	12	66.6	45.3	10.84	91.5	58.4	10.66
AutoCoronaryTree [32]	0	0	18.97	0	0	18.97	0	0	18.97
VirtualContrast [33]	0	0	18.97	0	0	18.97	0	0	18.97
Proposed method	53.5	29.05	n/a	20.9	11.52	n/a	60.0	30.0	n/a

Table 4.3: Accuracy in millimeters for *automatic* methods participating in the RCAAEF challenge and the proposed method on the training set. The accuracy is the average of the deviations from the reference standard given a successful tracking. Please refer to Section 1.3 for a more precise definition.

Method	Accuracy [mm]
ModelDrivenCenterline [7]	0.21
SupervisedExtraction [8]	0.27
COR Analyzer [29]	0.31
DepthFirstModelFit [9]	0.31
CocomoBeach [30]	0.32
TwoPointMinCost [31]	0.48
GFVCoronaryExtractor [11]	0.39
GVFTube’n’Linkage [10]	0.39
AutoCoronaryTree [32]	0
VirtualContrast [33]	0
Proposed method	0.57

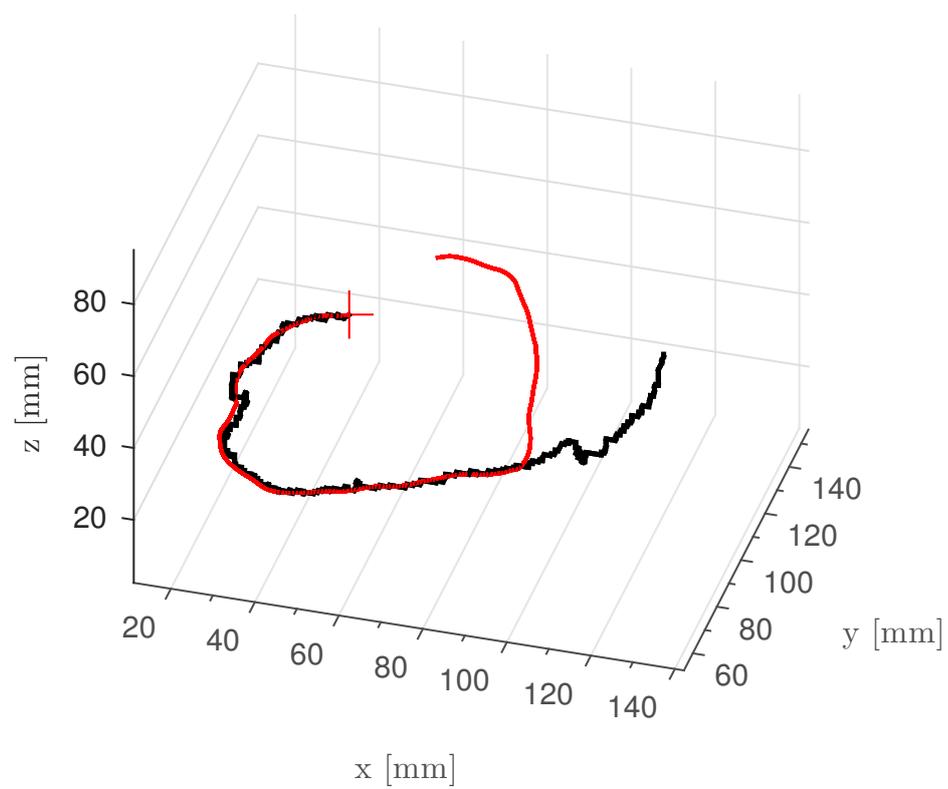
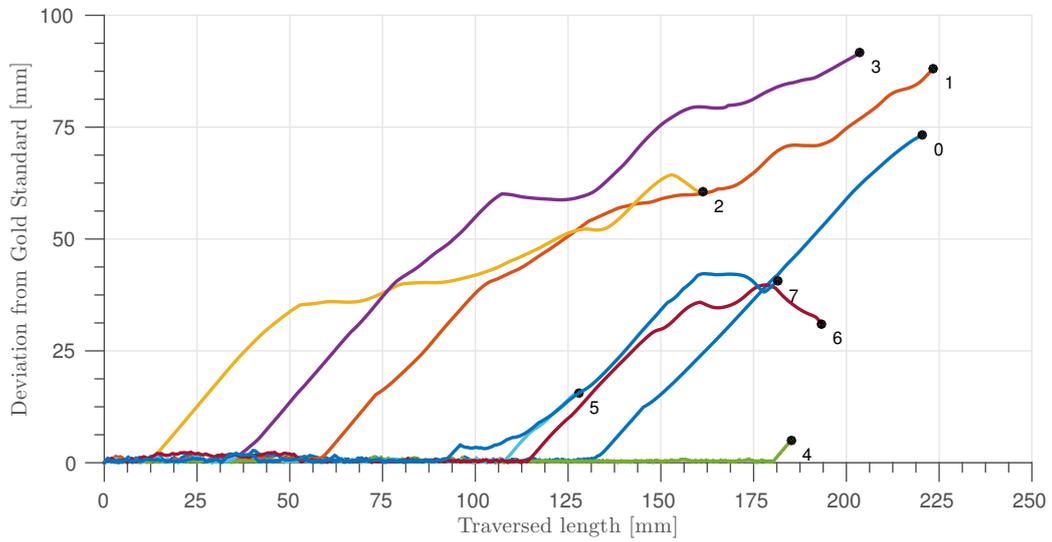
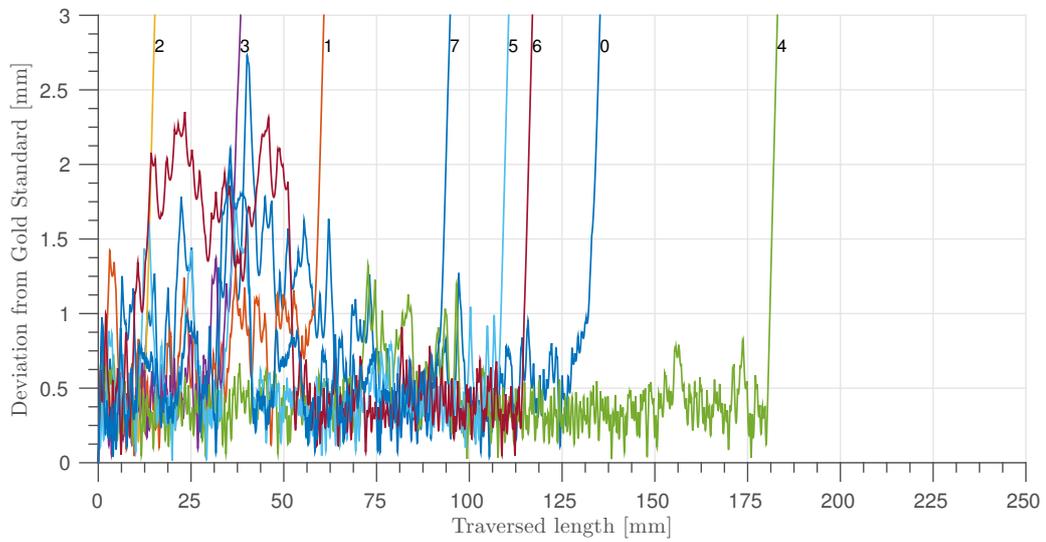


Figure 4.2: Visualization of one of the estimated centerlines of the RCA in one of the train images. The thick black line is the estimated centerline, and the red lines is the gold standard. The '+' denotes the starting point.



(a)



(b) Narrower y -scale.

Figure 4.3: Deviation of the estimated vessel centerlines from the from the gold standard as a function of traversed distanced in the gold standard, for each train image respectively. The small black digits represent the image number in the training set (0-7)

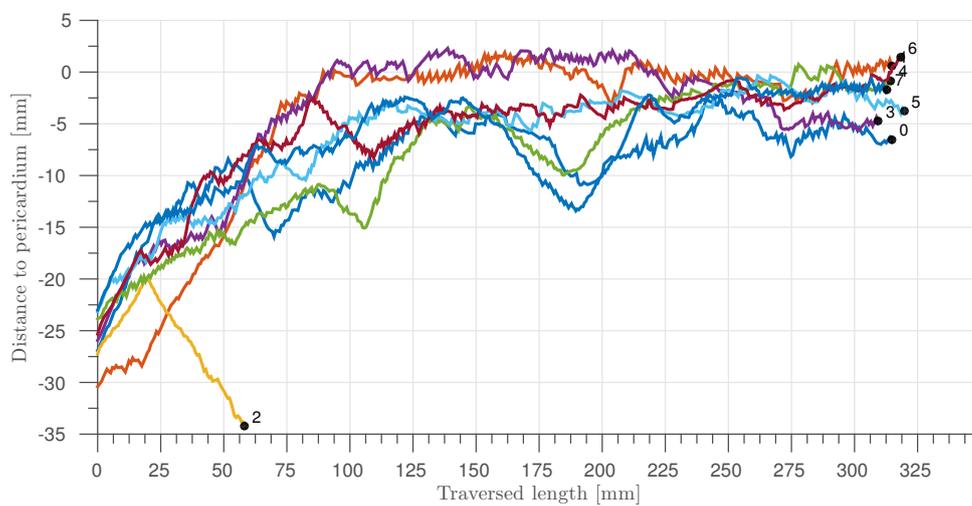


Figure 4.4: Distance to the pericardium in [mm] as a function of traversed length is the estimated centerlines of the RCA for different data sets (0-7). The small black digits represent the image number in the training set (0-7).

5

Discussion

In this chapter the results and outcome of this thesis are discussed. Also, thoughts and ideas for possible future work is presented.

5.1 Estimation of starting positions

Estimating the origins of the right coronary artery (RCA) and left anterior descending artery (LAD) using a feature-based image registration approach was quite successful: the estimates were close enough for the vessel tracking algorithm to start tracking the correct vessel. Also, the average error of estimating the position was 5 mm, which can be compared to the radius of the RCA at its origin: 2.5 mm. I.e. the average error was roughly the diameter of the RCA origin.

To improve this method, one could use the information contained in the pericardium segmentation to reject certain estimates, before creating the total estimator. Although computationally expensive, one could also incorporate a more complicated transformation (a “warping”) to have an even better estimated starting point. Further, the method would become more robust with more images. Thus it could be very useful to have an expert annotate the origins of the RCA and LAD in new images, which should not be as time consuming as manually tracking entire vessels.

Interestingly, the average error when using the centroid-estimate for the starting point of the RCA is much larger than when applying the same estimator for the LAD. Reasons for this have not been investigated, but it might be due to anatomical differences between the starting point of the RCA and LAD.

5.2 Tracking the vessel

Estimating the positions of the centerlines proved to be a difficult task. The main contributions to the results are listed and discussed below.

The Frangi vesselness measure is a simple yet unpredictable vesselness measure. It performs worse when vessels do sharper turns and bifurcates. Further, it is sensitive to parameter settings, and is time consuming to optimize. As shown in Figure 3.5 (p. 21), it is also quite erratic along a true vessel. Despite being almost always higher than its immediate surrounding, it is believed that these sudden drops in the measure, e.g. at ~ 40 mm, causes the algorithm to consider other possible vessels in its vicinity. Nevertheless, the concept of comparing a vessel-enhancement filter’s response at the position of the gold standard and with the its immediate neighbourhood may be a useful tool when evaluating other filters or vessel detection methods.

Further, no optimization of the parameters of the Frangi vesselness measure (α, β, c) or the parameters of the vessel tracking (e.g. the cost-function C , the radius r , k , the bifurcation threshold etc.) has been made. Perhaps the results could be improved by such an optimization, although the risk of overfitting is quite large considering the number of algorithm steps, parameters and the small size of the training set.

Another reason for the proposed method's poor performance could be its "lack of trust". It does not really pursue paths that might be more rewarding "down the road". The concept with clustering makes the algorithm somewhat "smarter" and able to make a better decisions on where to continue other than purely choosing by the best normalized score. But, as seen in the results, the algorithm tended to deviate and find vessels that were not coronary arteries. In a sense this is also due to the erratic behaviour of the vesselness measure. Other than applying another measure, perhaps the algorithm could be improved by having it locally investigating the radius to make the decision in which vessel to pursue.

Further, the proposed algorithm contains plenty of parameters. The one that seemed to affect the end result the most was the choice of the sphere's radius, r . This was noticed at locations where a bifurcation was nearby: then the choice of a smaller or larger radius would mean the difference of detecting the bifurcation or not. If the radius was larger, the algorithm would take a "safer" step forward, as the bifurcation might be a false lead and perhaps end prematurely or deviate outside of the pericardium. Thus it might prove meaningful to investigate a method with a dynamic radius of the sphere.

Lastly, the proposed algorithm could be improved by increasing the connectivity between voxels. This could also be done retroactively, as a post-processing step. In other words, the estimated centerline path could undergo local improvements, where e.g. the connectivity of every voxel is improved, and perhaps even by allowing intermediate positions between voxels using interpolation. As the size of the radius of coronary arteries is small (<2.5 mm), the result of the proposed method is most likely affected by the rather crude network of possible voxels. For example, the radius of the vessels is only 3-4 voxels big at the starting point.

6

Conclusion

This thesis presented a way to estimate the origins of the right coronary artery (RCA) and the left anterior descending artery (LAD) by means of the feature-based image registration method in [15]. This proved quite successful, with an average error of 5 mm when applied to the eight images in the training set of the Rotterdam Coronary Artery Algorithm Evaluation Framework (RCAAEF).

Furthermore, this thesis pursued to contribute to the RCAAEF-challenge of estimating the centerline positions of the coronary arteries. Using a shortest-path approach together with a pericardium segmentation, this proved only moderately successful. The main reason is believed to be due to the definition of path cost. Having the path cost mainly defined by the Frangi vesselness measure caused tracking of vessels deviate from the real vessel. When the proposed method was applied to the training set of the RCAAEF, the RCA was tracked with an average accuracy of 0.6 mm and an average overlap of 53 %, way less than other automatic methods contributing in the RCAAEF. Nevertheless, this thesis might provide inspiration for other shortest-path approaches to detect coronary arteries.

Bibliography

- [1] “World Health Organization. Top 10 causes of death. Fact Sheet N310.”
<http://www.who.int/mediacentre/factsheets/fs310/en/>. Updated: May 2014.
- [2] O. Sand, Ö. V. Sjaastad, E. Haug, K. C. Toverud, and I. Bolinder-Palmér,
Människans fysiologi. Liber, 2004.
- [3] G. Bergström, G. Berglund, A. Blomberg, J. Brandberg, G. Engström, J. Engvall,
M. Eriksson, U. Faire, A. Flinck, M. Hansson, *et al.*, “The Swedish
CARDiopulmonary BioImage Study: objectives and design,” *Journal of internal
medicine*, vol. 278, no. 6, pp. 645–659, 2015.
- [4] H. Gray, *Anatomy of the human body*. Lea & Febiger, 1918.
- [5] M. Schaap, C. T. Metz, T. van Walsum, A. G. van der Giessen, A. C. Weustink,
N. R. Mollet, C. Bauer, H. Bogunović, C. Castro, X. Deng, *et al.*, “Standardized
evaluation methodology and reference database for evaluating coronary artery
centerline extraction algorithms,” *Medical image analysis*, vol. 13, no. 5,
pp. 701–714, 2009.
- [6] “Rotterdam coronary artery algorithm evaluation framework.”
<http://coronary.bigr.nl/>. Accessed: 2016-01-15.
- [7] Y. Zheng, H. Tek, and G. Funka-Lea, “Robust and accurate coronary artery
centerline extraction in cta by combining model-driven and data-driven
approaches,” in *Medical Image Computing and Computer-Assisted
Intervention–MICCAI 2013*, pp. 74–81, Springer, 2013.
- [8] Y. Kitamura, Y. Li, and W. Ito, “Automatic coronary extraction by supervised
detection and shape matching,” in *Biomedical Imaging (ISBI), 2012 9th IEEE
International Symposium on*, pp. 234–237, IEEE, 2012.
- [9] S. Z. et al., “Shape and appearance models for automatic coronary artery tracking.”
http://coronary.bigr.nl/centerlines/pdf/123_zambal.pdf. Retrieved:
2016-02-15.
- [10] C. Bauer and H. Bischof, “Edge based tube detection for coronary artery centerline
extraction,” *The Insight Journal*, 2008.
- [11] G. Yang, P. Kitslaar, M. Frenay, A. Broersen, M. J. Boogers, J. J. Bax, J. H.
Reiber, and J. Dijkstra, “Automatic centerline extraction of coronary arteries in
coronary computed tomographic angiography,” *The international journal of
cardiovascular imaging*, vol. 28, no. 4, pp. 921–933, 2012.

- [12] Y. Zheng, M. Loziczonek, B. Georgescu, S. K. Zhou, F. Vega-Higuera, and D. Comaniciu, “Machine learning based vesselness measurement for coronary artery segmentation in cardiac ct volumes,” in *SPIE Medical Imaging*, pp. 79621K–79621K, International Society for Optics and Photonics, 2011.
- [13] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, “Multiscale vessel enhancement filtering,” in *Medical Image Computing and Computer-Assisted Intervention—MICCAI’98*, pp. 130–137, Springer, 1998.
- [14] P. Strandmark and F. Kahl, “Curvature regularization for curves and surfaces in a global optimization framework,” in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 205–218, Springer, 2011.
- [15] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson, “Improving robustness for inter-subject medical image registration using a feature-based approach,” in *International Symposium on Biomedical Imaging*, 2015.
- [16] “Oxford english dictionary.” <http://www.oxforddictionaries.com/>. Retrieved: 2016-05-03.
- [17] B. Zitová and J. Flusser, “Image registration methods: a survey,” *Image and Vision Computing*, vol. 21, no. 11, pp. 977 – 1000, 2003.
- [18] F. Khalifa, G. Beache, G. Gimel’farb, J. Suri, and A. El-Baz, “State-of-the-art medical image registration methodologies: A survey,” in *Multi Modality State-of-the-Art Medical Image Segmentation and Registration Methodologies* (A. S. El-Baz, R. Acharya U, M. Mirmehdi, and J. S. Suri, eds.), pp. 235–280, Springer US, 2011.
- [19] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [20] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, “Speeded-up robust features (surf),” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346 – 359, 2008. Similarity Matching in Computer Vision and Multimedia.
- [21] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [22] D. Barber, *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.
- [23] J. Kleinberg and É. Tardos, *Algorithm design*. Pearson Education India, 2006.
- [24] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [25] “graphshortestpath – Solve shortest path problem in graph.” <http://se.mathworks.com/help/bioinfo/ref/graphshortestpath.html>. Accessed: 2016-04-15.
- [26] O. Friman, C. Kühnel, and H.-O. Peitgen, “Coronary centerline extraction using multiple hypothesis tracking and minimal paths,” in *Proc MICCAI*, 2008.

- [27] A. Norlén, “Automated segmentation of the pericardium using a feature based multi-atlas approach,” 2014. Student Paper.
- [28] D.-J. Kroon, “Hessian based frangi vesselness filter.”
<http://www.mathworks.com/matlabcentral/fileexchange/24409-hessian-based-frangi-vesselness-filter>. Accessed: 2016-02-15.
- [29] R. M. S. Ltd, “Cor analyzer system.”
<http://www.rcadia.com/page.php?pageID=23>. Retrieved: 2016-05-22.
- [30] P. H. Kitslaar, M. Frenay, E. Oost, J. Dijkstra, B. Stoel, and J. Reiber, “Connected component and morphology based extraction of arterial centerlines of the heart (cocomobeach),” in *The MIDAS Journal: MICCAI Workshop–Grand Challenge Coronary Artery Tracking*, Citeseer, 2008.
- [31] C. Metz, M. Schaap, T. Van Walsum, and W. Niessen, “Two point minimum cost path approach for cta coronary centerline extraction,” *The Insight Journal*, 2008.
- [32] H. Tek and M. A. G. et al, “Automatic coronary tree modeling.”
http://coronary.bigr.nl/centerlines/pdf/137_tek.pdf. Retrieved: 2016-05-22.
- [33] C. Wang and O. Smedby, “An automatic seeding method for coronary artery segmentation and skeletonization in cta,” *The Insight Journal*, pp. 1–8, 2008.

