Coronary artery extraction from CT images

Richard Nordenskjöld

April 2, 2009
Master's Thesis in Computing Science, 30 ECTS credits
Supervisor at CS-UmU: Fredrik Georgsson
Examiner: Per Lindström

Umeå University
Department of Computing Science
SE-901 87 UMEÅ
SWEDEN

Abstract

A semi-automatic method that segments coronary arteries from CT images, and visualizes them on a 2D representation of the left ventricle, was realized. The results were obtained by following the ascending aorta down to the left ventricle. An ellipsoid approximation of the left ventricle was made and used for the 2D projection. The coronary arteries were located by looking around the border of the approximated ellipsoid. By doing this, medical staff can get a good view of the coronary artery tree without spending time looking in a CT image set.

Contents

1 Introduction **1**

1.1 CT image generation 1

1.2 Outline 2

2 Problem Description **3**

3 Methods for coronary artery measuring **5**

3.1 Segmentation 5

3.1.1 Region growing 5

3.1.2 Thresholding 7

3.1.3 Voxel mapping 7

3.1.4 CT segmentation problems 8

3.2 Centerline extraction 8

3.2.1 Direct methods 8

3.2.2 Skeleton based methods 8

3.3 Artery measuring 9

3.3.1 Watershed 9

3.3.2 Region growing 10

3.3.3 Resegmentation 10

3.3.4 Measuring 10

4 Development **11**

4.1 Parsing 11

4.2 Aorta identification 12

4.3 Aorta segmentation 13

4.3.1 Circle approximation 13

4.3.2 Region growing 14

4.3.3 Lines from Center 14

4.3.4 Region growing with object extraction 15

4.3.5 Final version 16

4.4 Left ventricle optimization 16

4.4.1 Point placement . . . 16
4.4.2 Optimization . . . 17
4.4.3 Error detection . . . 19
4.4.4 Manual localization . . . 19
4.5 Artery segmentation . . . 20
4.5.1 Thresholding . . . 21
4.5.2 Surrounding voxels . . . 21
4.5.3 Path walking . . . 21
4.6 Projection . . . 21

5 Planning and methods **23**
5.1 Preliminary planning . . . 23
5.2 How the work was done . . . 23

6 Results and discussion **25**
6.1 Program . . . 25
6.1.1 User interface . . . 25
6.1.2 Human interaction . . . 26
6.1.3 Resulting plot . . . 27
6.2 Algorithms . . . 27
6.2.1 Aorta identification . . . 27
6.2.2 Aorta segmentation . . . 28
6.2.3 Left ventricle optimization . . . 28
6.2.4 Artery segmentation . . . 28
6.2.5 Projection . . . 29

7 Conclusion **31**
7.1 Limitations . . . 31
7.2 Future work . . . 32

8 Acknowledgments **33**

References **35**

List of Figures

1.1 A slice from a CT image stack 2

2.1 An empty bullseye 4

3.1 Region growing. Green pixels are old seeds and black pixels are new seeds 6
3.2 Example of thresholded image showing all contrast filled objects. Coronary arteries are circled. 7
3.3 Geometric shapes with skeletons marked black 8

4.1 Top slice of a CT image. AA = Ascending Aorta, DA = Descending Aorta 12
4.2 Image with the aortic root marked with a black circle 14
4.3 Point placement while above the approximated centerpoint. Points are marked with gray squares 17
4.4 Figure showing how the manual points should be placed on the left ventricle 19

6.1 Screenshot of the program GUI while segmenting the arteries 25
6.2 Image showing where to place the point on the septum border 26
6.3 Image of the left ventricle showing where to place points 26
6.4 A result of running the program. Threshold set to 250 HU 27

List of Tables

4.1 Substances with their corresponding Hounsfield Units 12

Chapter 1

Introduction

This chapter is an introduction to this thesis and gives information required to lift the reader to an appropriate level of knowledge.

The coronary arteries are the arteries that supply the heart with blood [10]. If they get affected by stenosis, the heart will be affected and it could be fatal [3]. This is why it is important to find these narrowing parts of the arteries before they get to serious. This can be done with a medical imaging technique called computed tomography (CT). This technique allows the coronary arteries to be studied without the need of invasive catheters, but it can be hard to evaluate how these arteries supply the myocardium while looking at the CT generated images. This is why an efficient way of visualizing the arteries is needed.

The topic of this thesis is the creation of an application that extracts coronary arteries from a set of CT images and presents them on a standard called "bullseye". A bullseye is a 2D plot. It represents the outer left ventricle wall unfolded, and is divided into 17 sectors. By projecting and displaying the coronary arteries on this bullseye, medical staff can quickly get a grip of how much blood the heart is supplied with.

This project is part of a larger project which has the goal of expressing coronary arteries as splines on a bullseye, in order to allow a standard way of looking for defects on the arteries.

All work is conducted at Uppsala University Hospital for the department of oncology, radiology and clinical immunology (ORKI).

1.1 CT image generation

A CT scan is done by first, if needed for the images wanted, injecting contrast agent into the patients blood stream. This will block the radioactive rays emitted from the CT scanner, creating bright areas in the finished CT image. The CT scanner works by rotating around a gantry while transmitting radioactive rays. A receiver located on the opposite side registers with what intensity the rays are received on the opposite side. While the scanner rotates, a board on which the patient lies is moving slowly into the gantry. This means that the scanner is moving in a spiral pattern relative to the patient. Cross sections of the patients body can be created from the information obtained by scanning. A set of cross sections (normally called slices) can be generated and put together into a stack. This stack can then be used to show information of the body's insides in 3D.[27]

Figure 1.1: A slice from a CT image stack

1.2 Outline

This report comprises the following parts:

Chapter 2 - Problem description states the problem that this thesis handles.

Chapter 3 - Methods for coronary artery measuring is an in-depth study of existing methods in coronary artery measuring.

Chapter 4 - Development describes the developed methods and how they work.

Chapter 5 - Planning and methods describes how the work was conducted.

Chapter 6 - Results and discussion describes the results of this project and how it is effected by different parameters.

Chapter 7 - Conclusion gives information on the quality of the results.

Chapter 8 - Acknowledgments is where all credits are given.

Chapter 2

Problem Description

This chapter contains the description of the project, it's goals and it's application area.

A project with the goal to digitally visualize patient health information has started at Uppsala University Hospital. A part of this project is the visualization of the coronary arteries projected onto a so-called bullseye plot. This bullseye is a 2D representation of the left ventricle and is used in order to more easily see which sections of the heart is supplied by the coronary arteries. Figure 2.1 on the following page illustrates an empty bullseye divided into regions which represents different sections of the left ventricle.

The goal for this project is to find the coronary arteries from a CT image set. The found arteries then have to be measured in order to visualize stenosis and other factors that affect the blood flow supplying the heart. After this is done the data of the segmented arteries has to be reduced and projected onto the bullseye. In order for this to happen, the orientation of the left ventricle is needed. The projected points are then the output of this program and are later visualized by an already existing application in the larger project.

The important qualities for this program is that it has to be rather fast, because medical staff has little time to spare for the production of these artery images. It is also important that the program fails rather than generating a faulty result. It is preferred to have the generation fully automatic for the same reason as why the execution time is important.

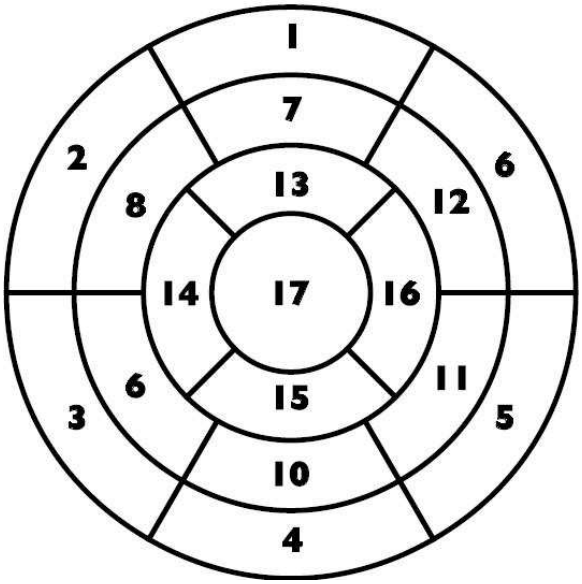


Figure 2.1: An empty bullseye

Chapter 3

Methods for coronary artery measuring

Which algorithms and techniques can be, and are used in existing programs for coronary artery measuring? How good are these techniques? In this in-depth study, some methods for measuring vessel properties are examined. This is a vital part in this thesis because it will help in the implementation choices made later on in the implementation stage of the project.

The sections in this chapter are divided into key steps of the measuring process and are in execution order.

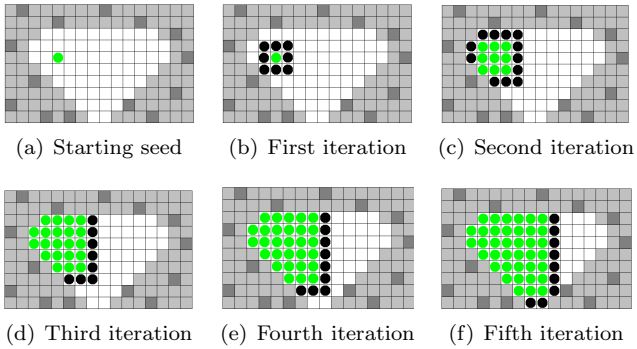
3.1 Segmentation

In medical image analysis there is much to gain from using segmentation to further visualize the information in medical images. Segmentation is one of the cornerstones in image processing.

3.1.1 Region growing

In [5, 6, 25] the segmentation is done with a method called region growing. This is a method that with the help of seeds (starting points) grow the region surrounding the seeds until the whole object is filled. This is done using a threshold and all neighbors that pass the threshold become new seeds. The same procedure is done until no more new seeds can be created. When two different seed regions meet they merge into one region which makes the use of more than one starting seed possible without effecting the result [15].

Region growing may seem straight forward but can be improved and altered in many different ways. You could for example have more than one threshold or use gradient information as a way of detecting new seeds. The placement of the seeds also vary between used techniques. Usually it is up to the user to plant the initial seeds in different locations. If that is not the case some part in the image is recognized by the method which enables automatic seed placement. A model based semi-automatic seed placement method is one way to place the seeds. In [25] the user marks a starting seed on the aorta. The aorta is assumed to have an elliptic cross-section in the image and this



(a) Starting seed (b) First iteration (c) Second iteration

(d) Third iteration (e) Fourth iteration (f) Fifth iteration

Figure 3.1: Region growing. Green pixels are old seeds and black pixels are new seeds

assumption is used in order to find the coronary arteries which has their roots near the bottom of the aorta [14]. These findings are then used together with region growing in the segmentation step. The problem with this technique is that the assumptions made are not always true. If the patient in the image has aortic aneurysm (dilated aorta) [11] the aorta is deformed and may not have an elliptic cross-section. Most patients get images acquired with Magnetic resonance imaging (MRI) [28] or CT scan because there is something believed to be wrong, so it is not that far fetched to think that the aorta can be in some other form than the original elliptic. Another model based method is to assume vessels to have tubular structures as in [12]. This should work fine as long as it is combined with another segmentation method. It could for example be used for a growing criteria in a region growing method which would help prevent leakage.

A problem with the region growing method is that it is sensitive to noise and artifacts. This can somewhat be handled by adding a few parameters to the seed qualification step. As proposed in [7] there is a global lower threshold to exclude darker regions. There is also an upper threshold in combination with a size limit to exclude large bright regions. If the bright regions are below the size limit the upper threshold has no effect. This is because small calcifications and irregularities in the contrast agent should have no effect. A final criteria in this approach is a gradient threshold, which means that the neighboring voxels must be close in gray level to be considered seed points. To make sure that the segmentation is as good as possible, and that there are no leaks into other objects, many segmentations with different global threshold are done. To save execution time the old segmentation is used as seed points for the upcoming segmentation. With the segmentation results a graph with threshold plotted against segmented volume is generated. By examining the derivative of the resulting graph, precise points where leakage occur can be found. This method requires human interaction in the way that a human selects the appropriate threshold based on the resulting graph and/or segmentations, but can generate an initial guess that a user can discard if the result is bad or use if it is good.

If human interaction is to be avoided, how can region growing be achieved? The seed points and the thresholds need to be determined automatically. In a coronary CT image the use of contrast agent helps to highlight different parts depending on the specific image's purpose. If arteries are to be found they should be highlighted in a good image. The problem is that other parts are highlighted as well because the contrast agent does not travel through the system as one connected blob. To isolate the vessels of interest one needs to both look at the contrast and at some object properties in order

to distinguish them from other contrast enhanced objects. As seen in figure 3.2 not only the desired coronary arteries are highlighted with the contrast agent.

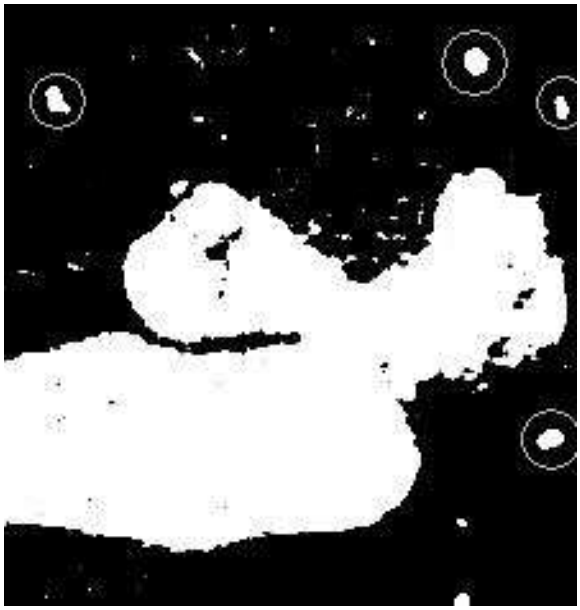


Figure 3.2: Example of thresholded image showing all contrast filled objects. Coronary arteries are circled.

It can be bad to assume that the shape is consistent in all patients because of various diseases that can change this property. If shape is not an object quality to look for, then perhaps size is better.

3.1.2 Thresholding

A different segmentation method is based on thresholding. The image is thresholded based on the intensity. That should remove all objects that are not contrast filled for the coronary artery segmentation problem. All remaining objects are separated, and a size threshold is done. All objects larger than the desired threshold are removed, as well as the objects that are to small. Left is the desired object [19]. The downside with this technique is that it needs the user to mark the desired vessel.

Vessels are the thinnest objects visible around the heart in a coronary CT. This can be used, together with the contrast agent property, in order to eliminate all other objects. Even though there are diseases that affect the size of the arteries, they would still be smaller than all other objects that are filled with contrast agent. There is one exception: Lung vessels can be the same size as the coronary arteries and are filled with contrast agent. However, these are located outside the heart structure and are surrounded by air [4]. This makes them easy to distinguish from coronary arteries.

3.1.3 Voxel mapping

Another coronary artery segmentation method is described in [33]. This method is based on the placement of seed points in different objects. Each voxel in the image is then

mapped to one of these seed points, meaning that they are in the same object as that seed point. The mapping is done by calculating the best path from a tested voxel to each seed point, which is determined by the intensity of the voxels in the path. The seed point that the best of all paths lead to then represents the object which the tested voxel belongs to. This method is neither automatic or fast [34], which makes it to time consuming for medical staff to use frequently.

3.1.4 CT segmentation problems

A problem that exists when segmenting objects from a CT image, is that the slices are created from a spiral pattern as discussed in section 1.1. This means that the information of the object in a slice is approximated based on the information contained in the spiral that is closest to the desired slice. This makes the border of an object difficult to distinguish.

3.2 Centerline extraction

In [5] *centerline extraction* is described as a key step in order to get a morphometric quantity of vessel systems. There are many ways to get a centerline and methods for this can be grouped mainly into two categories.

3.2.1 Direct methods

The first category is those methods which directly computes the centerlines with optimal paths that connect a set of points which are usually set by the user. The optimality is determined by internal factors (shapes) and external parameters (color).

3.2.2 Skeleton based methods

The other category is skeletonization which is reduction of a shape into a graph. A skeleton is created by finding centerline points. To be one of those points, the shortest distance from it to the object's wall must be the same as to another point on the wall [16].

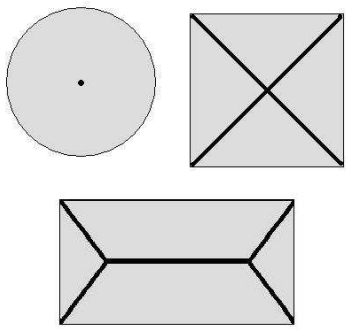


Figure 3.3: Geometric shapes with skeletons marked black

Both methods has their advantages. The direct method does not require a segmentation of the object while skeletonization does in order to measure distance to the object wall. A skeleton can on the other hand create a much more accurate centerline depending on the precision of the segmentation. The direct method can be imprecise in bifurcations and strongly bent segments because its optimality measure is dependent of length and curvature.

Considering that the accuracy along with robustness, verifiability and reliability is important in medical image analysis, which of these techniques is the better one? The skeleton produces a more accurate result, assuming a good segmentation and can easily use manual input for e.g. end points (points where skeleton stops) that increases the robustness of the method. The direct method can not take consideration to user input that easily, which results in a less robust solution where users have to accept that it is not possible to get a good skeleton in some cases. A skeleton however is sensitive to the surface structure of the segmented object. Branches that are not supposed to exist can be formed by one voxels oversegmentation. This means that there is pre-processing to be done before a successful skeletonization can occur, or post-processing removing unwanted branches. The skeleton method also provides a reliable result as it can be done in a verifiable way. If it is done as in [8] it can be shown that the theoretical accuracy is 1 voxel distance. The direct method can be executed faster but has less accuracy, and since accuracy is important in measuring small arteries, the skeletonization technique is suited fitted for these types of problems.

3.3 Artery measuring

In order to get a correct measurement it needs to take place on a cross section instead of an image slice, and the cross-section needs to be perpendicular to the vessel direction. A straight forward way to get a cross-section is to find a plane orthogonal to the skeleton that is used to cut the vessel and create a cross-section at that location. By using this method an incorrect answer may be found because of the geometrical nature of the artery. If the plane is orthogonal to the skeleton it does not need to be orthogonal to the artery because the skeleton may not have a smooth appearance. In [35] earlier calculated contours are used to create an overdetermined system which is solved using the singular value decomposition (SVD) [29]. This yields the cosine of the normals direction to the best fitting plane, and enables the plane to be uniquely solved using the acquired normal and centerpoint (skeleton). This can be done since all that is needed to define a plane is the normal and a point on the plane, which is now acquired. A problem with the location of a cross-section is that the voxels in the image are often not square. This can result in an inaccurate measure if the method used does not handle this in some way.

Since the arteries are quite small, precision is crucial. It might be a good idea to re-segment the artery with stricter rules for each cross-section. This can be done as in [8] with the watershed method.

3.3.1 Watershed

The watershed method is based on the idea that a gradient of the image represents a height map and water is rising, making more and more pixels come under water forming ponds in the image. When two or more ponds try to merge, a dam is built to keep them separate from each other. When the algorithm is complete, all there is in the image are dams. These represent borders between objects and can be used to segment the original

image [17]. Since the centerline is already calculated it can act as a marker for the region to save, and the rest of the boarders can be ignored. The watershed algorithm is however sensitive to noise and artifacts, which could lead to oversegmentation. It can be solved by using "markers" as described in [18]

3.3.2 Region growing

Another way to re-segment the arteries is with the region growing method, since every 2D layer has seed points in form of the centerline. By segmenting every cross section separately, less contrast deviation needs to fit inside the tolerance level. This enables the use of a more precise threshold for each slice, which results in a finer segmentation.

3.3.3 Resegmentation

The reason why a second segmentation is good is because it is the free artery path inside the artery wall that needs measuring. In the first segmentation it is likely that the artery wall as well as calcifications inside the artery pass as part of the artery, which will increase the inaccuracy of the measurement.

The watershed algorithm is perhaps better for re-segmentation because more information is gathered. In the vessel there can be calcifications and other non diameter contributing substances. These can give vital information to medical staff about the patients condition and should be kept. The watershed returns boundaries for all types of objects which makes them easy to locate, while the region growing method only finds one type of gray-scale object. Even if the calcifications in themselves are of no interest, the flow capacity of the artery in a healthy state versus the current capacity might be of interest, in order to see how the clogging might affect the patient. Even if the watershed is used, some other method is needed since boundaries are only used to locate the different sections, not returning specific objects. An advantage using the watershed is that in the case of severe calcification, the centerline might be in the calcified part of the artery, which results in an incorrect segmentation with region growing. This could be troublesome to implement reliably without human input, but with the watershed one can use the shape of the borders in order to select which is the vessel lumen and which is the calcified region. This can be determined by the convexity border shape of the calcified region and concavity of the lumen region.

3.3.4 Measuring

Once the cross-sections are found, measurements can take place. The method for this final step is very dependent of the preceding ones so this measurement varies among most techniques. Basically what is done is to calculate the cross-sectional diameter or area. Using the diameter, stenosis is often underestimated and other errors may occur because the vessel lumen may not be perfectly circular. The diameter is often calculated because of comparison with traditional measuring methods. In modern measurements however the area measure is the better choice because it is more precise and has a direct connection to the flow in the artery [9].

Chapter 4

Development

In this chapter the development and function of the created methods is described.

The program can be divided into several steps and the sections in this chapter are representing one step each. The steps are listed in this chapter in the same order as they are executed in the program.

4.1 Parsing

The images that the program should support are stored in the DICOM (Digital Imaging and Communications in Medicine) file format. This is the dominating standard for image storage in the radiology field. A DICOM file consists of tags specifying what information is stored in the specific line. There is a header section and a image section. The header section stores information of the patient, pixel representation and the camera used to create the image. To interpret the tags, the PixelMed Java DICOM Toolkit was used. This toolkit interprets the DICOM file and makes the information easily accessible.

The parsing is done by reading a series of files where each file represents a CT slice. To determine the order of the slices the header-tags "Image Orientation Patient" (IOP) and "Image Position Patient" (IPP) are used. IOP specifies the direction cosines of the first column and row for the slice with respect to the patient, and IPP specifies the coordinates of the upper hand corner of the image relative to some origin specified by the scanner settings. With these values the distance from the origin can be calculated as shown in equations 4.1 and thus the slices can be sorted in correct order [2, 1].

$$\begin{aligned} normal &= IOP_{0-2} \times IOP_{3:5} \\ distance &= normal * IPP^T \end{aligned} \tag{4.1}$$

Since the goal is to measure the coronary arteries, the real world pixel dimensions needs to be found. This is done with the tag "Reconstruction Diameter" which tells how many millimeters a pixel represents.

Other header-tags needed are "Rescale Slope" and "Rescale Intercept" and they are used to get the correct intensity values in the image. This is calculated as in equation 4.2 [13].

$$intensity = readPixelValue * RescaleSlope + RescaleIntercept\,[HU] \tag{4.2}$$

The acquired intensity is in Hounsfield Units which is a radiointensity scale, enabling more reliable thresholding. See table 4.1 for different HU values in a few substances [4, 20].

Substance	HU
Air	-1000
Fat	-120
Water	0
Muscle	40
Contrast	>130
Bone	400-2000*

*depending on the density of the bone

Table 4.1: Substances with their corresponding Hounsfield Units

4.2 Aorta identification

To create an automatic method, the program needs to orientate itself. The method implemented to do this is to find the aorta since it can be distinguished in the image sets, and it is also connected to the coronary arteries and left ventricle, which will be needed later. After studying a few CT images it was noticed that all images start around

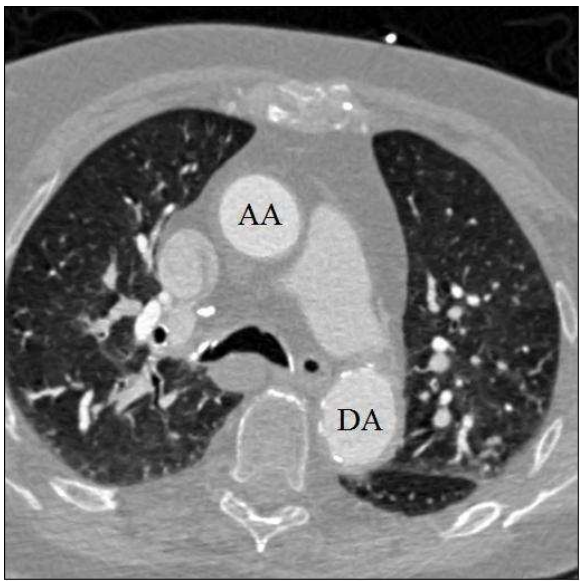


Figure 4.1: Top slice of a CT image. AA = Ascending Aorta, DA = Descending Aorta

the same place in the body. That is, the top slice in one image has a high probability of showing exactly the same organs as another images top slice. This can be used in order to find the aorta automatically. The descending aorta and ascending aorta are divided in the top slice and it is the ascending aorta that is the starting point for the coronary arteries. In order to find this part of the aorta, two assumptions are made. The first is

that the aorta is filled with contrast fluid, and the other assumption is that the aorta is larger than a circular shape with the radius of 10 mm.

The aorta is localized by a raster based search in the topmost slice. A circle with a specified radius is moved over the image, row after row, starting in the top left corner. When it reaches the end of a row it is moved to the left of the image again starting on the next row. At each step the pixels contained in the image are thresholded and a value of how much the circle is filled is calculated. If the circle has a fill rate of under 25% it is moved one radius to the right. This is done until the fill rate is over 90% which indicates that the aorta has been found. The threshold is set to 90% in order to allow noise and a slight irregularity in the contrast agent.

Because the circle moves from the top down, it is impossible to find the wrong part of the aorta first because it is located in the lower part of the slice just above the spine. The fill threshold enables the circle to pass bone, lung veins and other material that is as bright as contrast agent without the risk of classifying these as the aorta, as none of these objects has a chance to fill the whole circle.

4.3 Aorta segmentation

This section describes the different techniques for the aorta segmentation. They are described in the order that they were implemented.

4.3.1 Circle approximation

This method is based on the assumption that the aorta has a circular shape and that the only thing extending from the aorta is the coronary arteries.

The aorta in $slice_i$ is segmented with a region growing based approach. The segmented object is however never saved because it is not needed. Instead the center of the segmented pixels (p) are calculated as in equation 4.3.

$$center = \frac{1}{n} \sum_{i=0}^{n} p_i \tag{4.3}$$

The number of segmented pixels (n) are used in order to find the radius of a circle covering the area that the segmented pixels cover as in equation 4.4 [32].

$$\begin{aligned} Area_{circle} &= \pi * r_{circle}^2 \\ r_{circle} &= \sqrt{\frac{Area_{circle}}{\pi}} \\ r_{aortaapprox} &= r_{circle} * 1.4 \end{aligned} \tag{4.4}$$

The radius of the circle covering the area is then multiplied by 1.4 in order to allow slight irregularities in the aorta shape and to reduce the effect from noise in the image data.

After this is calculated for $slice_i$, a circle is drawn on it with the calculated parameters and an intensity of -1000 HU. This is done to prevent any connections between the aorta and the coronary arteries. The calculated center of the circle is then used as a seed point for the region growing done on $slice_{i+1}$.

Every segmented pixel, which is not covered by the approximated aorta circle, is considered to be part of an artery and is stored to be used as a starting point for future arterial segmentation.

The aorta segmentation is done for every slice until the aorta root is encountered. The root is recognized as the aorta split into three by the dark borders of the aortic valves as shown in figure 4.2

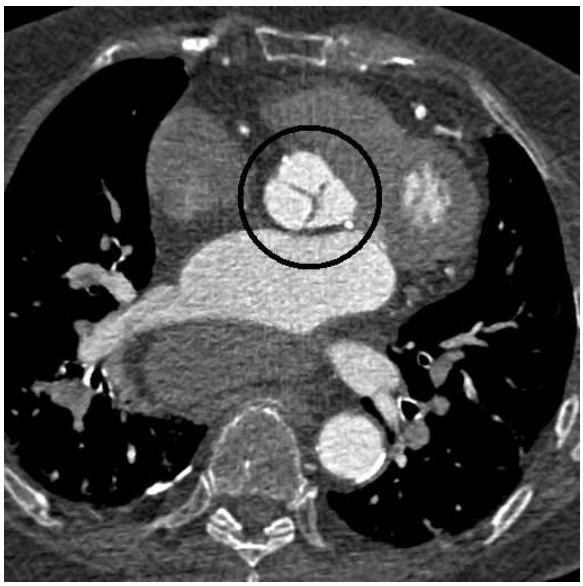


Figure 4.2: Image with the aortic root marked with a black circle

The major flaw with this method is that it cuts off the part of the artery that is closest to the aorta. This was at first considered to be tolerable, but it was noticed that in an image, the right coronary artery was changing to a downwards direction while still inside the approximated circle. This has the effect that the whole artery tree is missed because no seed points are placed for the artery segmentation algorithm.

4.3.2 Region growing

With the aorta circle approximation discarded an attempt to use the information from the region growing is made. The aorta in each slice is segmented with region growing in the same way as in the circle approximation method, but uses the actual shape of the aorta instead of an approximated circle, and checks for arteries outside a dilated version of the segmented aorta. This removes the problem of missing arteries in the segmentation. Unfortunately it was discovered when more test images became available, that the quality of the images vary and in most images the aorta merges with the superior vena cava and/or pulmonary artery. This enables leakage in the region growing, which makes this method insufficient. Another problem discovered in the newly acquired images was that the aortic root is inseparable from the rest of the aorta making the stop criteria for this algorithm useless.

4.3.3 Lines from Center

Since the region growing method was a little slow and did not work properly, a new approach was implemented. This new approach is based on path creation. The aorta

itself is not interesting except for the fact that it leads to the arteries. Thus the shape of the aorta can be ignored and only a path following the aorta needs to be acquired. This is done by creating lines from the center of the aorta. From each point on these lines a ray is shot in a direction that is perpendicular to the line and another ray in the opposite direction. These rays stop at the aortic border where the intensity in the image is below the intensity threshold. The mean value of the two points are calculated and stored as a point p_i.

This procedure is done in eight different directions resulting in many p. If the aorta is perfectly round, all p should form eight straight lines. If on the other hand an artery appears in a slice, the lines appear to be attracted to the artery location, thus enabling the discovery of the arteries.

A new center was calculated as in equation 4.5 and used as center for the following slice.

$$center = \frac{1}{n}\sum_{i=0}^{n} marker_i \tag{4.5}$$

This method is extremely fast compared to the ones described above. It is however very sensitive to noise. Most CT image sets contain quite a bit of noise and the contrast fluid is irregular in the aorta. The method was discarded due to the lack of robustness.

4.3.4 Region growing with object extraction

The lack of robustness in the previous methods encouraged the implementation of a more complex, robust and unfortunately also slow method. This method returns to the use of region growing because of it's insensitivity to noise.

The first step of the algorithm is dependent on that the aorta has not merged with any other objects in the image in the starting slice. The aorta is then segmented with region growing. All pixels segmented are stored in a new image to enable use in the following slice. The new image is used to calculate a subimage for the following slice as in equation 4.6 where p_i is a segmented pixel.

$$\begin{aligned} width &= (max(p_{x_i}) - min(p_{x_i})) * 1.4 \\ height &= (max(p_{y_i}) - min(p_{y_i})) * 1.4 \\ image_x &= center_x - (width/2) \\ image_y &= center_y - (height/2) \end{aligned} \tag{4.6}$$

The creation of the subimage is done to improve speed in the following steps.

Once $slice_i$ and $slice_{i+1}$ have been acquired, $slice_i$ is dilated. The slices are then compared and $slice_i$ should cover the entire $slice_{i+1}$. If that is not the case, the pixels in $slice_{i+1}$ that are not covered are counted. If the count is larger than what is considered to be noise in the image, an object extraction is done.

The object extraction is done by checking which of the pixels that are remaining in $slice_{i+1}$, after removing all pixels that were in both $slice_i$ and $slice_{i+1}$. The object size is calculated and compared to two different thresholds. The first is an upper threshold. This is a value that is considered to large to be an artery, and therefore must be some unwanted object, and is removed from the subimage. If on the other hand the object is smaller than the upper threshold, it is compared to a lower threshold. If the size is larger than this the object is considered to be an artery.

Since the aortic root is not visible in some image sets, another stop criteria has to be implemented. The final criteria is that when objects larger than the upper threshold have

been found on enough consecutive slices, the slice's location in the body is considered to be just below the aortic root, where the aorta merges with the left ventricle and the pulmonary artery.

4.3.5 Final version

The method used in the final version of the implementation is quite similar to the one described in section 4.3.4, except that the goal of the method has been changed. Instead of locating the artery starting points, the method is used only to locate the merging point between the aorta and left ventricle. This is done after establishing that the CT images has to low quality to segment the arteries. They tend to merge with many different organs which makes it to hard to determine what is part of the artery. This merging is occurring in the regions close to the aortic root so it is possible to bypass these merges by using an approach as the one described in section 4.5.

4.4 Left ventricle optimization

The left ventricle has to be found in order to enable the projection of the arteries onto a bullseye plot. Since the bullseye uses the approximation that the left ventricle is an ellipsoid that has been cut in approximately half, it came natural to try and find the ellipsoid that best follows the left ventricle wall. To find this ellipsoid an optimization method that requires points to optimize is used. The points are placed on the left ventricle wall according to the methods described below.

4.4.1 Point placement

The points that are to be optimized are placed with the help of the point where the aortic segmentation stopped. Since the stop criteria was that the aorta merges with the left ventricle, the centerpoint from that last slice is on the same slice as the left ventricle.

In case of a patient with no dilated heart chambers, the left ventricle is located directly right of the aortic root in the image. The left ventricle is thus located by following a straight line from the aortic root until an intensity lower than 0 HU is detected. This point is on the right outer edge of the left ventricle. From the edge point a line in the downwards left direction is followed until contrast fluid is encountered. This point p_{inside} is located inside the left ventricle. In order to find an approximate centerpoint of the left ventricle, a line going straight to the left is followed until a point $p_{otherside}$ on the other side of the contrast agent is found.

$$p_{otherside_x} + \frac{p_{inside_x} - p_{otherside_x}}{2} \tag{4.7}$$

Equation 4.7 gives a point p_{center} located in the middle of the contrast fluid. By following a line straight up and straight down $line_{updown}$ from p_{center} until an edge point on the left ventricle is found, a centerpoint p_z and a distance $dist_z$ in this direction can be found. Points can now be placed by following $line_{updown}$.

While following $line_{updown}$ upwards in the image stack points are placed as in figure 4.3.

When placing points below p_z in the stack it gets a more complicated. The left ventricle is filled with contrast agent, but not entirely. There are papillary muscles in the ventricle that appear as ellipsoids coming out of the right side of the ventricle

Figure 4.3: Point placement while above the approximated centerpoint. Points are marked with gray squares

wall. This is important to notice because in order to place optimization points on the septum, the thickness of the left ventricle wall needs to be approximated. This is done by measuring the wall while placing points on the right side. In order to get the correct distance, the papillary muscles are removed by calculating the minimum convex hull of the contrast fluid bolus. By doing this and setting pixels inside the hull to an intensity that is the same as contrast fluid, a correct wall measurement can be done.

There is one more set of points that needs to be placed. This are the points that prevent the optimized ellipsoid to become infinitely large. The points placed are on the right side on the border between contrast fluid and the ventricle wall.

4.4.2 Optimization

The optimization method used is the Levenberg-Marquardt method which optimizes a transformation matrix that transform all points onto an ellipsoid. For this optimization the Apache commons math library was used.

Levenberg-Marquardt method

The reason that the quicker Gauss-Newton method is not used is because a good starting guess can not be provided for the optimization. Knowing this it is worth using the more time consuming and robust Levenberg-Marquardt method.

The definition of this method is:
The vector p is a solution of

$$\begin{aligned} &\min_{p} ||Jp + r||^2 \\ &subject\ to\ ||p|| \leq \Delta \end{aligned} \tag{4.8}$$

if and only if there is a scalar $\lambda \geq 0$ such that

$$\begin{aligned} &(J^T J + \lambda I)p = -J^T r \\ &\lambda(\Delta - ||p||) = 0 \end{aligned} \tag{4.9}$$

where p is a vector containing the current optimized values, J is the Jacobian matrix and $\Delta > 0$ is a trust-region radius [26].

Optimization setup

The general function for an ellipsoid is shown in equation 4.10 [21].

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \tag{4.10}$$

From this it is determined that the length of the axes (a,b and c) needs to be found. There are more values to be found if the criteria of equation 4.10 is considered. This formula only works if the ellipsoids axes are aligned with the coordinate systems axes. The center of the ellipsoid is also required to be in the origin. Since the left ventricle in most cases does not follow the images coordinate axes and is not located in the system's origin, one needs to expand equation 4.10 into equation 4.11.

$$\frac{x'^2}{a^2} + \frac{y'^2}{b^2} + \frac{z'^2}{c^2} = 1 \tag{4.11}$$

This means that the image coordinates need to be transformed into a coordinate system that fulfills the requirements for equation 4.10 to be applicable.

All points found on the border of the left ventricle are assumed to lie on the outline of the resulting ellipsoid, which means that equation 4.11 holds. The points that were taken from inside the ellipsoid are given a value corresponding to their distance from the center and from the border.

Transformation method

At first an affine transformation matrix 4.12 and a translation vector 4.13 is to be optimized [22].

$$A = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \tag{4.12}$$

$$T = \begin{pmatrix} cx \\ cy \\ cz \end{pmatrix} \tag{4.13}$$

This is all that is needed, since the axes for the ellipsoid is contained in the scaling factors of 4.12, leaving a relatively easy formula to partially differentiate. This however does not work due to the fact that the rotation matrix allows for skewing, which resulted in non perpendicular coordinate axes for the resulting system. This is not acceptable since it violates the definition of the mapping between the ellipsoid and the bullseye. Another reason why this method was discarded is that the skewing affected the resulting scaling factors, making the ellipsoids axes hard to attain.

A new approach is to optimize three rotation angles, ellipsoid axes and translation vector. The rotation angles are used to construct a pure rotation matrix as in 4.14 [23], and the translation matrix is kept separate since the values are the center for the ellipsoid

in image coordinates.

$$
\begin{aligned}
R_x &= \begin{pmatrix} 1 & 0 & c \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix} \\
R_y &= \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix} \\
R_z &= \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
R &= R_z * R_y * R_x
\end{aligned}
\tag{4.14}
$$

The coordinates can then be transformed as in 4.15

$$x' = (x - T) * R \tag{4.15}$$

where x is the point in image coordinates and x' is the resulting point.

4.4.3 Error detection

Since the program needs to be reliable an error detection function exists. It assigns a score to the optimization result, and by looking at that score medical staff can determine the reliability of the resulting bullseye plot.

The score is determined by looking at the lengths of the axes in the resulting ellipsoid. These are compared to the size of a healthy person's left ventricle. If the axes are close to the size of a healthy left ventricle, the score is low and increases as the comparison gets worse. The root mean square of the optimization residual is also taken into consideration, as the score also needs to represent the adjustment of the ellipsoid to the points placed on the left ventricle wall.

If the score is high enough, a manual localization of the left ventricle is requested as the optimization results are determined to contain too much error for an accurate projection to occur.

4.4.4 Manual localization

If the point placement or the optimization should fail, there is another possibility to find the left ventricle. This is based on the manual placement of points as in figure 4.4.4.

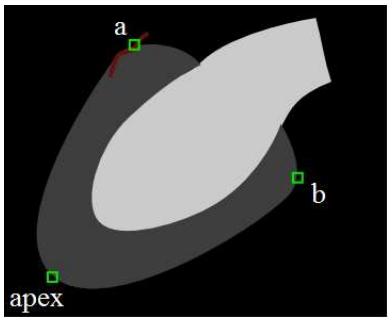


Figure 4.4: Figure showing how the manual points should be placed on the left ventricle

These points are all that is needed for an ellipsoid approximation of the left ventricle. First the center of the ellipsoid is calculated as in equation 4.16.

$$center = \frac{a+b}{2} \tag{4.16}$$

Then the C axis is determined as the vector going between the center calculated in equation 4.16 and the apex. The A axis has to be orthogonal to the C axis, and it does not matter in which direction it faces since an approximation that the A and B axis have the same length have been made. Then equation 4.17 can be used for determining the A axis [30].

$$\begin{aligned} dir &= (a - center) \times (apex - center) \\ dir_{norm} &= \tfrac{dir}{||dir||} \\ A_{stop} &= dir_norm * distance(a, center) \end{aligned} \tag{4.17}$$

The A axis is determined to be the vector between center and A_{stop}. In a similar way the B axis is determined. This is calculated to be the vector between center and B_{stop} determined in equation 4.18 [30].

$$\begin{aligned} dir &= (A_{stop} - center) \times (apex - center) \\ dir_{norm} &= \tfrac{dir}{||dir||} \\ B_{stop} &= dir_{norm} * distance(a, center); \end{aligned} \tag{4.18}$$

The lengths of the axis are for C the distance between the center and apex. For A and B the length is the distance between the center and point a. Once the axes have been determined the transformation matrix needs to be determined. This time a rotation around an arbitrary vector is calculated as in equation 4.19. The vector to rotate around as well as the angle are determined by looking at the difference between the current z axis and the wanted z axis [24].

$$\begin{aligned} n &= apex - center \\ l_{long} &= n \times (0,0,1) \\ l &= \tfrac{l_{long}}{||l_{long}||} \\ \alpha &= \arccos(n \cdot (0,0,1)) \\ R &= \begin{pmatrix} l_x^2 + (1-l_x^2)c & l_x l_y (1-c) - l_z s & l_x l_z (1-c) + l_y s \\ l_x l_y (1-c) + l_z s & l_y^2 + (1-l_y^2)c & l_y l_z (1-c) - l_x s \\ l_x l_z (1-c) - l_y s & l_y l_z (1-c) + l_x s & l_z^2 + (1-l_z^2)c \end{pmatrix} \end{aligned} \tag{4.19}$$

where $c = \cos(\alpha)$ and $s = \sin(\alpha)$.

The translation is already determined to be the same as the centerpoint calculated in equation 4.16. Now all the necessary variables are determined.

4.5 Artery segmentation

The coronary arteries that supply the left ventricle lie in a layer of fat which surrounds the heart, and the only other contrast filled object that lies there are the veins. This should not be a problem however, because of the difference in intensity.

The artery is segmented by scanning the area between two ellipsoids. These two ellipsoids are two different scalings of the ellipsoid representing the left ventricle. One ellipsoid is larger and one is smaller. This is needed because there might be an error in

the preceding optimization step and to cover the possible area where an artery can be, one needs to look both inside and outside of the approximated left ventricle wall.

There are many approaches when trying to determine if a voxel is part of an artery or not.

4.5.1 Thresholding

The first method tried was thresholding. This finds all artery voxels but several other voxels as well. The outer ellipsoid can stretch into the lung, making many lung veins appear in the segmentation area. These have the same intensity as the coronary arteries so they will also be segmented.

4.5.2 Surrounding voxels

By adding the criteria that an artery voxel has to be surrounded by fat [4], all lung veins should be removed since these are surrounded by air. The only thing that is left should be the artery voxels. This is however not the case. Due to noise and low resolution, the veins appear to be surrounded by a thin layer of fat. This is because fat has an intensity between contrast and air, and when creating a CT slice the boundary voxels get an intensity between the two original intensities.

4.5.3 Path walking

The best method, and also the slowest, is a path walking method. By walking from the center of the ellipsoids and out to the voxel, things can be noticed along the path. A voxel is said to belong to an artery if the path contains contrast fluid, muscle and fat. Nothing else is allowed, excluding small areas that are determined to be caused by noise. This means that no lung veins can be segmented because there is air on the path [4].

All these criteria are of course depending on that an initial thresholding has taken place.

4.6 Projection

The projection onto a bullseye plot is made by determining two angles. The first is the angle between the point and the xy plane (α) and the second is the angle between the point and the xz plane (β). This is needed in order to express the coordinates in radial coordinates. The length (r) is calculated as in equation 4.20 [31]

$$\begin{aligned} \alpha &= \arccos\left(\frac{p \cdot normal}{||p||\,||normal||}\right) \\ r &= \frac{2\alpha}{\pi} \end{aligned} \tag{4.20}$$

where *normal* is the z axis which is the normal to the xy plane and p is a point in the ellipsoids coordinates.

The direction in which to draw the point is calculated as in equation 4.21.

$$\begin{aligned} &\beta = \arccos(p_x) \; or \; \arcsin(p_y) \; depending \; on \; which \; quadrant \; p \; is \; in \\ &dir = (\cos(\beta), \sin(\beta)) \end{aligned} \tag{4.21}$$

β needs to be calculated for further use, even though it could be skipped when calculating dir. The point on the septum is used to determine what the direction represents, as it is just grabbed out of the ellipsoids random coordinate system. The septum point should always be between section 1 and 2 in the bullseye. This is an angle of 120 ° which means that β needs to be normalized with respect to the septum point before dir is calculated.

The point is then projected onto the bullseye according to the equation in 4.22.

$$p' = r * dir * radius_{bullseye} \tag{4.22}$$

Chapter 5

Planning and methods

This chapter describes the initial planning, what methods were used and how the planning held up.

The program was written in Java with the goal that it would work on all major operating systems. Three external libraries were used during the project:

Pixelmed DICOM toolkit which was used for reading CT image sets which are stored in the rather complex DICOM file format.

Jama which is a basic linear algebra package, which was used for complex matrix operations.

Apache commons math which was used for it's optimization methods.

5.1 Preliminary planning

The department had a general idea how the program was to be implemented and provided an initial plan and schedule for the project. There was no room for an in-depth study so the time for that would be taken out of the reading part of the schedule, since they are highly related. The implementation planning was in this stage divided in several more or less related steps, which allowed downscaling of the project if time was a problem.

5.2 How the work was done

The in-depth study was used to gather information about techniques in the same field as the program that was made. Some articles were provided by the department but most of the time of the first weeks was used for information gathering. By doing the in-depth study in the area of medical image analysis with focus on arterial segmentation, a good starting point for the implementation stage was created.

The image sets differed a lot and it was determined that the methods learned in the in-depth study would not work well for this program. This, and factors like unplanned human anatomy studies, caused the implementation to take longer than expected. A few parts were skipped in order to make the ones that time allowed to do as robust as possible.

Chapter 6

Results and discussion

In this chapter the results of the development are shown.

6.1 Program

The extraction of the coronary arteries are performed and presented on a bullseye plot. This plot is the result of a semiautomatic procedure which takes about 25 seconds to generate on a 2 GHz Pentium dual core running Windows Vista.

6.1.1 User interface

The graphical user interface (GUI) looks as in Figure 6.1. The field to the right contains the current activity of the program and different controls. The controls are used for adjusting the window/level of the CT stack display and manipulating as well as regeneration of the bullseye plot.

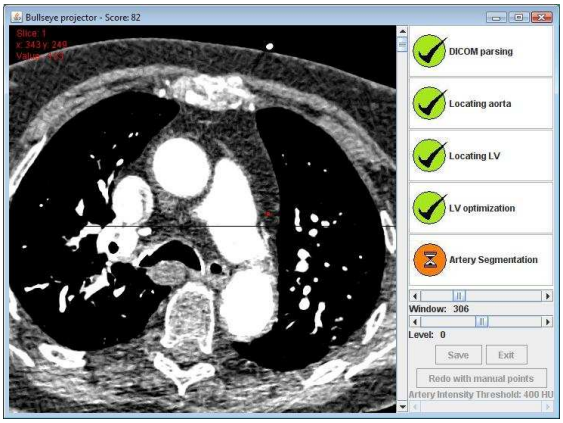


Figure 6.1: Screenshot of the program GUI while segmenting the arteries

6.1.2 Human interaction

The program always requires a manual definition of a point where the septum and outer heart wall meet. This is the point that defines the line between segment 1 and 2 on the bullseye. This is to be done when a panel as in figure 6.2 is shown. The panel disappears automatically when the point is selected.

Figure 6.2: Image showing where to place the point on the septum border

If the automatic left ventricle optimization fails, an instruction for manual point placement is shown. This looks like in figure 6.3 and disappears automatically when all points have been placed.

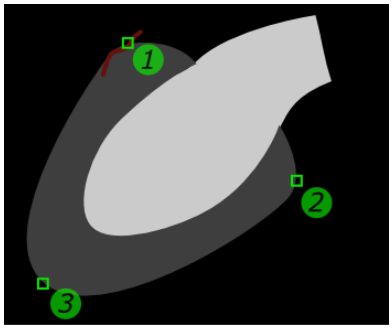


Figure 6.3: Image of the left ventricle showing where to place points

When the result has been calculated it is presented in a bullseye plot. This can be saved into a PNG image if desired, and fine tuning can be done with a scroll bar. This bar changes the thresholding value used for presentation. If the resulting plot fails to meet the requirements of the user, the choice to place manual points for the left ventricle finding can be done and the program does a new segmentation and projection based on this new information.

6.1.3 Resulting plot

A resulting image of the bullseye plot generation can be seen in figure 6.4. The left artery tree can be seen clearly, starting from between section 1 and 2 and going past the center of the plot. The right coronary tree can be spotted in sections 3 and 6. In figure 6.4 there are interfering structures near the center of the bullseye. These are veins and are very hard to separate from the coronary arteries with the existing method.

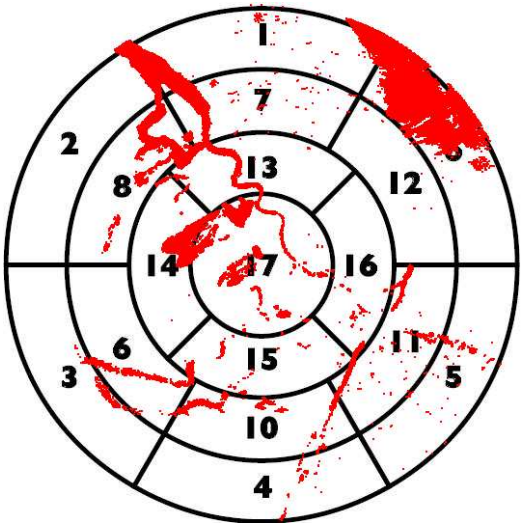


Figure 6.4: A result of running the program. Threshold set to 250 HU

6.2 Algorithms

During the creation of the algorithms 2-8 images were used. The number of images depends on when the implementation was done. When all algorithms were implemented two more images were obtained making the total number of images 10. The program was tested for all images and the result is presented in this section.

6.2.1 Aorta identification

The identification step is successful for all CT images available for testing.

An assumption made is that the ascending aorta has to be located closer to the patients chest than the circular shaped vena cava and pulmonary artery. These, if found first by the method, has the possibility to qualify as the aorta. Another assumption is that the top slice of the image set has to be located on or below the aortic arch. This is because the method looks at the top slice, and if it is located above the aortic arch, the method will return a faulty or no result.

The circle fill rate of 90% is used in order to allow noise and irregular in the contrast agent. Since the only large contrast filled objects allowed to be located above the aorta are ribs, the fill rate of 90% is sufficient because of the ribs shape. The shape of a rib in a slice looks like a non-filled ellipse. This has a fill rate much lower than 90% and will thus not be interpreted as the aorta. When a fill rate of 95% was used some noisy

images generated a faulty result for the aorta identification. The highest fill rate that resulted in the correct identification was, for all test image sets, was 93%. To be on the safe side a final fill rate of 90% was set in case of increased irregularity in future images. A lower fill rate is not desired because of the increased chance of finding the vena cava or pulmonary artery. This is the case because if the fill rate is low, the circle in which the fill rate is calculated can be located more outside a contrast filled object with lower required fill rate. This means that a faulty object can be located farther from the chest and still be identified as the aorta, the lower the fill rate is.

6.2.2 Aorta segmentation

The final version of this method works for all tested images.

Since the image consists of a large amount of voxels, finding a region of interest (ROI) is important. By approximating a radius of the aorta on each slice, the aorta in the following slice can be approximated to be about the same. Using this, the ROI for the aorta in the following slice is assumed to be within a circle with a radius slightly larger than the radius just calculated. This is fine if the aorta is the only thing desired to find. This is however not the case for this method, since surrounding objects needs to be located. This means that the ROI is insufficient. By testing, the ROI is determined to be the aorta radius times 1.4. If a smaller ROI is used the method can miss small or sparse objects which surround the aorta. A larger ROI will make the program run slower since there is a larger amount of voxels that have to be processed.

To determine when a merge between the aorta and left ventricle has taken place, large objects have to be found on a number of consecutive slices. This was determined by a study of the test images. If the value for this detection is to low, there is a chance of finding a merging point that is due to poor image quality rather than being an actual merge between the aorta and left ventricle. It is therefore safer to use a larger value than the absolute minimum since the only downside with a to large value, within reasonable boundaries, is unnecessary computation.

6.2.3 Left ventricle optimization

In 9 of the 10 test images used the automatic point placement detects the left ventricle boundary and places points for the ellipsoid approximation. In the image that does not work all points are placed in the right ventricle. This is because the patient in that image has a very dilated right ventricle, which makes the left ventricle appear closer to the lower right corner in each slice. The right ventricle is at the place where the left ventricle should be and the algorithm thinks that this is the left ventricle.

The method has difficulty placing points near the apex of the left ventricle, because it is hard to distinguish the left from the right ventricle in this area. This sometimes result in an approximated ellipsoid stretching past the apex and outside the final slice of the image set. This is problematic in the artery segmentation step.

6.2.4 Artery segmentation

The images used for testing are to noisy for this method to work well. There are both missed voxels and wrongly segmented voxels in the resulting arteries. This has the effect that the bullseye generated can have a very noisy appearance, as seen in figure 6.4 on the preceding page.

6.2.5 Projection

The projection is straight forward and is completely dependent on the other steps. If all preceding steps return a correct result, the projection will be correct.

Chapter 7

Conclusion

In this chapter the results of this thesis will be discussed.

The initial goal of this project was to make a program that, automatically and fast, finds and parametrizes coronary arteries from a CT image set taken of a human heart. The parametrized arteries have to be projected onto a bullseye and then returned to another preexisting application.

After acquiring enough image sets it was discovered that none of the existing methods studied in chapter 3 would work. The images were not as uniform as first thought and most of the time for this project was spent on getting each step as robust as possible before continuing, rather than getting all steps completed.

The requirement for automatic methods had to be skipped in order to make it possible to find a solution for as many images as possible. This resulted in the manual left ventricle localization. The point placement at the septum was not due to that factor, even though it has that side effect, but to the fact that time for the project was almost up.

Because there were insufficient time to implement the parametrization, this project could not be linked to the preexisting application. The goal was instead changed to a projection on a voxel by voxel basis which does not need the parametrization. This goal is simply to be able to show what the program could do by visualizing a bullseye plot.

7.1 Limitations

One of the limitations for this program is that the top slice in the image set has to be located between the top of the aortic arch and the top of the left ventricle, or in other words showing the ascending aorta. This should be the case for all images, but the patient can have oddly sized organs or the scanning of the patient can be a little off in some way.

Another limitation is that the aorta can not be merged with another organ in the top slice. This is because it is used as a template for how the aorta should look like in the following slices. If this template contains another object, then this too will be considered as part of the aorta for all upcoming slices.

When segmenting the coronary arteries, the method used has a hard time distinguishing arteries from other objects. If there is too much noise in the image it will be very hard to notice any arteries in the resulting bullseye.

7.2 Future work

The program will be integrated with the preexisting application in the future. In order to do this, the segmentation of the arteries needs to be improved. Then the segmented arteries need to be measured and transformed into key points in order to fit the required data model that the preexisting application needs as input.

Chapter 8

Acknowledgments

I would like to thank the following people who helped in completing this masters thesis:
Arvid Rudling - supervisor at ORKI, Uppsala University Hospital
Fredrik Georgsson - supervisor at Umeå University
Tomas Bjerner - medical advisor at ORKI, Uppsala University Hospital

References

[1] National Electrical Manufacturers Association. Digital imaging and communications in medicine (dicom), part 3. In *Digital Imaging and Communications in Medicine (DICOM)*, pages 330–332. "National Electrical Manufacturers Association, 2008.

[2] National Electrical Manufacturers Association. Digital imaging and communications in medicine (dicom), part 6. In *Digital Imaging and Communications in Medicine (DICOM)*, pages 7–90. "National Electrical Manufacturers Association, 2008.

[3] A. Bengtsson and E. Setterberg. *Medicinsk Grundkurs*, pages 258–259. Liber, Sweden, 1994.

[4] Tomas Bjerner. Personal correspondence, 2008.

[5] T. Boskamp, H. Hahn, M. Hindennach, S. Oeltze, B. Preim, S. Zidowitz, and H.-O. Peitgen. Geometrical and structural analysis of vessel systems in 3d medical image datasets. In *Medical Imaging Systems Technology*, pages 1–60. World Scientific, 2005.

[6] T. Boskamp, D. Rinck, F. Link, B. Kümmerlen, G. Stamm, and P. Mildenberger. New vessel analysis tool for morphometric quantification and visualization of vessels in ct and mr imaging data sets. *RadioGraphics*, 24:287–297, 2005.

[7] T. Boskamp, D. Rinck, F. Link, B. Kümmerlen, G. Stamm, and P. Mildenberger. New vessel analysis tool for morphometric quantification and visualization of vessels in ct and mr imaging data sets. *RadioGraphics*, 24:289, 2005.

[8] T. Boskamp, D. Rinck, F. Link, B. Kümmerlen, G. Stamm, and P. Mildenberger. New vessel analysis tool for morphometric quantification and visualization of vessels in ct and mr imaging data sets. *RadioGraphics*, 24:291, 2005.

[9] T. Boskamp, D. Rinck, F. Link, B. Kümmerlen, G. Stamm, and P. Mildenberger. New vessel analysis tool for morphometric quantification and visualization of vessels in ct and mr imaging data sets. *RadioGraphics*, 24:292, 2005.

[10] M. Budowick, J. G. Bjålie, B. Rolstad, and K. C. Toverud. *Anatomisk ATLAS*, page 172. Liber, Sweden, 1993.

[11] Cleveland Clinic. Aortic aneurysm — health information, 2008. [Online; accessed 12-September-2008].

[12] P. Fallavollita and F. Cheriet. Towards an automatic coronary artery segmentation algorithm. *Proceedings of the 28th Annual International Conference of the IEEE EMBS*, 1:3038, 2006.

[13] D. W. Fanning. Converting ct data to hounsfield units, 2006. [Online; accessed 08-Mars-2009].

[14] H. Feneis and W. Dauber. *Anatomisk bildordbok, femte upplagan*, pages 230–231. Liber, Sweden, 2006.

[15] R. C. Gonzalez and R. E. Woods. *Digital Image Processing, Third edition*, pages 763–766. Pearson Prentice Hall, USA, 2008.

[16] R. C. Gonzalez and R. E. Woods. *Digital Image Processing, Third edition*, pages 812–815. Pearson Prentice Hall, USA, 2008.

[17] R. C. Gonzalez and R. E. Woods. *Digital Image Processing, Third edition*, pages 769–776. Pearson Prentice Hall, USA, 2008.

[18] R. C. Gonzalez and R. E. Woods. *Digital Image Processing, Third edition*, pages 776–778. Pearson Prentice Hall, USA, 2008.

[19] G. Grabner, R. Modritsch, W. Stiegmaier, S. Grasser, and T. Klinger. Aorta cross-section calculation and 3d-visualization from ct or mrt data using vrml. 2005.

[20] GE Healthcare. Medcyclopedia — hounsfield unit, 2009. [Online; accessed 08-Mars-2009].

[21] D. Hearn and M. P. Baker. *Computer Graphics with OpenGL*, pages 408–409. Pearson Prentice Hall, USA, 2004.

[22] D. Hearn and M. P. Baker. *Computer Graphics with OpenGL*, pages 271–272. Pearson Prentice Hall, USA, 2004.

[23] D. Hearn and M. P. Baker. *Computer Graphics with OpenGL*, pages 264–266. Pearson Prentice Hall, USA, 2004.

[24] D. Hearn and M. P. Baker. *Computer Graphics with OpenGL*, pages 266–273. Pearson Prentice Hall, USA, 2004.

[25] A. Hennemuth, T. Boskamp, D. Fritz, C. Kühnel, S. Bock, D. Rinck, M. Scheuering, and H.-O. Peitgen. One-click coronary tree segmentation in ct angiographic images. *International Congess Series*, 1281:317–321, 2005.

[26] J. Nocedal and S. J. Wright. *Numerical Optimization, second edition*, pages 258–259. Springer, USA, 2006.

[27] RadiologyInfo. Cat scan (ct) – body, 2008. [Online; accessed 11-February-2009].

[28] RadiologyInfo. Magnetic resonance imaging (mri) – body, 2008. [Online; accessed 11-February-2009].

[29] G. Strang. *Introduction to Linear Algebra, Third edition*, pages 352–359. Wellesley-Cambridge Press, USA, 2003.

[30] G. Strang. *Introduction to Linear algebra, third edition*, pages 265–267. Wellesley-Cambridge Press, USA, 2003.

[31] G. Strang. *Introduction to Linear algebra, third edition*, page 15. Wellesley-Cambridge Press, USA, 2003.

[32] Su, Francis E., and et al. Area of a circle or regular polygon, 2009. [Online; accessed 08-Mars-2009].

[33] C. Wang and O. Smedby. Coronary artery segmentation and skeletonization based on competing fuzzy connectedness tree. *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, 10:311–318, 2007.

[34] C. Wang and O. Smedby. Coronary artery segmentation and skeletonization based on competing fuzzy connectedness tree. *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, 10:316, 2007.

[35] Y. Yang, L. Zhu, S. Haker, A. R. Tannenbaum, and D. P. Giddens. Harmonic skeleton guided evaluation of stenoses in human coronary arteries. In *Medical Image Computing and Computer-Assisted Intervention . MICCAI 2005*, pages 493–494. Springer Berlin / Heidelberg, 2005.