Automatic segmentation of multi-stain histology images of arteries

Laura Leal Taixé

Department of Electrical Engineering

Northeastern University

Masters Thesis

July 2008

Abstract

The main topic of the project is the automatic localization and segmentation of porcine arteries in histology images. This is a project between Northeastern University and The Brigham and Women Hospital in Boston. A set of images is taken from the arteries of a group of diabetic pigs (which contain a lot of plaque). Once the images are taken, the aim is to analyze them fast and automatically. Therefore, the first step is to get the Region of Interest (ROI) of those images. In order to accomplish that, several image processing and computer vision techniques such as splines, snakes, morphological operators, statistical processing, region growing, etc. are applied, all adapted to the specific images which contained a lot of noise and undesired artifacts due to calcification, bad staining, etc. Once the image is well-segmented then the amount of fat can be determined automatically as well as other measurements of interest. This will help overall to better understand the disease of atherosclerosis.

To Gerard, :)

"If music be the food of love, play on"

Acknowledgements

I would especially like to thank my thesis advisor, professor Dana Brooks, who always helped me in every step of the way and agreed to meet even when he did not have the time. A wonderful inspiration, professionally and personally. I would also like to thanks professor Umit A. Coskun, for providing me with all the images and valuable advice during the meetings. Thanks to all the people of the Biomedical Imaging group for all the interesting conversations and group meetings. Many thanks to my parents for their loving support and for passing on their wisdom. Finally, thanks to Gerard for being there all the time.

Contents

Resum de la tesi

La tesi de màster titulada "Segmentació automàtica d'imatges histològiques amb diversos tints" és un projecte en col·laboració entre la Northeastern University i el Brigham and Women Hospital de Boston, USA.

La malaltia relacionada amb aquest projecte s'anomena Cardiopatia Coronària, i està causada per l'acumulació de placa (bàsicament greix) a les parets de les artèries. Per tal de conèixer amb més profunditat aquesta malaltia i la seva evolució, es va començar un projecte per analitzar la quantitat de placa acumulada en les artèries d'un conjunt de porcs. L'alimentació d'aquest porcs diabètics ha estat controlada per tal de fomentar l'acumulació de greix a les artèries. Més endavant, els porcs es maten i es prenen imatges microscòpiques de les seves artèries. Totes aquestes imatges s'han d'analitzar, però actualment l'únic mètode d'anàlisi és manual i realitzat amb Photoshop. Això fa que no es puguin analitzar totes les imatges de cop i que l'obtenció de resultats sigui massa lenta per a realment estudiar l'evolució de la malaltia. És per això, que els metges estan interessats en trobar un mètode d'anàlisi automàtic d'aquest tipus d'imatges.

L'objectiu de la tesi és trobar la Regió d'Interès (ROI) d'aquestes imatges, que inclou la part de l'artèria on s'acumula la placa. Un cop trobada, és més fàcil calcular el tant per cent de greix i, per tant, saber com s'ha vist afectada aquesta artèria en particular. La nostra regió d'interès conté la mitjana de l'artèria que és precisament on s'acumula la placa. Així doncs, la nostra ROI vindrà definida per una frontera interior (la capa elàstica interna) i una frontera exterior (la capa elàstica externa).

En aquest projecte la frontera interior es determina mitjançant un algorisme de creixement de regió. L'usuari haurà d'introduir la posició de la llavor.

La frontera exterior ve determinada per un algorisme que aprofita la informació de color de la imatge. Les imatges tenen diversos colors (lila, rosa, marró...) a causa del

procés de tint, és per això que realitzem una equalització de l'histograma per tenir més uniformitat de color en les imatges. Amb el tint Verhoeff la capa elàstica externa es tenyeix d'un color fosc. El que fem és modelar aquest color seleccionant una sèrie de punts de la frontera de manera manual. Un cop tenim el model o histograma, creem una imatge de probabilitat en la qual analitzarem cada píxel i li assignem la probabilitat a posteriori segons el model de color creat. En la imatge de probabilitat, els píxels corresponents a l'artèria són blancs i el fons negre. Apareixen petits artefactes blancs fora de l'artèria que eliminem amb una obertura per reconstrucció. Un cop aconseguida aquesta imatge, utilitzem *snakes* o contorns actius representats amb *splines* per tal d'aconseguir la nostra frontera exterior. El *snake* s'inicialitza fora de l'artèria i és conduït cap a l'interior. Per tal de garantir la inicialització fora de l'artèria però no massa enfora (per tal que el contorn actiu no s'enganxi als artefactes blancs que encara hi ha en algunes imatges), prenem la frontera interior com a punt de partida i la dilatem fins a assegurar-nos que estem fora de l'artèria. L'energia que condueix l'*snake* té 3 termes: un que assegura que el contorn sigui continu i amb baixa curvatura, un que dirigeix el contorn cap als píxels blancs i l'altre que intenta que els veïns damunt la corba del píxel a analitzar també siguin blancs.

Amb aquest algorisme s'aconsegueix un 55% de resultats positius. Els resultats es mesuren comparant l'àrea de la ROI trobada automàticament i la que considerarem com a *ground truth* (trobada manualment).

Com que l'objectiu del projecte és facilitar a l'usuari la segmentació d'artèries, decidim a més crear una Interfície d'Usuari Gràfica (GUI). La creem segons els problemes de segmentació observats (bàsicament una errònia imatge de probabilitat i una errònia frontera interior ja que es necessita més d'una llavor). Amb aquesta GUI es poden corregir aquests errors fàcilment, i si tornem a calcular els resultats ara considerant que hem corregit aquests petits errors, trobem que ara tenim quasi un 65% de bons resultats.

Si mirem els resultats comparant el número de clics realitzats per l'usuari per a fer la segmentació, veiem que necessita una mitja de 4.76 clics per imatge. Aquesta és una millora significativa respecte la mitjana de 16 clics per imatge que s'havien de realitzar utilitzant els mètodes anteriors al projecte.

Per tal de millorar aquests resultats, es treballa breument en la informació que aporta la textura i es veu que pot ser un bon punt de partida per a la continuació del

projecte.

En conclusió, aquest projecte aconsegueix bons resultats i tot i que no és una solució completament automàtica, significa un pas important cap a la segmentació automàtica d'aquestes imatges histològiques.

List of Figures

LIST OF FIGURES

Chapter 1

Introduction

Coronary Heart Disease (CHD), also called Atherosclerotic Heart Disease or Coronary Artery Disease (CAD) is the leading cause of death in the United States in both men and women. According to the American Heart Association, nearly 2400 Americans die of cardiovascular diseases each day, an average of one death every 37 seconds. In 2004, 52% of these deaths were caused by CHD (1).

The formation and progression of a plaque, which causes the disease, is complicated both to model and to predict. In the laboratories we can obtain high quality images of plaque by doing histology studies of harvested artery sections. Currently these images are analyzed manually or semi-automatically which is a slow and not error-free process.

Ideally we'd like to have an automatic program that would analyze most of our images automatically so we could extract useful information from millions of images with minimal human interaction. The main aim of this project is to develop imaging tools to automatically analyze large number of images to help the characterization of the progression of the disease. In the long run, this would help understand and cure the disease.

In the following sections we present the characteristics of this disease, the motivation for our project, our objectives, an explanation of the methods used and finally our results.

1.1 The heart, the arteries and the CHD

CHD is the end result of the accumulation of atheromatous plaques within the walls of the arteries that supply the myocardium (the muscle of the heart) with oxygen and nutrients (2).

As we'll present images of different arteries throughout the thesis, we introduce now the anatomy of the heart in figure 1.1.

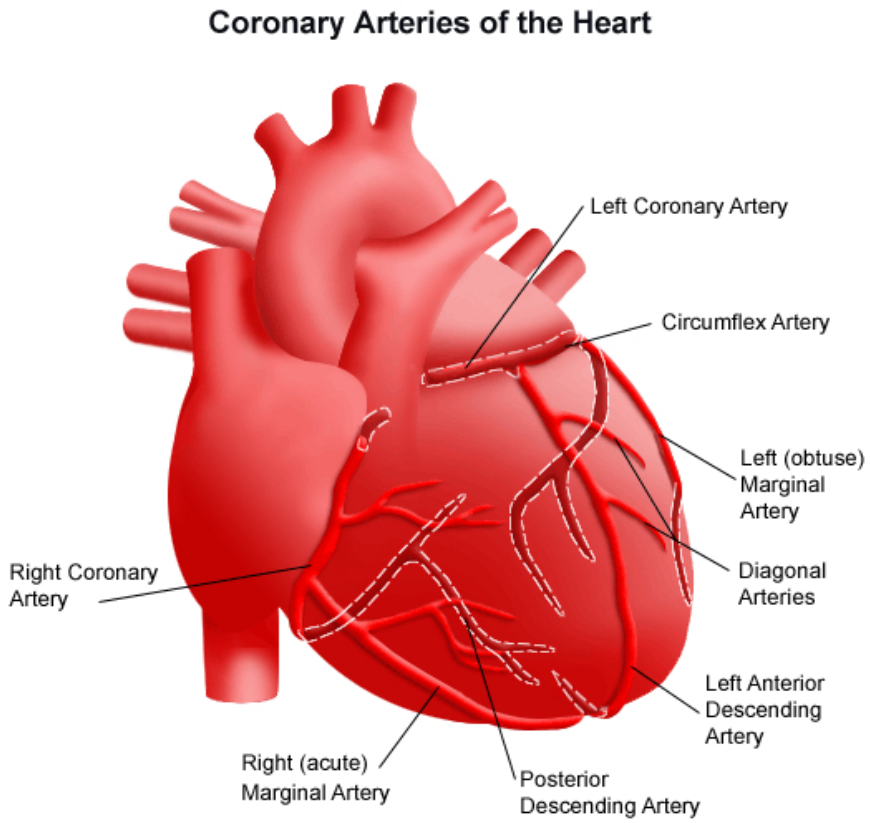


Figure 1.1: The arteries of the heart

For our project we'll analyze the plaque of the histology images of four of these arteries: the Right Coronary Artery (RCA), the Left Circumflex Artery (LCX), the Left Anterior Descending Artery (LAD) and the Obtuse Marginal Artery (OM).

It is also important to know the characteristics and layers of the arteries. We can see in figure 1.2 (3) the opening inside the artery where the blood flows, it is the lumen.

The first layer that we find is called the intima and it is made up of a single layer of endothelial cells. This is where the lipids are deposited to form an atherosclerotic plaque. The second layer is called the media and it is made up of smooth muscle cells and elastic tissue. The third and outermost layer of the artery is called the adventitia which is mainly composed of collagen which gives stability to the blood vessel. Between the intima and the media we find the Internal Elastic Lamina (IEL) and between the media and the adventitia we find the External Elastic Lamina (EEL).

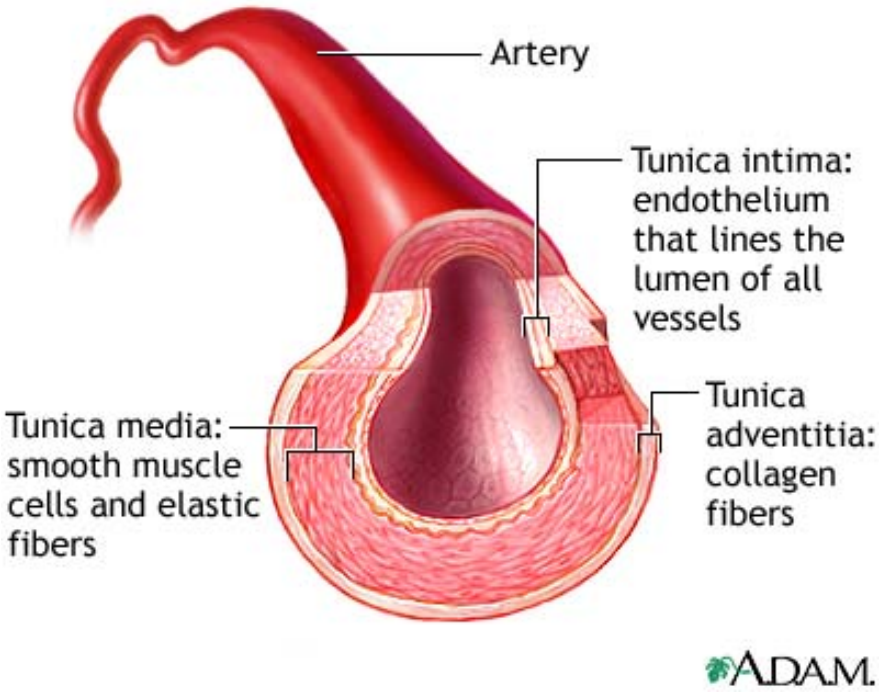


Figure 1.2: Anatomy of the arterial wall

1.1.1 How does the plaque form?

The plaque that is formed in the inner lining of an artery consists of cholesterol, calcium, fibrin (a protein involved in the clotting of blood), cellular waste products and other substances.

On the early stages the accumulation is formed of white blood cells, especially macrophages (cells within the tissue that originate from monocytes) that have taken up oxidized low-density lipoprotein (LDL). After they accumulate large amounts of cyto-

plasmic membranes (with associated high cholesterol content) they are called foam cells. When the foam cells die, their contents are released, which attracts more macrophages and creates an extracellular lipid core near the center of the inner surface of each atherosclerotic plaque. Conversely, the outer, older portions of the plaque become more calcific, less metabolically active and more physically stiff over time. This process of plaque formation is called atherogenesis (shown in figure 1.3 (4)) and the overall result of the disease process is called atherosclerosis.

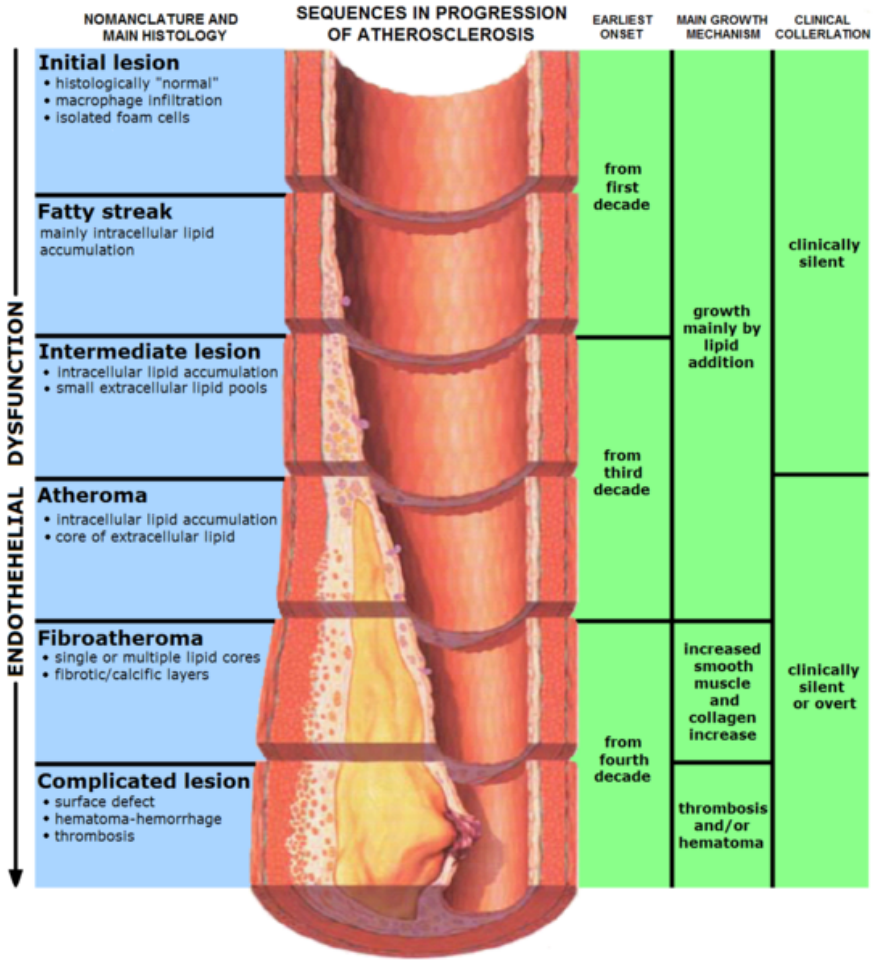


Figure 1.3: Evolution of a plaque

Atherosclerosis causes two main problems. First, the atheromatous plaques eventually lead to plaque ruptures and stenosis (narrowing) of the artery and, therefore, an insufficient blood supply to the organ it feeds, with the consequent damage of the

heart as seen in figure 1.4.

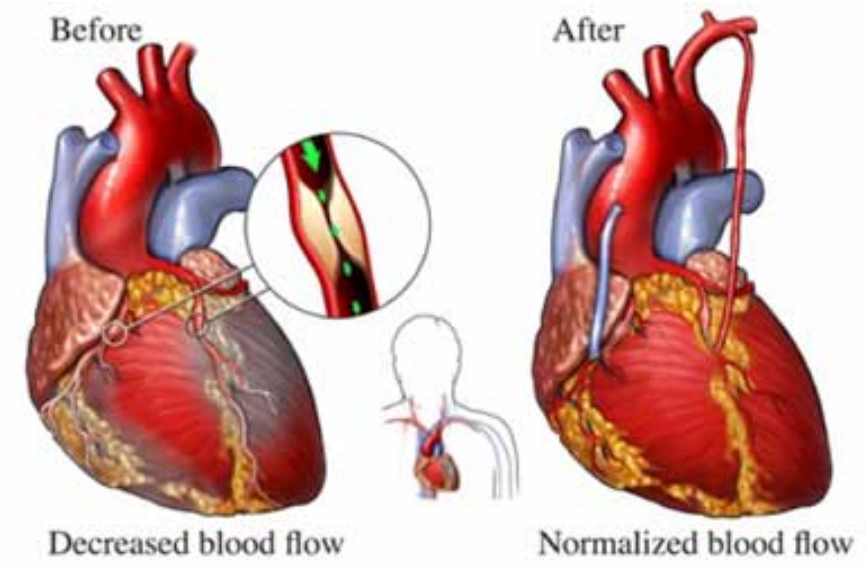


Figure 1.4: Healthy Heart vs. Damaged Heart

If the compensating artery enlargement process is excessive, then a net aneurysm results. An individual may develop a rupture of an plaque at any stage of the spectrum of coronary artery disease. The acute rupture of a plaque may lead to an acute myocardial infarction or heart attack.

1.2 Motivation and presentation of the project

Atherosclerosis is usually identified at its late stage after a major cardiac event. Several tests are used for the diagnosis of atherosclerosis including contrast angiography, CT angiography, Magnetic Resonance Angiography (MRA) and Intravascular Ultrasound (IVUS). IVUS is an *in vivo* medical imaging method that uses a catheter and a probe attached to its end. Using the ultrasounds technology it can help the visualization of the inner wall or intima of the blood vessels. There's a relatively new medical imaging technology called Optical Coherence Tomography (OCT) that uses advanced photonics to obtain high resolution images. Unfortunately, none of these techniques helps to detect this disease at its early stages. A better understanding of the atherogenesis and the formation of the plaque could have a huge clinical impact. Imaging methods

such as IVUS don't provide detailed information for the study of the coronary arteries from a cellular perspective, therefore *in vitro* studies of coronary arteries using stained histology images are essential.

The data used in this thesis was obtained from ongoing research to follow the disease and hemodynamic environment changes as the disease progresses from onset of early plaques to more advanced stages in a diabetic pig model. The study is being conducted in collaboration between Northeastern University and Brigham and Women's Hospital. It was designed to test the hypotheses that arterial subsegments with low Endothelial Shear Stress (ESS) are the regions where plaque will develop and progress, and in particular, plaque with high-risk characteristics (i.e. large lipid pool, inflammation, thin fibrous cap, internal elastic lamina degradation, expansive remodeling) will develop in those subsegments with the lowest level of ESS. The hypothesis was tested in a diabetic, hyperlipidemic porcine model, known to develop human-like atherosclerotic plaques. At different stages of the disease, coronary arteries were harvested and frozen cut at predetermined locations and cryosectioned (7 μm thick sections). The sections were then stained to indicate the desired cellular properties and later photographed with a camera attached to a microscope. They used four different stains in order to bring out different parts of the artery, H&E, Verhoeff, CD45 and Oil Red O. Verhoeff's elastin allows us to see the IEL and EEL of the artery which appears in a darker color, while Oil Red O gives the lipids of the artery a pink or red color.

After being stained, the images are used for the quantification of the subendothelial deposition of lipids, by manually counting the pixels of similar color using Adobe Photoshop, a program still widely used for manual analysis of medical images (5; 6). We clearly see that this process is long and often inaccurate which means that the number of images that can be analyzed is always limited. Therefore, the main motivation for this research is to provide more tools to help analyze a large number of histology images automatically or semi-automatically which will eventually lead to a better understanding of the process of plaque formation and so will help detect the disease in its early stages.

Chapter 2

Objectives of the thesis

We have seen in the introduction that being able to analyze large sets of histological images is a key point in the understanding of CAD. Before starting with the discussion of the methods used, we need to understand our data set to see what we want to achieve and what is the best approach we can take. In this chapter we will state the primary objectives of the thesis, analyze the histological images we have used, discuss the existing methods used to analyze them and briefly present the tools we have used to improve this analysis.

2.1 Main objectives

Before starting the project we talked to the doctors who provided us with the images and we established the objectives of the thesis based on their needs.

The main objective is to automatically segment the artery from the IEL to the EEL, leaving in the middle the Region Of Interest (ROI) which is the place where the plaque is deposited.

The second objective is to quantify the amount of lipids deposited in the ROI to know if the disease is on its early stage or it is clearly developed.

To present the tools to analyze the images we agreed that it was convenient to design a Graphical User Interface (GUI) to efficiently work with the images.

2.2 Our data set: the histological images

As we stated in section 1.2 there are four types of stains and every one of them gives us different information about the artery.

We have sections of different arteries: LAD, RCA, LCX and OM (as described in section 1.1).

Every image is coded in a consistent way so we know to which animal the artery belongs, the specific artery we are looking at, the particular section of the artery, the stain used and the magnification of the microscope. So just by reading the title of the image we would know:

Title: 10346 LAD A Ver 2X

Animal number 10346

Artery: Left Anterior Descending Artery

Section: A (every artery is cut into different sections, this way we know that section B goes after A)

Stain: Verhoeff

Magnification: 2X

That way we can keep track of all the arteries and document their evolution. It is also important to find out which arteries tend to deposit more lipids.

We have seen in section 2.1 that we want to segment the ROI presented between the IEL and the EEL. The stain that best shows us these two laminas is the Verhoeff, therefore we will start our work on this images to obtain our ROI using different segmentation techniques.

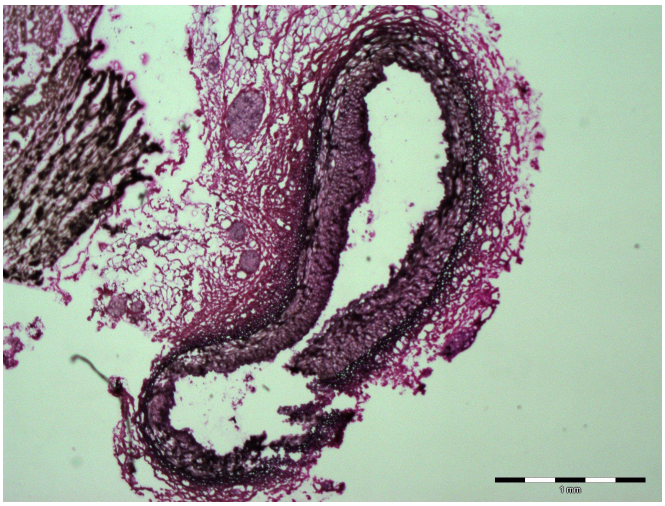


Figure 2.1: Artery 10346 LAD A Ver 2X

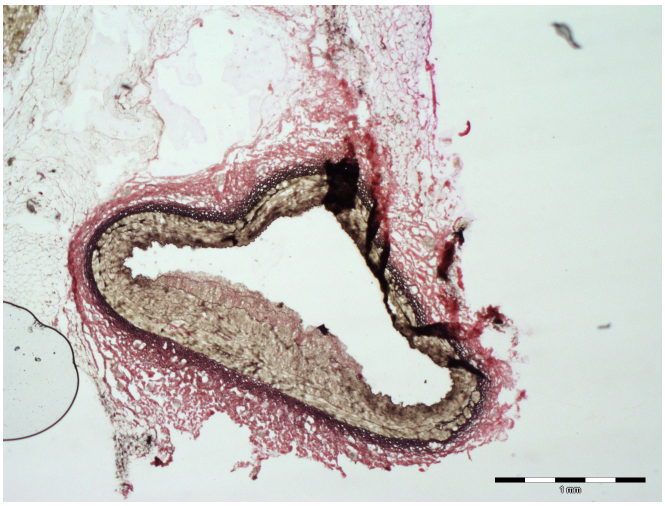


Figure 2.2: Artery 10348 RCA A Ver 2X

As we can see in Figures 2.1 and 2.2, the EEL is specially enhanced with a dark brown or dark purple color. The lumen can be easily segmented as it has a light white

or bone color.

In order to quantify the amount of lipids present in the IEL of the artery we have to turn to the Oil Red O images that enhance this particular characteristic.

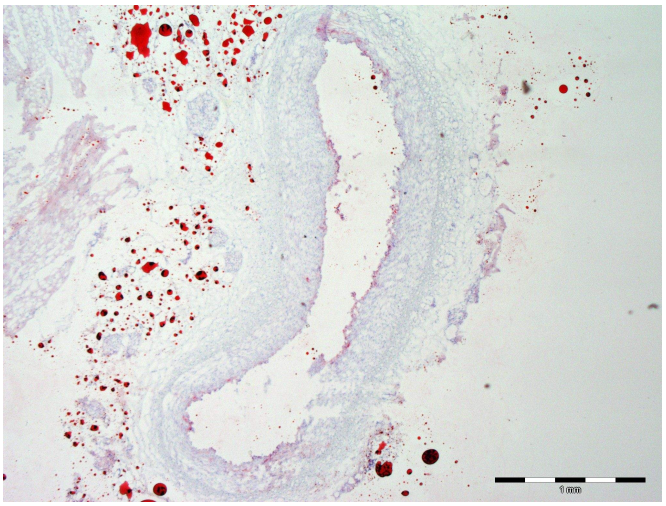


Figure 2.3: Artery 10346 LAD A Oil 2X

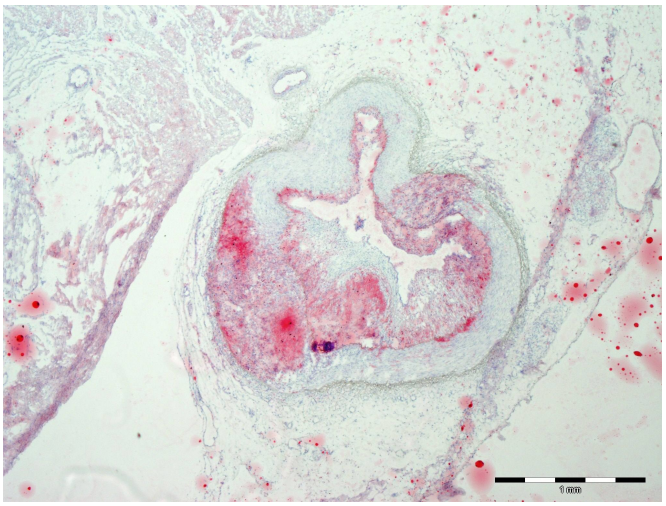


Figure 2.4: Artery 10348 LAD E1 Oil 2X

We can see in Figure 2.3 an artery that has not developed any plaque and therefore there are no red pixels around the IEL. In Figure 2.4, on the other hand, we can see an artery with a large quantity of lipids deposited in the IEL. The plaque has narrowed

the lumen (stenosis) and therefore the blood flow may be compromised. We can see there are red pixels also outside of the artery created during the staining process, that's why it is critical to correctly segment the images in order to count only the red pixels inside the artery.

We have seen that the images we need to correctly segment the arteries and quantify the amount of lipid are the Verhoeff and the Oil Red O, therefore we will only focus on those two images during the thesis. Our final data set consists of 52 images of magnification 2X, of which we will only use 42. We classify 10 images as incorrect for various reasons: the artery is broken, the image magnification is too high, etc.

2.3 Our starting point

Before starting the work on the project we must understand how the images are currently segmented and the previous research done on this problem, to see what the starting point of our project should be. To analyze histologic images of arteries doctors currently do manual segmentation on the images using an image program such as Adobe Photoshop. This involves around 20 to 30 mouse clicks, which can lead to errors and limits the amount of images that can be properly analyzed.

A Masters student at Northeastern University, George Masganas, started this project in 2007 (7). He worked with a data set of 12 images. He correctly segmented the inner boundary by just clicking once at the center of the lumen and using region growing. In order to find the outer boundary or EEL he used a technique called livewire in which the user had to click around 10 points to correctly segment the image. Now the segmentation was reduced from 20-30 clicks to 10 clicks, which made the process faster but not automatic.

The main reason to start this project was to follow the previous work and get a step closer to the complete automatic segmentation of the histology images of arteries.

2.4 Presentation of the methods used in the thesis

2.4.1 Finding the inner boundary

The method used in the previous project to segment the inner boundary or lumen region was clearly efficient as it only needs 1 click done by the user inside the lumen

region, which does not require much precision. As the outer boundary finding process is the one that is less automatic, we will focus on finding a new method to segment it and we will continue using region growing for our lumen boundary.

2.4.2 Finding the outer boundary

The method used to find the EEL is totally different from the one used by George Masganas and it is based on the color information that our image has.

2.4.2.1 Pre-processing

If we want to use color information on all the images, we have to make sure that they are all similar. As we will see later on, we are presented with pink, brown and purple images. We will use **histogram equalization** on all of them to make our span of images more uniform.

2.4.2.2 Creation of the color model and the probability image

We clearly see that the unique characteristic of the boundary we want to detect is color. We can say that the pixels that correspond to the boundary have a different color than most of the surrounding pixels. Therefore, we will **create a color model manually** by clicking EEL pixels and taking their Red, Green and Blue values to create a graphic that tells us how many of the clicked pixels have a certain range of R, G and B values. Once we have this model we want to use this information automatically on all the images that we have. To do this, we'll scan each pixel of each image and we'll assign a probability value of the color model based on its R, G and B values. At the end of this process we'll have an image with range [0,1] that will show us which pixels are more probable to be EEL. Ideally our **probability image** would be white on the EEL pixels and black elsewhere. Of course, we don't have an ideal image with just white and black pixels, therefore we apply the morphological operator of **erosion** to clean the image before going to the next step.

2.4.2.3 The use of splines and snakes

Once we have our probability image we're interested in finding only the white pixels that correspond to the EEL in an automatic way. To do that we will start from the

inner boundary that we have correctly found and we will expand this contour using dilation until we are sure that we are outside of the artery region. We will sample this starting contour and only work with several points and create the rest of the contour by using **splines**. In order to make this contour fit the EEL marked by our white pixels, we will consider it as a **snake or active contour** and apply several energy terms that will drive our snake to the desired boundary.

2. OBJECTIVES OF THE THESIS

Chapter 3

Pre-processing of the images

3.1 Color differences

As we mentioned briefly in chapter 2, our data set of histological images contains different cuts of different arteries all stained with Verhoeff. Although the process of staining is always the same, there can be differences in the resulting images due to, for example, the amount of stain used.

This creates color differences in the images; after analyzing the image set we distinguished three types of images: purple (figure 3.1), pink (figure 3.2) and brown (figure 3.3). Note this classification is based solely on visual aspects, not on mathematical characteristics.

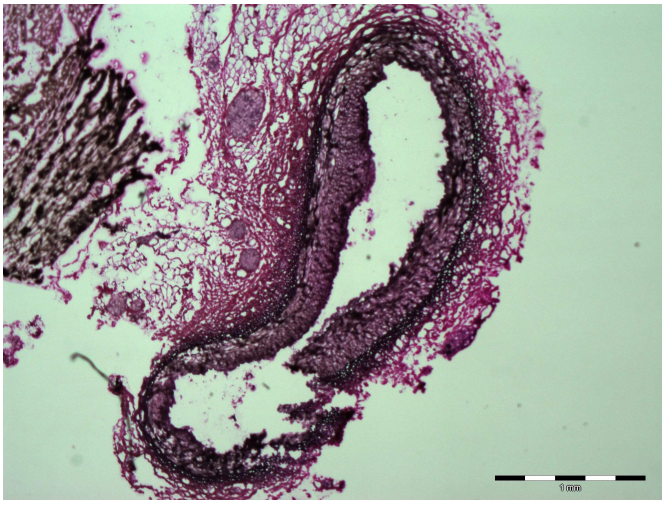


Figure 3.1: Image classified as purple

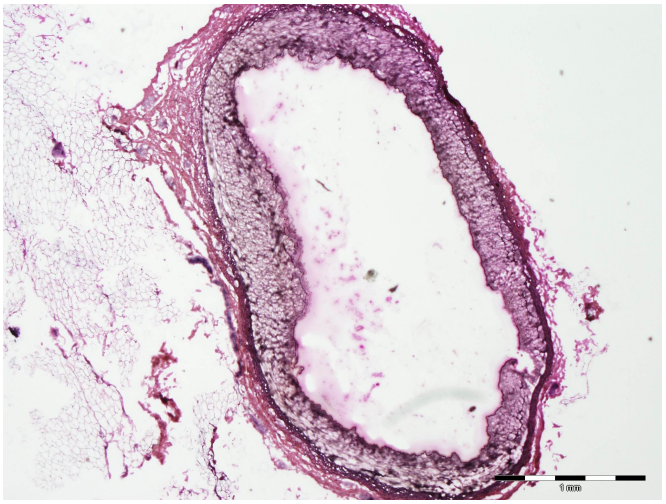


Figure 3.2: Image classified as pink

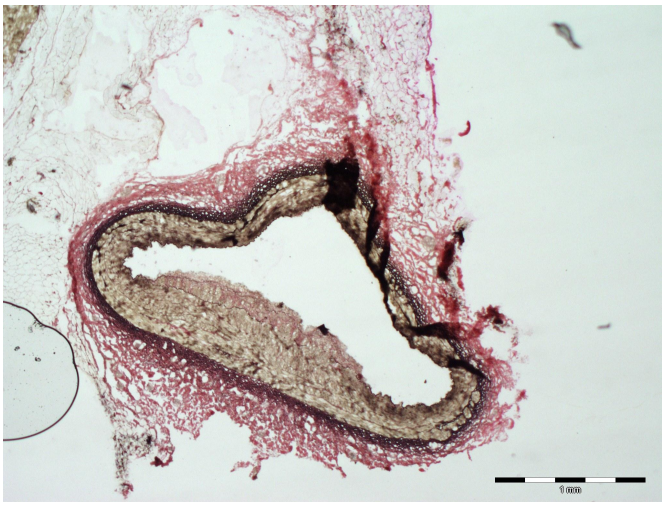


Figure 3.3: Image classified as brown

As we stated above, we are going to base our analysis on the color information so we need to have images more uniform in color.

3.2 Histogram equalization

A well know method to make any characteristic of a signal uniform is **equalization** (8). In this case we are going to work on the histogram, a graphic display that shows

the distribution of pixels according to its color components. In our case we work on the histogram of each color component separately, so we have a histogram for R, another for G and another for B. Each histogram has an x axis that goes from 0 to 255 that represents all the values of that component that a pixel can have (since we have an image that is uint8), and on the y axis we have the number of pixels that have each of the x values. Below in figures 3.4, 3.5 and 3.6 we have an example of the three color component histograms for the image 10346 LAD A Ver 2X.

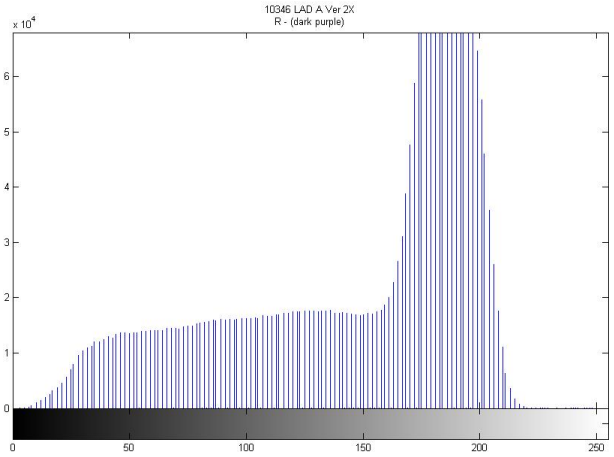


Figure 3.4: 10346 LAD A Ver 2X, Red component, Histogram

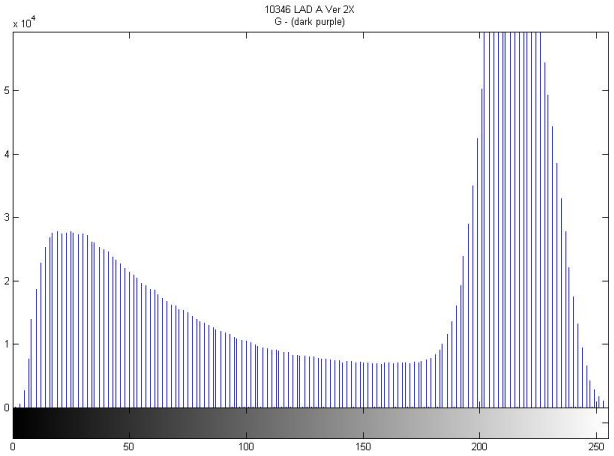


Figure 3.5: 10346 LAD A Ver 2X, Green component, Histogram

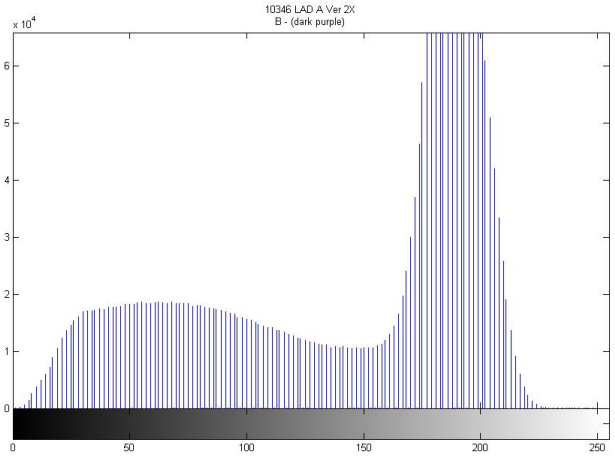


Figure 3.6: 10346 LAD A Ver 2X, Blue component, Histogram

Histogram equalization is a technique used to have a more uniformly distributed histogram. It accomplishes this by effectively spreading out the most frequent pixel values. This way we eliminate the "peaks" of the histogram. This creates false colors on the image, but since we don't need the exact color information but rather the differences in color between the EEL and the media and adventitia, we can successfully apply this technique. We apply histogram equalization using the *histeq* MatLab function.

Below in figures 3.4, 3.5 and 3.6 we can see how the histograms are transformed after applying the equalization.

Now we are going to see the results of the histogram equalization by looking at the transformation that three representative images (purple, pink and brown) have suffered. The original images appear in figures 3.1, 3.2 and 3.3. We can see the equalized versions in figures 3.10, 3.11 and 3.12.

As we can see, all the images have now more similar colors. Now we are ready to implement an algorithm that takes advantage of the color differences between the EEL and the adventitia and media, without having to adapt it for three different types of images (purple, pink and brown).

Note that the algorithm was tested too for the three types of images, purple, pink and brown, so three color models were created with three components R, G and B each. But using histogram equalization and just one color model gave better and more consistent results in the end.

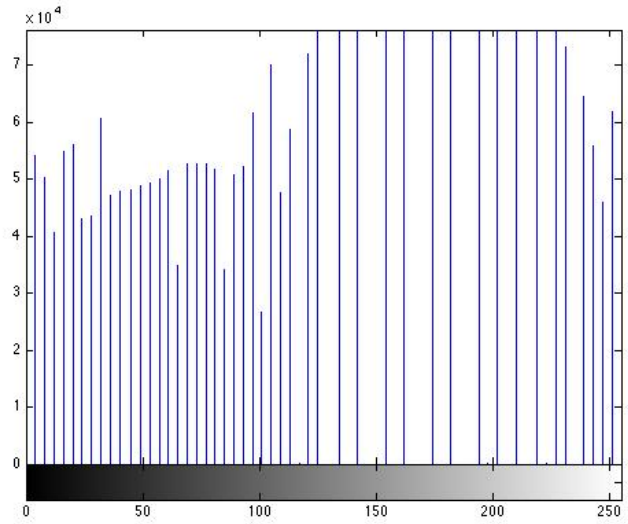


Figure 3.7: 10346 LAD A Ver 2X, Red component, Histogram Equalized

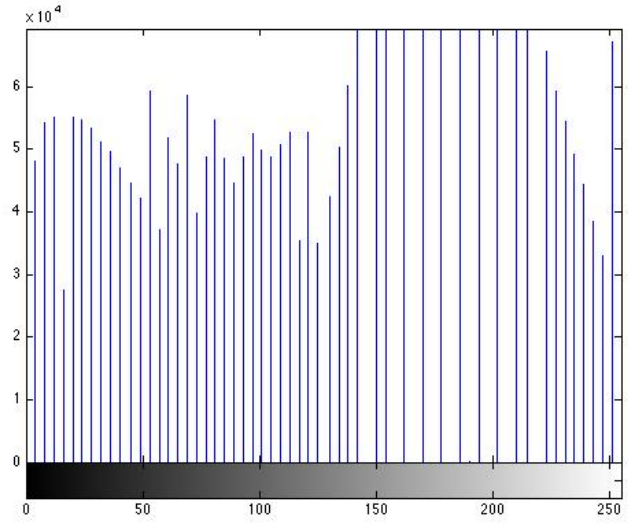


Figure 3.8: 10346 LAD A Ver 2X, Green component, Histogram Equalized

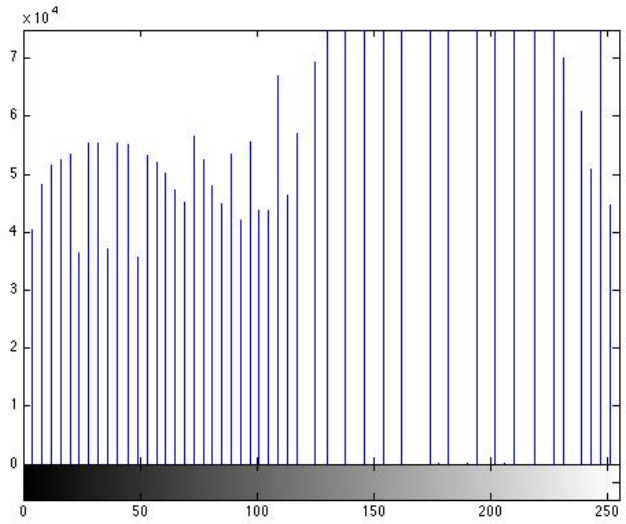


Figure 3.9: 10346 LAD A Ver 2X, Blue component, Histogram Equalized

Figure 3.10: Image classified as purple equalized

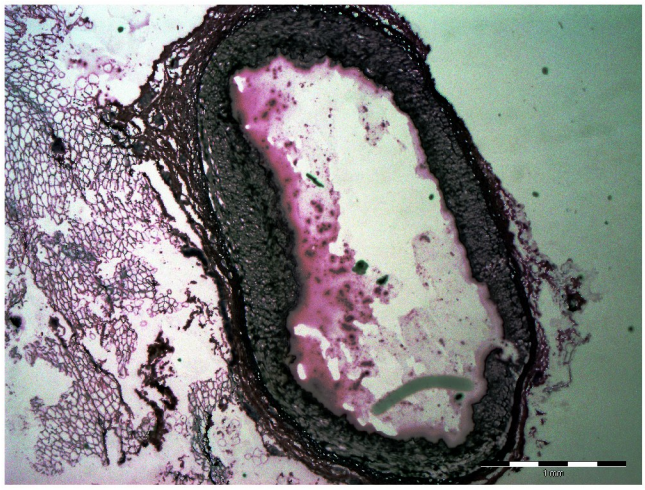


Figure 3.11: Image classified as pink equalized

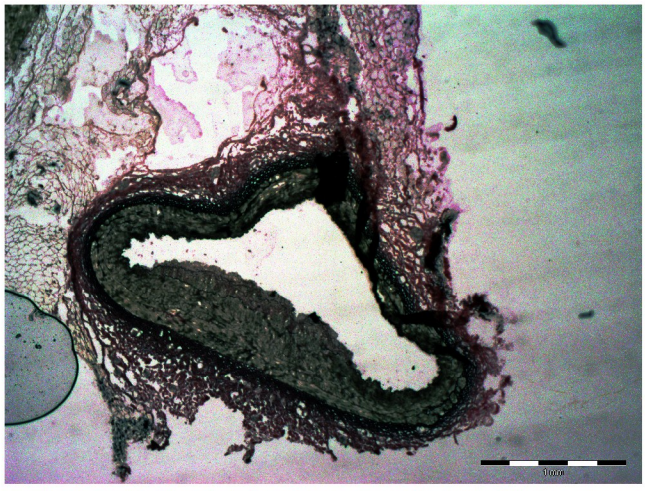


Figure 3.12: Image classified as brown equalized

3. PRE-PROCESSING OF THE IMAGES

Now that we have our images pre-processed we are ready to explain with detail the methods applied to solve the segmentation problem.

Chapter 4

Automatic segmentation of the inner boundary

As we presented in previous chapters, the method that we are going to use to determine the inner boundary or lumen region is the same as the one George Masganas applied in his Masters Thesis. The technique we use is the well-known region growing (8).

4.1 The algorithm

Region growing works very well and in a consistent way for our images because the lumen, which is the part that we want to segment, is always a bone color that is very different from the dark color of the IEL. We use only the intensity information of the image, so basically we apply the algorithm on a grayscale image. We automatically apply this algorithm (Algorithm 1, (9)) to all the images but first we need to manually determine the seed for every image. We do this manually by clicking at the center of the lumen. Our algorithm gets an NxN window centered at our seed pixel and choose the lowest intensity value within that window. This is considered as value V and the seed pixel is the one with such value.

This returns a closed region with all the pixels with similar values to our seed pixel. In our case, the region is the lumen of the artery. Here are a sequence of images that show how the region growing algorithm evolves.

As we can see, the region growing method evolves from Step 1 when we choose the seed manually to Step 5 when the region covers all the lumen of the artery. In Step 6

Algorithm 1 Region Growing

Require: V = intensity value of the seed pixel, R = initial region (seed pixel), T = threshold

repeat

Take $R' > R$

if $p \in R'$ has intensity value V' and $\mid V' - V \mid \leq T$ **then**

$p \in R$

end if

until No 8-connected pixel satisfies the condition to belong to the region

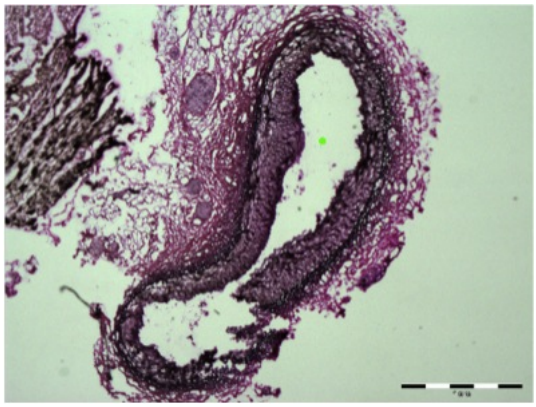


Figure 4.1: Step 1

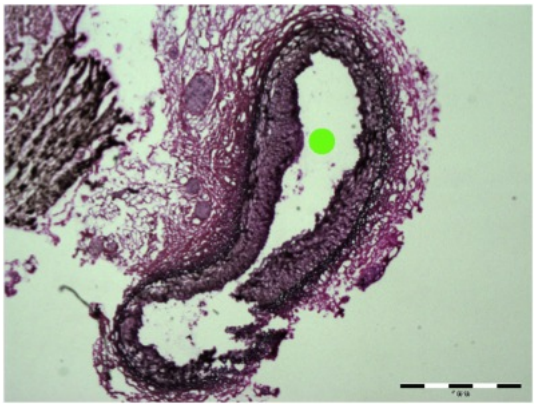


Figure 4.2: Step 2

we just keep the boundary of that region which is what we are interested in.

4.1.1 The threshold value

The most important parameter of the algorithm is the threshold value. Here we present a graphic with the distribution of the values for all the 42 images for which the algorithm correctly detects the region of the lumen.

As we can see, a value of T=40 can be used to automatically find the inner boundary of most of the images (71%). Only 1 of the 42 images of our data set needed T=30. On the other hand, the other 26% left needed to yet adapt another parameter to correctly determine the lumen boundary. In the process of cutting and freezing the arteries, the tissue might deform and the walls of the IEL might come together, leaving the lumen divided into 2 regions. Therefore, in some images we might need to specify more than one seed so the algorithm detects the 2 or more separate regions of the lumen. We have

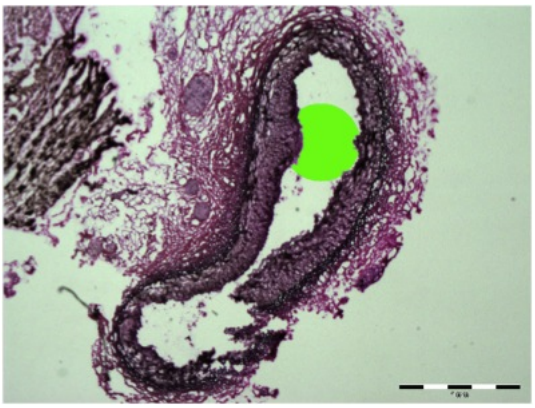


Figure 4.3: Step 3

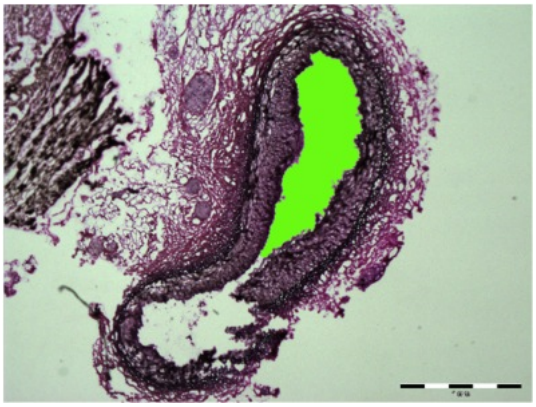


Figure 4.4: Step 4

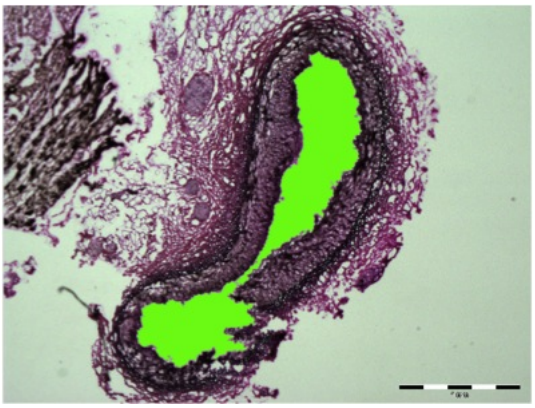


Figure 4.5: Step 5

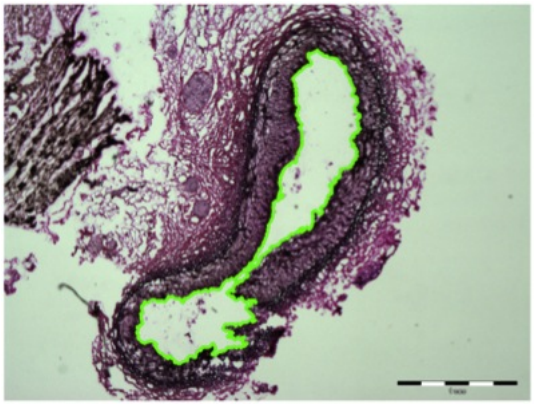


Figure 4.6: Step 6

taken this into account to design the Graphical User Interface (GUI) so the user can modify these parameters in an easy way. The threshold value that we use as default is T=40.

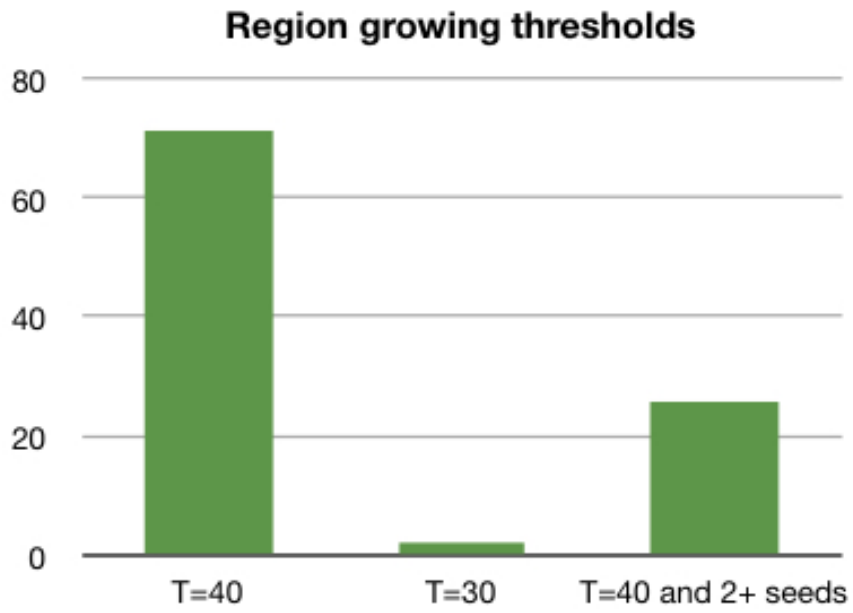


Figure 4.7: Distribution of the threshold values used

Chapter 5

Color model and probability image

We have seen an outline of how we will create the color model using the color information of our image. Color algorithms are very important for medical image analysis (10; 11; 12). In this chapter, we explain exactly how our algorithm works.

5.1 Model training

Our first step to create the model is to define which R, G and B values we consider as "correct boundary". To do that, we choose several points in an image that we consider to be on the EEL and tag them as "boundary points". We show an example of this process in figure 5.1 in which the yellow dots mark the points that we mark as boundary.

We do that on several images (already equalized) so the model can take into account the little color differences that can appear between images. We manually click on 15 boundary points of 9 different images that we think represent the different types of artery images we have in our data set. These points are the center of a square window of 11x11 pixels and we take all the pixels in the window to create the model in order to incorporate the little color differences within the boundary.

$$N_p = N_i * N_p t * w^2 \tag{5.1}$$

As we can see in Equation 5.1, we have a total of 16335 boundary pixels to create our color model. We consider this to be enough to create our model considering we use

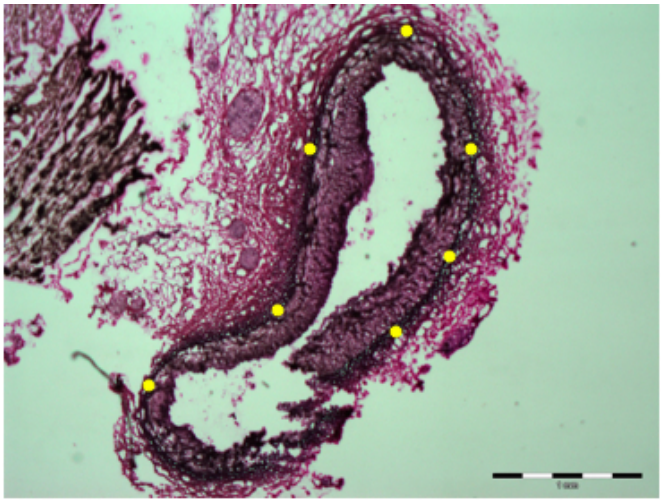


Figure 5.1: Marking the boundary points to create the color model

M bins with M being lower than the 256 color values that we originally have. In order to choose the points easily, we also incorporate this step in our GUI.

5.2 Creation of the model

Now that we have chosen the correct boundary pixels, we have to extract the necessary information from them and create a model that we can use later on. As we are interested in the color of these boundary pixels, we create a histogram for each of the color components R, G and B of the pixels tagged as boundary. If we do the histogram with the 256 values for each component, we won't have enough boundary pixels to get an accurate histogram. That is why we are more interested in: having a more compact graphic and fewer values to represent in the x axis. Instead of 256 bins, we use M bins, After trying several values for M, we have found that M=20 is the one that gives us the best results.

To create the R histogram, we count the boundary pixels that have a red component value that falls into every one of the M bins. Then we convert those counter values into probabilities, therefore dividing them all by the maximum value. We do the same with the other two color components, B and G, and obtain the histograms shown in figures 5.2, 5.3 and 5.4 (image 10346 LAD A Ver 2X).

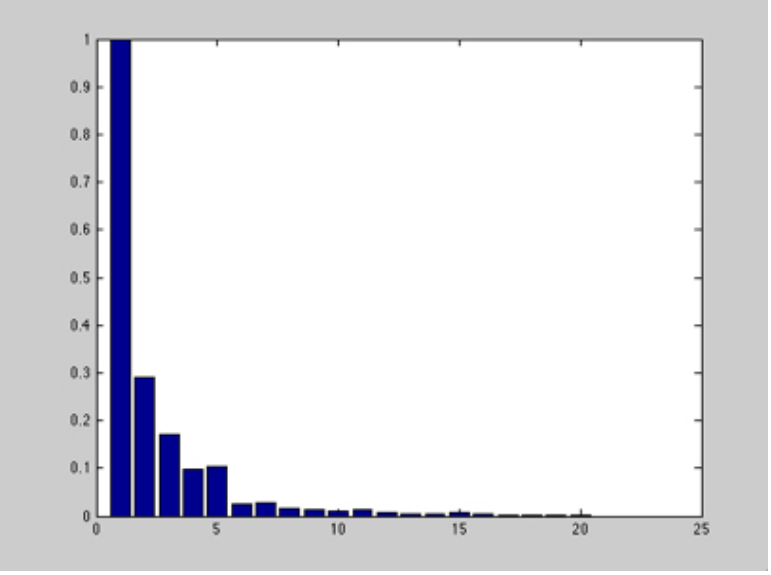


Figure 5.2: Red histogram for image

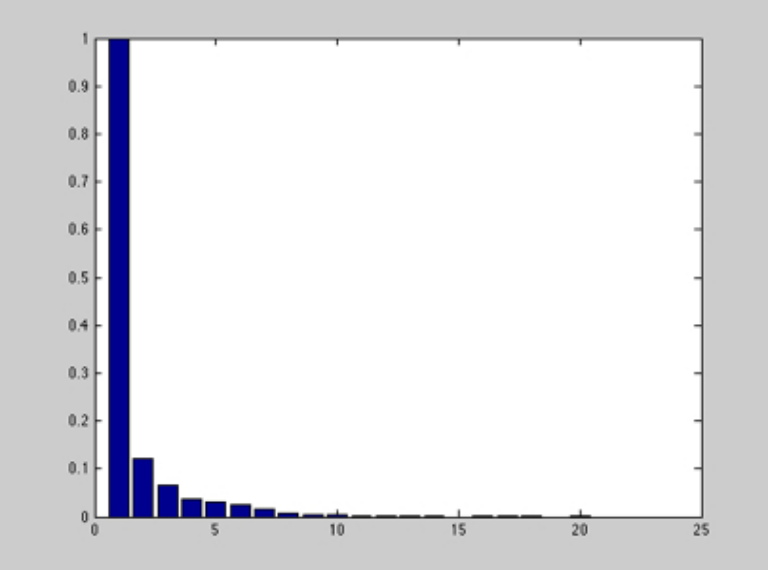


Figure 5.3: Green histogram for image

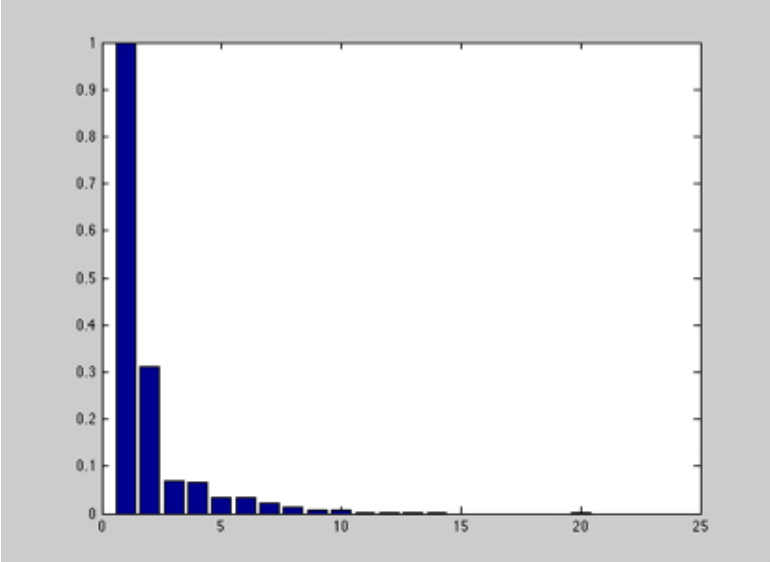


Figure 5.4: Blue histogram for image

As we can see, the probability accumulates in the lower bins, where the dark colors are represented. This is consistent with the pixels we are trying to represent, which have very dark colors. Now that we have created the boundary model with the three components R, G and B, we are interested in translating this into actually detecting the EEL of the image. For that, we have created the probability image which we will explain in the next section.

5.3 Probability image

The probability image will help us translate the information of the histogram of the color of the boundary into the 2D image by finding which pixels fit into the model and will give us an idea of the distribution of these pixels.

In order to accomplish that, we scan every pixel of the image we are analyzing and we find its R, G and B values. After that, we look at the model of each component, find the bin that holds the value that we are looking for, and assign the probability value to that pixel. Now we have 3 probability values assigned to each pixel, one for each color component, and we convert them into one value with the relation expressed in 5.2.

$$P = M_R * M_G * M_B \tag{5.2}$$

With this expression, we make the approximation that the color components are independent. This is not entirely true because it's not the same to have R=0.2, G=1, B=1 as R=1, G=1, B=0.2, although these combination of values give the same P under our considerations. We consider this approximation correct given our image set and the colors that we are working with. Giving dependency to the expression doesn't improve our results but it does require more tagged boundary pixels and a 3D model, which is more complication for no improvement in the probability image.

Now that we have only one value associated with each pixel, we create an image to represent these probability values. These values have a range of [0 1], therefore we are going to obtain a grayscale image. We will normalize this image to the highest value to give more contrast to the image, because when we multiply three probability values under 1, the total P value obtained is very small.

If the model is correct, the probability image should be white (value 1) in the boundary pixels and black (value 0) elsewhere. In figure 5.5 we can see an example.

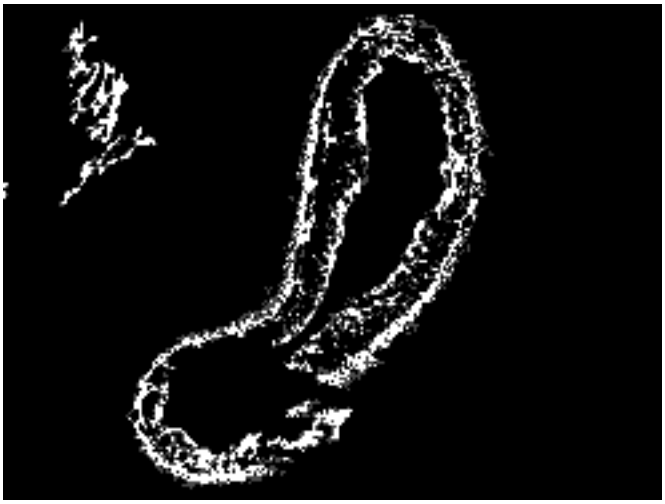


Figure 5.5: Probability image of 10346 LAD A Ver 2X

We can see that we have simplified the image and we have obtained a representation that clearly shows the boundary of the artery.

Note that the scale of the picture appears in all the images. As it is black and white it could interfere with the algorithm. That is why we cover that region with a black mask created manually.

5.4 Problems:how to solve them

Of course this method presents some problems. We are going to find them and quantify them and see which steps we can add to the algorithm to solve them and make it more robust.

5.4.1 Little white artifacts

The first problem that we see in figure 5.5 is that we have some white pixels on the upper left part of the image that don't represent the artery. These white pixels in the probability image correspond to dark pixels in the original image (according to the color model created). These dark pixels appear in several images and can be caused by staining of tissue that doesn't belong to the artery or simply stain accumulations. In order to reduce this problem, we use morphological operators, a technique often used in medical imaging (13; 14; 15; 16). We apply **image reconstruction** using an eroded version of our image as the marker.

Image reconstruction is a popular technique used to eliminate white artifacts (17). The marker is a binary image with a black background and some white pixels that represent the marker. Image reconstruction puts the marker image on top of our image and just keeps the closed objects that are in contact with the white marker. Here we show a visual example of how we can reconstruct our image 5.5 by using the image marker 5.6. We will obtain the result shown in 5.7.

As we can see, the reconstructed image just shows the artery part which was in contact with our white circle marker. We have effectively eliminated the undesired white artifact. We can express this process in a more formal way, image reconstruction is an anti-extensive, increasing and idempotent operator. Considering X as our original image and Y as our marker, we can say that image reconstruction is a transform that preserves the connected components of X that are marked by Y.

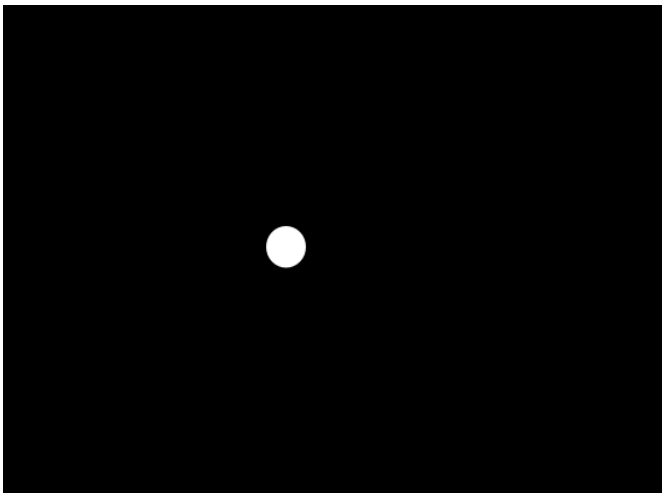


Figure 5.6: Marker image

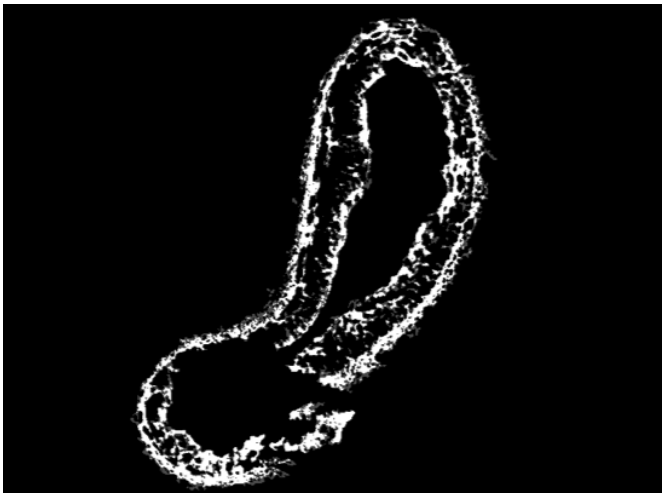


Figure 5.7: Probability image reconstructed

We start the reconstruction process with the marker, we apply the minimum dilation possible and use the operator AND with our original image. We will repeat this process always dilating the resulting image, as expressed in 5.3.

$$\delta(...\delta(\delta(Y) \wedge X)... \wedge X) = (\delta_1^X(Y))^\infty = \delta_\infty^X(Y) \tag{5.3}$$

We can say that the succession of dilations and AND operators creates the reconstruction as stated in 5.4. Note that it's not the same to do dilation+AND λ times, as to dilate the marker λ times and then apply the AND operator, as shown in 5.5.

$$\gamma^{rec}(X;Y) = (\delta_1^X(Y))^\infty = \delta_\infty^X(Y) \tag{5.4}$$

$$\delta_\lambda^X(Y) \neq \delta_\lambda(Y) \wedge X \tag{5.5}$$

Now we have to decide a marker image that can work with all the images. The problem we face is that none of the images is similar: the shapes of the arteries vary, the sizes vary, etc. therefore we can not choose a single marker image that can work for all our data set.

What we do is to apply yet another morphological operator, erosion, to our original image to create our marker image. With a significant erosion, we hope to keep just part of the artery and erase the white small artifacts. When used as marker image, this will just recover the artery shape.

Now we study which values of erosions give us good probability images and we obtain the following statistics 5.8. By value of erosion we mean the size of the disk that we will use as structuring element to apply the erosion.

As we can see, a little bit over 50% of the images require erosion=50. But if we use a fixed value for the erosion, we will have to modify almost 50% of the values manually. Therefore, we create an algorithm (Algorithm 2) to calculate a value of erosion in a more or less adaptive way. The biggest problem of our images is when we erode them too much and so there are not enough whites to show the shape of the artery. Just 5 of the 42 images present problems because there is not enough erosion. That is why we start with a value of erosion of 50 as it is the correct value for half of the images, and we reduce erosion if the number of white pixels left in the image is less than 60% of the original white pixels. We repeat this process a maximum of 4 times, so the

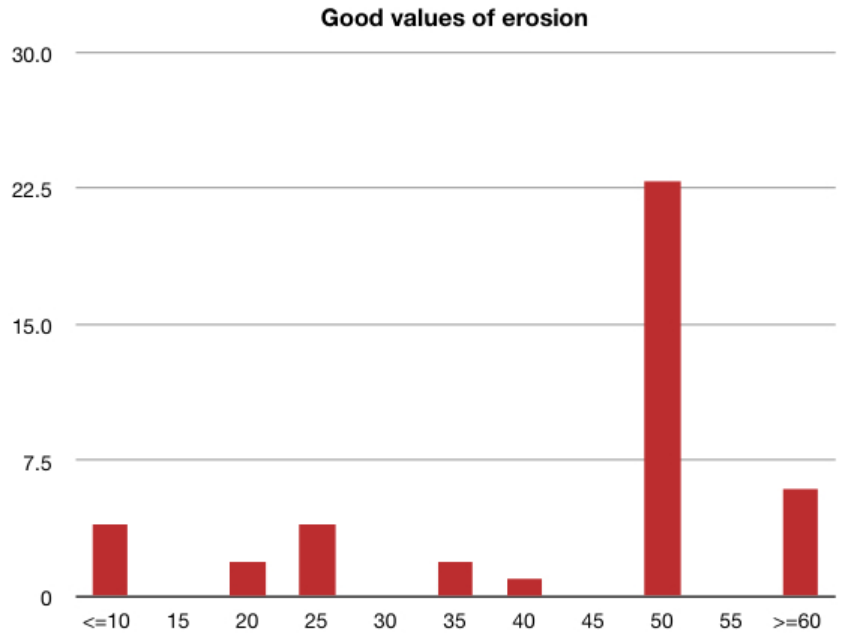


Figure 5.8: Correct erosion values

minimum erosion value it can reach is 30. We are aware that we have 10 images with ideal erosion of less than 30, but if we apply the reduction more than 4 times we obtain worse results.

With this algorithm we obtain a variety of erosion values that is more similar to the good values seen in 5.8. Instead of having just 50% of correct erosion values, we'll now have the distribution that appears in 5.9.

As we can see we have 60% of the images with a correct erosion value found automatically. This improvement of 10% may not seem remarkable, but as we will see in the results later on, from the 17 images with incorrect erosion, just 3 will depend solely on the erosion parameter to go from badly segmented to well segmented. Therefore, we can say that what is important for us now is that we have 5 more images that will be correctly segmented automatically.

After this process we finish cleaning the image by erasing all the values of the probability image lower than 0.01, and giving them the value 0. This way, we make sure that all the pixels in the background are always 0. This will be very important for the calculation of the snake starting position and we will see in future chapters.

Algorithm 2 Automatic calculation of the erosion parameter

Require: Erosion parameter $ERO = 50$; # of white pixels of original image = N_w

repeat

Apply erosion with value ERO. Calculate new N'_w

if $N'_w < 0.6 * N_w$ **then**

$ERO' = ERO - 5$

else

End program

end if

until This algorithm has been done 4 times

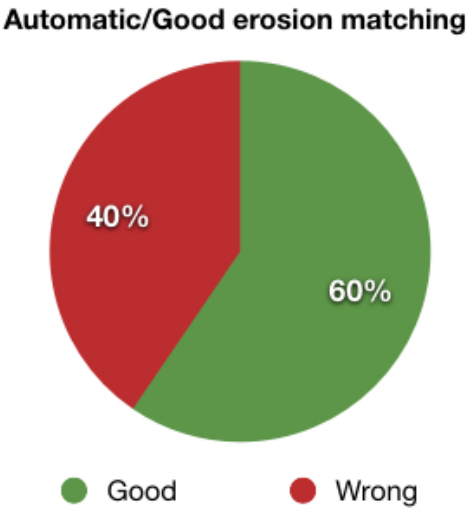


Figure 5.9: Probability images correct/incorrect

5.4.2 Missing boundaries and false boundaries

After successfully cleaning the image using reconstruction by erosion, we see that we have other problems in our image, which are bigger and therefore not erasable by the previous morphological operator.

If we look at figure 8.12 we can see that the boundary of the artery is perfectly defined but we have a white extension due to an accumulation of stain.

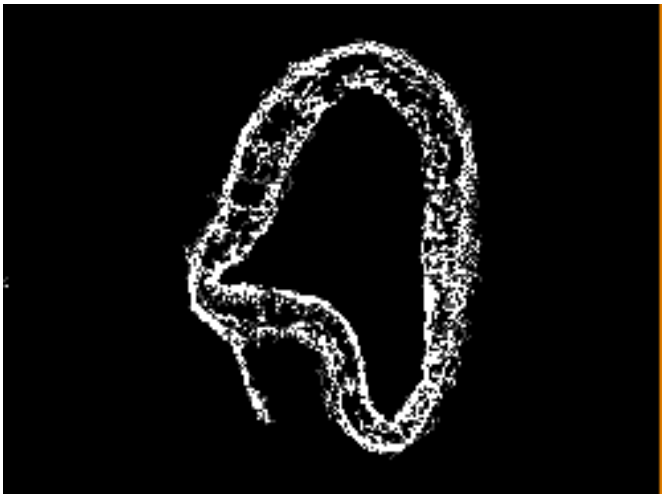


Figure 5.10: Probability image - Problem 1

Figure 8.11 shows us the artery boundary except in the lower part where the boundary disappears.

These two images are a representation of the two big problems that we will be facing in the next step of our algorithm. We will need to find a contour that can adapt itself easily to any shape that is able to ignore white artifacts, as the one in figure 8.12, but that is also able to "imagine" where the boundary is when some white pixels are missing, as in figure 8.11, based on the position of all the other boundary pixels. In order to do that, we decide that the best tool we can use is **snakes or active contours**. In the next chapter we will explain how we apply them to find the EEL of the arteries and we will see the importance of the inner boundary to find a good starting point for the snake.

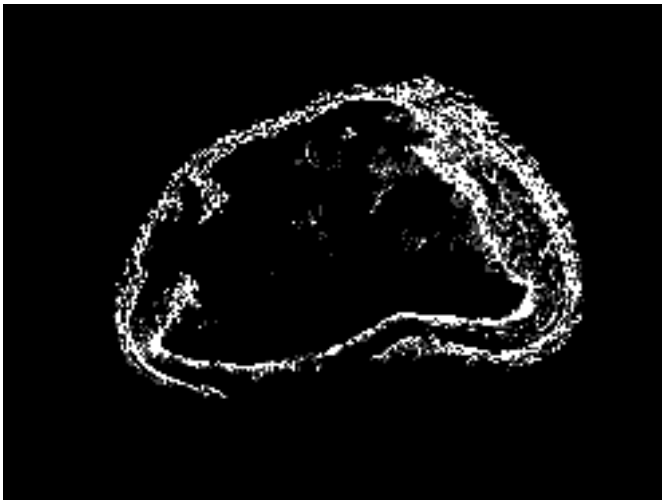


Figure 5.11: Probability image - Problem 2

Chapter 6

Finding the outer boundary

In this chapter we are ready to see how we have prepared our images to automatically find the outer boundary or EEL, the most difficult part of this thesis. In order to do that we will use snakes and splines and explain why we use these, what they are and how we apply them to our purpose.

6.1 Why do we use snakes and splines?

As we mentioned earlier, we have a variety of sizes and shapes of the arteries in our image set. This means that we can use little shape information to segment the EEL and we also have to take into account that some arteries can fill all the image while other are significantly smaller. In the previous chapter we created a probability image that greatly simplifies our segmentation problem. Now what we need is a tool that detects the "white pixels" contour, ignoring external white artifacts and other undesired objects. In order to do this, we decide to apply the **active contours** or **snakes** (18) which are closed curves that evolve towards the points with high energy, where energy is a parameter that we can define according to our needs. We can show how snakes work in a simple example, where we have a black background and a white circle. What we want is the snake to detect the contour of the white circle as in Figure 6.2. We represent the snake curve in red and its initial position is shown in Figure 6.1. Now we will drive the snake towards the white circle by making the curve: minimize its length so the snake goes towards the inside and approach the white pixels without crossing them. We have defined, in a simple way, the two energy terms that will drive our snake.

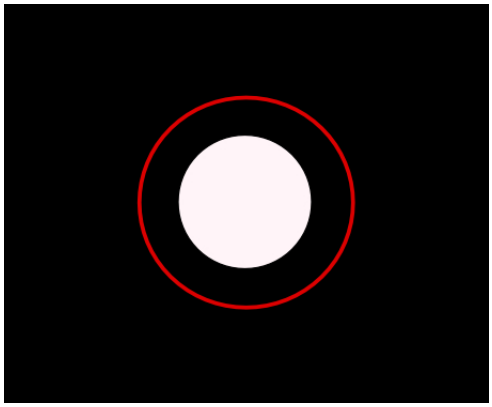


Figure 6.1: Example: Initial position of the snake

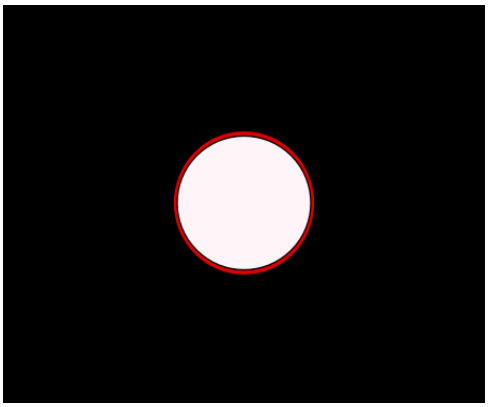


Figure 6.2: Example: Final position of the snake

In order to make this snake work, we need to assure that the starting point of the snake is out of the artery EEL. This is a very important step and will be discussed in the following section.

While applying the snakes we see that it is computationally costly to represent the snake as a set of N points where N has to be high if we want the snake to correctly follow the contour of the artery. Therefore, we have to figure out ways to simplify the representation of the snake. Even though we have seen that artery shapes are not consistent, we can safely say that all the arteries have rounded contours, so none of them has a sharp corner. These contours can be represented by polynomials and this is where the idea of applying splines comes in.

A **spline** (19) is a piecewise polynomial parametric curve. Mathematically we can define a spline $S : [a, b] \rightarrow \Re$ that consists of piece polynomials such as $P_i : [t_i, t_{i+1}] \rightarrow \Re$, where:

$$a = t_0 < t_1 < ... < t_{k-1} = b$$

Now we define the spline as:

$$S(t) = P_0(t), t_0 \leq t < t_1$$
$$S(t) = P_1(t), t_1 \leq t < t_2$$
$$\vdots$$

$S(t) = P_{k-2}(t), t_{k-2} \leq t < t_{k-1}$

If all the piece polynomials have a degree of n at most, the spline degree is $\leq n$. In order to create this spline we use the function *cscvn* that returns a **natural or periodic interpolating cubic spline curve** which has degree 3 and continuity C^2.

t_k is what we define as control points or **break points**. Considering we have a cubic spline, the conditions that must be met in these break points are:

$S(a) = S(b)$
$S'(a) = S'(b)$
$S''(a) = S''(b)$

If the spline is also natural, we can add the following condition:

$S''(a) = S''(b) = 0$

Now we have a much simpler representation of our snake. We just need to apply the snake algorithm to N break points and create the rest of the contour using polynomials. This means that we will only need to check our energy conditions on N finite number of points, not on every point of the curve.

We can see this in figure 6.3 where we have N blue break points and the rest of the curve created using a natural cubic spline and represented in red.

After seeing the tools that we use to find our outer boundary, we move forward to explain how we need to apply the snakes and splines, the energy terms that we can use and the problems that we are going to face.

6.2 Starting points for the snake

One of the most important steps to see our segmentation algorithm succeed, is finding good starting points for the snake. If we were to start inside the EEL, the snake would not be able to move towards the outside and it would probably find the IEL instead. If we were to start too far outside the EEL, the snake could get "stuck" in the white artifacts that sometimes appear in the probability images. This means the starting

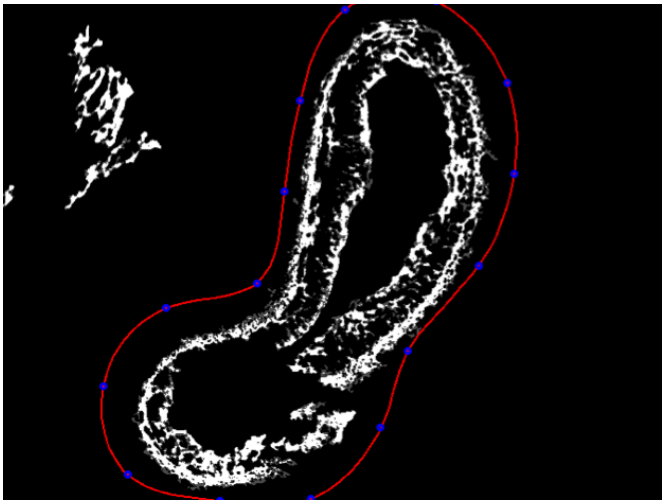


Figure 6.3: Blue break points and red spline

points are very important and they depend on the shape and size of the artery. In order to find different starting points for every image, we need to use some contour information of the artery, which we already have. The best approach is to use the inner boundary that we have found using region growing as explained in previous chapters. This technique is robust and gives us good results, therefore we can rely on its results to find the outer boundary.

What we want to do is shown in Figure 6.4. We start from the inner boundary (green) and dilate (orange arrows) the contour until we find that it is outside the EEL (red).

In order to do this, we apply Algorithm 3.

- Apply dilation to our inner boundary contour and find new contour C. We consider the cases in which the curve goes out of boundaries as in figure 6.4.

- Sample C with just N=15 points (same as the number of break points of our spline)

- Verify condition 1, that the value of the probability image at M=200 points equidistantly distributed along the spline is 0, which means that all of them are on a black pixel. We use this condition to make sure that C is located outside the artery, outside the region with white pixels.

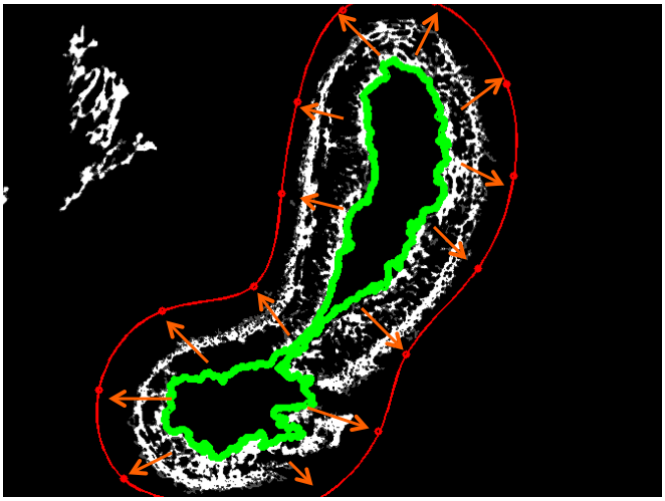


Figure 6.4: Finding the snake starting points

- Verify condition 2, that at least we have inside C 50% of the white pixels of the total probability image. We use this condition so C does not find itself on black background but still inside the artery.

Algorithm 3 Automatic calculation of the starting points of the snake

Require: Dilation parameter $DIL = 40$; A = array containing probability image

repeat

Apply dilation with value DIL on inner boundary contour. Obtain new boundary C.

Sample C. Output = P_i with $i = 1...N$.

if $\sum_j A(P_j) == 0$ with $i = 1..M$ AND # of white pixels inside $C > 0.5$ * # of white pixels of A **then**

End program. Current N points will be starting points for snake.

else

$DIL = DIL + 40$

end if

until Maximum 15 repetitions

Now that we have a very good algorithm to automatically find the starting points for the snake, we are going to study which energy terms should we use in order to drive the snake towards the EEL.

6.3 Energies that drive the snake

The calculation of the energy parameter is a key aspect for the correct use of snakes as the curve will be driven towards the points with high energy. Therefore we are going to explain in detail each and every one of the terms, why we have chosen them and finally the weights we will give to every one of them.

In order to better understand all the terms, we are going to include their mathematical expression. To understand them, we are going to define the variables first.

We calculate the energy for each break point, so only the current point P_i and its neighbors P_{i+1} and P_{i-1} affect the calculation of the energy of P_i. In order to make it easier to do the calculation in MatLab, we define two vectors, one for each coordinate, x and y, of the points.

$x(1), y(1)$ = components of P_{i-1}
$x(2), y(2)$ = components of P_i
$x(3), y(3)$ = components of P_{i+1}

Once the energy terms are calculated, we move on to the next point and so the vectors x and y change accordingly.

The average distance of the segment of a snake, which is the distance between two consecutive break points, is also an important value and we express it as d.

A is the array containing the probability image.

6.3.1 Closeness term

The first term is typically used in the calculation of snake energy and its mathematical expression appears in equation 6.1.

$$E_c = |x(2) - x(1)|^2 + |y(2) - y(1)|^2 \tag{6.1}$$

Minimizing this expression is equivalent to minimizing the first derivative. As we can see, this term is trying to reduce the length of each segment of the snake. We explained before that we want our snake to have this behavior, since the contour starts outside the artery and we need to drive it towards the inside until it finds the EEL.

This term can also be expressed as in equation 6.2.

$$E_c = |d - |x(2) - x(1)|^2 + |y(2) - y(1)|^2| \tag{6.2}$$

Minimizing this expression is more complete since it doesn't necessarily shrink the contour but rather keep the points equally distributed along the snake.

6.3.2 Smoothness term

The second term is also typically used in energy calculations and its mathematical expression appears in equation 6.3:

$$E_s = (x(1) - 2 * x(2) + x(3))^2 + (y(1) - 2 * y(2) + y(3))^2 \tag{6.3}$$

This expression is equivalent to minimizing the second derivative. What this term does is move the points also according to the positions of their neighbors, so no corners or oscillations are created. We are interested in having a smooth curve according to the shapes of the arteries, so this term will be of great importance. The smoother the curve is, the lower the smoothness term.

6.3.3 Probability term

The third term is created depending on the images on which we are applying the snakes, the probability images. We are interested in driving the snake towards the white pixels, so we will give this probability term the maximum intensity value found in a 5x5 window centered on the point we are analyzing, as described in equation 6.4.

$$E_p = max(A(x(2) + w, y(2) + w)), w = -2..2 \tag{6.4}$$

We are interested in maximizing this term, so the points with higher energy are the whiter ones.

6.3.4 Spline term

The fourth term is based on the same idea as the third one, but instead of looking at the highest value in a small window, we look at some values along the spline segment we are analyzing. We sample the segment from P_i to P_{i-1} with 30 points, and do the same for the segment between P_i to P_{i+1}. After this, we find the average intensity

value of all these sample pixels along the spline. This solves the situation when we have a break point on an isolated white artifact. In that case, the probability term is maximized as the break points is on a white pixel. On the contrary, the spline term is minimum as most of the points on the spline are on black background. This new term helps the break point decide to jump from that white artifact to look for another region with higher energy.

6.4 The snake algorithm

Now that we have seen all the energy terms that are going to drive our snake, we can explain the exact algorithm that we use to automatically detect the EEL.

We analyze, for every break point of the spline, the new possible positions where these points can move according to the energy terms described before. As the starting position of the snake is outside the EEL, we are interested in driving the break points in the direction of the inner boundary. Therefore, we only move the break points in the direction formed by the break point and the point of the inner boundary at minimum distance, as shown in figure 6.5.

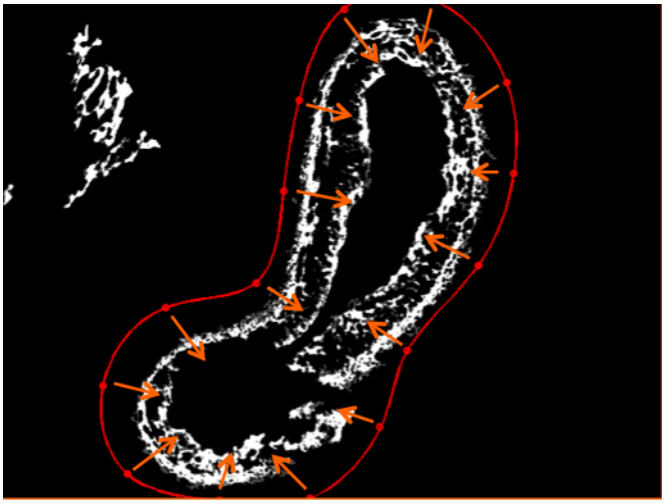


Figure 6.5: Search direction for break point candidates

In this direction, we look at the break point candidates within a window. If any of the candidates fulfills the energy conditions, we choose it as the new break point. If

not, we increase the window size to include more candidates.

The conditions for the new break point are:

- The energy of the candidate must be significantly higher than the energy of the previous break point. This way, we avoid unnecessary jumps and we also make it difficult for the curve to go towards the inside of the EEL.
- The current break point's two neighbors have moved significantly and that has left the break point isolated, creating a high curvature in the spline.

We repeat the process for every point and we only stop when we are looking for candidates beyond the inner boundary.

The process is detailed in Algorithm 4.

In this algorithm we have another parameter that we can vary to get better results. After many experiments, it was determined that the value that gave us the best and the most stable results was K=1.8.

As we can see, the energy of each break point is recalculated on every iteration to include the possible movements of its neighbors. Also after each iteration, the spline is recalculated using the new break points. This is a complete algorithm that was updated as the research went on. More conditions were added in order to avoid the problems that appeared in every experiment.

Another important aspect of the algorithm is the choice of the weights for every energy term. We also experimented with these and finally found the weights that gave us the best results.

6.4.1 Weights of the energy terms

After experimenting with the energy weights, we came to the conclusion that there is an energy term that we do not need in our algorithm: the closeness term. This is a very useful term when the window of search is fixed and therefore the algorithm cannot reach all the pixels. This means that some pixels are never analyzed as candidates and therefore we need a term to push the snake towards the inside, where the "good pixels" are supposed to be.

Algorithm 4 Snake algorithm

Require: $windowsize(j) = 11\ \forall j = 1..N$, N = number of break points

repeat
for $j = 1..N$ **do**
if We are looking for candidates beyond the inner boundary **then**
$finish(j) == 1$
end if
for All the pixels in the search window $(P_j + 1..windowsize(j))$ **do**
Calculate the total energy
end for
Normalize the energy by the maximum
Find point in window with maximum energy, max(E). If there are several, pick the one nearest to the P_j.
Calculate the curvature on each P_j, $curv(j)$
if $(max(E)$ >K*previous energy of $P_j)$ OR $(finish(j) == 1$ AND $curv(j) >$ 0.03) **then**
Choose this point with max(E) as new P_j
else
Increase $windowsize(j)$
Recalculate energy of P_j in case its neighbors have moved
end if
end for
until $finish(j) == 1\ \forall j$

In our case, we prefer to use a growing window that is different for each break point because the closeness term gave us some problems, like driving the snake towards the IEL instead of staying on the EEL.

As for the rest of the terms, we clearly see that the probability and the spline term are equally important.

We show the final expression for the calculation of the energy in equation 6.5:

$$E = 0 * E_c + 8 * E_s + 10 * E_p + 10 * E_{spl} \tag{6.5}$$

- Closeness term = 0
- Smoothness term = 8
- Probability term = 10
- Spline term = 10

Now that we have defined all the methods used, it is time to explain why we created a Graphical User Interface (GUI), how it minimizes the user interaction and makes the whole process more automatic.

Chapter 7

Graphical User Interface (GUI)

After explaining in detail algorithms and ideas, and presenting several equations and images, it would be time to present the results of all this work. But before that, we want to give importance to the more visual side of the project, the creation of the GUI. Although sometimes it is not considered relevant or even part of a project, we think that it deserves a whole chapter. In the next pages we will present the reasons to create a GUI as part of the project, its menus and buttons, how we organized the files and finally a brief guide for creating a new project, so we can see the advantages of having this GUI.

7.1 Why do we need a GUI?

After working for some months on the project, we had a big number of functions, many images to test, and we had to find the best parameters (erosion, region growing threshold, energy terms of the snake, etc) to obtain the best results. Testing all the images and all the parameters was a tedious job and it was difficult to keep the results organized. That is why we decided to put all our work together by creating a GUI. Organizing the functions and the resulting images would be very helpful to analyze the final results of the project.

The main goals for the creation of this GUI are:

- Make the whole segmentation process more automatic

- Allow the user to make small changes to the parameters so the algorithm has more flexibility and can work with a wider variety of images.

What we want is to create another tool to help the user solve the segmentation problem with fewer steps.

7.2 Menus & buttons

The GUI has been programmed so the user can do all the actions on the images in an easy way. The organization of the interface is very important, as the GUI has to be easy to manage and it has to have access to all the functions of the program through the menus and buttons.

In figure 7.1 we present a general view of the interface.

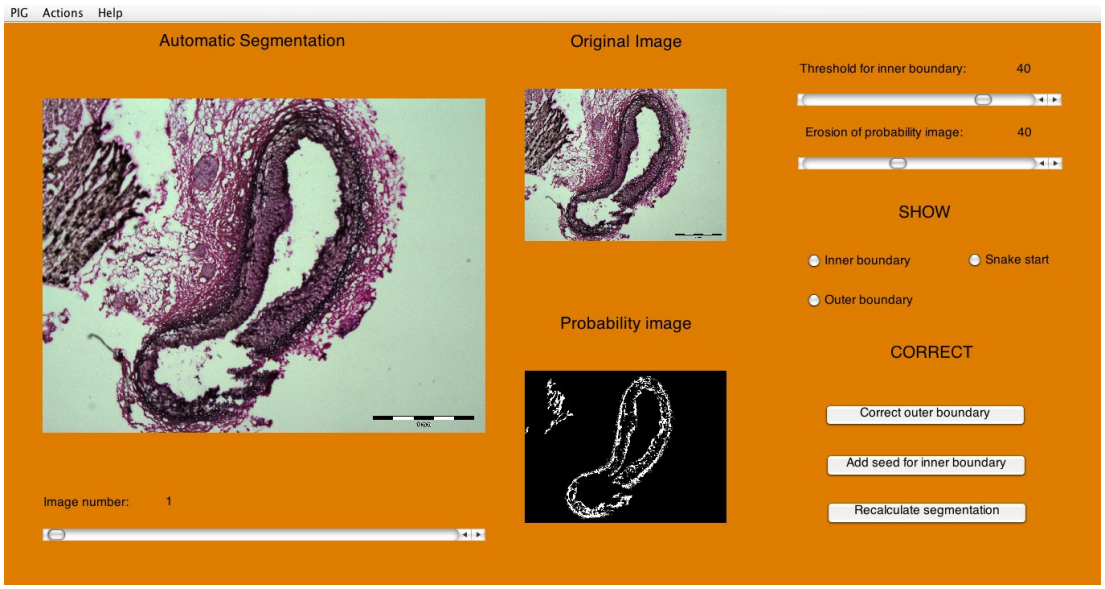


Figure 7.1: Graphical User Interface

As we can see, the GUI shows the user three images. The first bigger one shows the original image. On it, the user can choose to see the outer boundary, the inner

boundary and the start position of the snake. The other two smaller images show always the original image and the probability image.

Below the big image we can see a bar that allows us to go from one image to the other if the project contains more than one image to analyze. There is no limit to how many images a project can contain.

On the right we see separate regions of buttons. The first one contains bars to choose parameters like the erosion value used to clean the image or the threshold used by region growing. The erosion value can be changed in steps of 5 and the threshold value in steps of 10. The maximum value for the erosion is 100 while the threshold's is 50. These maximum values are decided based on our experiments. The second region, named "SHOW", contains the three buttons that allow the user to see the inner boundary, the outer boundary and the snake starting position in the bigger image, as described before. In figure 7.2 we have activated all the buttons and we can see the inner boundary in green, the outer in blue (its break points in yellow) and the snake starting position in red.

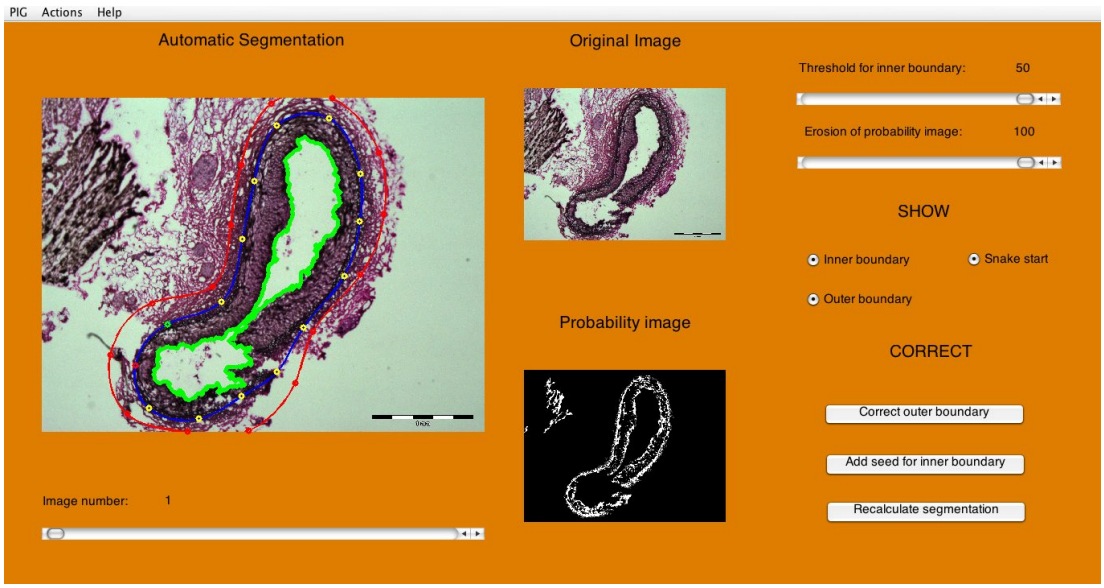


Figure 7.2: "SHOW" options activated

Finally, we have the region called "CORRECT". It contains all the buttons that

the user needs to correct the results once the automatic segmentation is done. The user can correct the outer boundary manually, add a seed for the inner boundary calculation or recalculate the segmentation once changes in the parameters have been made. For example, if we change the parameters from P1 to P2 with the bars but the Recalculate Segmentation button is not clicked, the P1 parameters will still be the valid ones. If we then go to another image and come back, the parameter bar will show the valid parameters P1. The parameter bar always shows the parameters used for the current segmentation.

At the top of the GUI we can see the menus of the interface. An scheme of the menus is showed in 7.3.

Figure 7.3: GUI menus

In the first tab named PIG, we find the basic actions to start working with the GUI: create a new project or open an existing one. This means that we can be working on a project, close the program and continue working on it the next day with all our results saved.

In the second tab, Actions, we have:

- "Choose seeds for the inner boundary segmentation", which shows the user all the images one by one on the big image frame and lets the user click one point on each image which the program stores as the seed point for the region growing algorithm. This way, the least automatic part of the algorithm, which is clicking all the seed points, is made a lot easier.

- "Segment all". All the images are segmented automatically. It finds the boundaries, stores the results and when the process is finished the user can see the results in the GUI.

There is a third tab where we can find a User's Guide for the program.

Now that we are familiar with the interface, we are going to present how the files are organized and where the images and results are stored.

7.3 File organization

To start using the GUI, we first recommend to copy all the files to a folder with the name "Working Place". Once this folder is chosen as the current directory in MatLab, we can create new projects and all the information is stored in that folder. When we create a new project, the GUI asks for the name that we want to use for that project. It then creates a folder in the Working Place with the name we have just provided.

Once inside this folder we see 3 folders and 3 files:

- erosions.txt = file that stores in a column the erosion parameters used for the segmentation of each image

- Original images = folder that stores all the original images (.tif) of the project

- Probability images = folder that stores all the probability images (.mat) once the automatic segmentation has been done at least once

- seeds.mat = file that stores the position of the seeds for each image for the region growing algorithm

- thresholds.txt = file that stores in a column the threshold parameters used to find the inner boundary

- Variables = folder that stores all the variables (.mat) that contain the information of the inner boundary, outer boundary and snake positions

7.4 Steps to create a new project

We are going to define briefly the steps to create a new project, so we can see how the GUI is used.

1. Open MatLab, choose Working Place as the current directory and type *pig_project*.

2. Go to the menu and under the tab named PIG, choose *New Project*.

3. Introduce a name for the project in the message box that has just appeared.

4. Choose the folder where you have the images to analyze.

5. Now the first image of the project appears in the bigger frame.

6. Go to the menu and under the tab Actions, *Choose seeds for inner boundary segmentation*.

7. A cursor appears on the bigger image. Click where you want to place your seed (inside the lumen). The second image of the project appears. Choose the seeds for all the images.

8. Go to the menu and under the tab Actions, choose *Segment all*.

9. Now you have all the images automatically segmented. It is time to see the results. You can see all the images of the project by using the bar under the bigger frame. You can see the results of the inner boundary, the outer boundary and the snake starting point by clicking at the buttons on the right.

Now that we have all of the automatic results, we present how to make changes in case some results are not accurate enough.

10. If the outer boundary has a few incorrect points, you can correct them manually. Click on *Correct outer boundary*. A cursor appears as shown in figure 7.4 .

If you want to delete a point, place the cursor next to it and press "D".

If you want to add a point, place the cursor next to the place where you want to add it and press "A". An example is shown in figure 7.5.

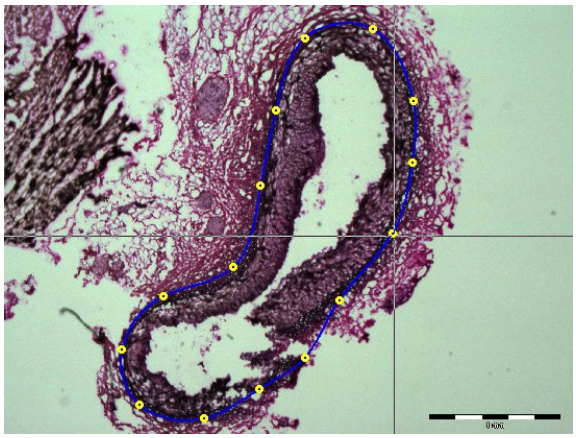


Figure 7.4: Example: Cursor appears

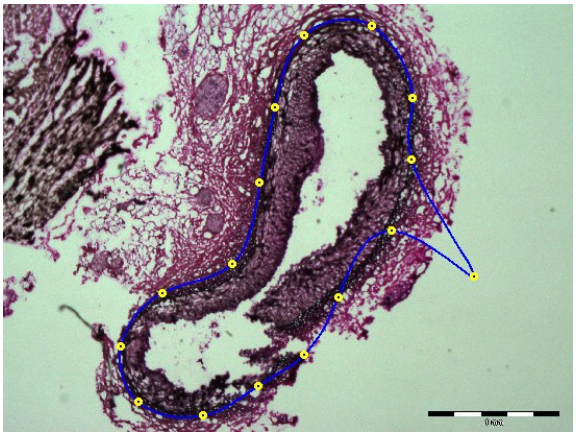


Figure 7.5: Example: Point added

If you want to change one existing point with a new one, place the cursor where you want the new point to be and press "C". The program erases the nearest point to the cursor location.

11. If the inner boundary is incorrect, because the lumen is divided in unconnected regions, click on *Add seed for inner boundary*. A cursor appears, place it where you want to add another seed for the region growing algorithm. Click on *Recalculate segmentation* to obtain new results.

12. If the inner boundary contains parts outside the lumen or does not contain all the lumen region, change the *Threshold for inner boundary* value. Click on *Recalculate segmentation* to obtain new results.

13. If the probability image does not show all the artery contour or it has too many undesired white artifacts, change the *Erosion of probability image* value. Click on *Recalculate segmentation* to obtain new results.

Now that we have seen all the aspects of the work done in this project, it is time to see the results.

Chapter 8

Results

In this chapter we will explain the final results that we obtained and we will present a method to measure the statistics of these results to see if our algorithm gets us closer to the automatic segmentation. We will also analyze the problems that our algorithm presents and how we can solve a few of those by just making few adjustments.

8.1 How to measure the results?

In order to be able to find out if the results of this project are good, we have to decide how to measure the statistics of these results. To quantify the errors that our automatic algorithm makes, we segmented the images manually beforehand. We consider this manual segmentation to be the "ground truth" and therefore all the errors are calculated comparing the automatic result and the manual one.

As we are interested in finding the ROI of the artery, we consider appropriate to measure the error based on areas. We define area as the number of pixels contained inside a boundary. This measure can be applied to the region inside the inner boundary, inside the outer boundary or between those two. The expression of the error measure that we use is presented in equation 8.1, considering N_a as the number of pixels of the area found automatically and N_c the number of pixels of the correctly segmented area (the one segmented manually).

$$Error = |N_a - N_c|/N_c * 100 \tag{8.1}$$

Now we are going to present the results with images and statistics.

8.2 Results of the automatic segmentation

In this section we present all the results obtained automatically, with the parameters described in previous sections: 0-8-10-10 as the weights of the energy terms, 50 as the erosion parameter, 40 as the threshold for the region growing and 1.8 as the parameter of the snake algorithm.

In our case, we show all the error measures relative only to the outer boundaries. We count the area inside the correct outer boundary and compare to the area inside the automatic outer boundary, without considering the inner boundary. This is because the errors on the calculation of the inner boundary affect also the outer boundary, so we do not need to quantify them for now.

We start by presenting a very bad result shown in figure 8.1.

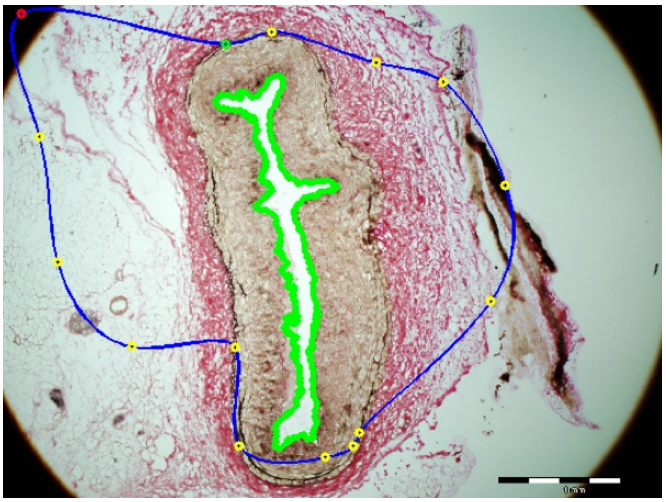


Figure 8.1: Automatic result classified as bad

We clearly see that some of the points are not able to move towards the EEL, because they are stuck in dark pixels outside the actual boundary.

This image has an error measure of 153.2%. The error is obviously higher than 100 because the incorrect area is much larger than the correct one.

The second image we want to present is shown in figure 8.2. It is a result that is not visually too bad, but it gives us a lot of error when we compare areas. Given the dark color of the adventitia of this artery, the algorithm detects this region as possible

boundary and stops the snake before it reaches the EEL. We have to mention that the general position of the artery has been indeed detected by the algorithm.

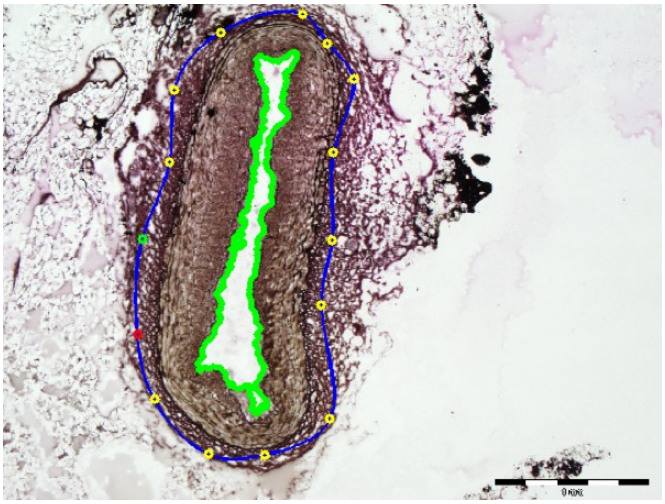


Figure 8.2: Automatic result classified as medium

This image has an error measure of 21.9%. We cannot consider this result to be correct, but the error shows that we are not too far from the correct segmentation.

The third image, figure 8.3, shows the perfect adaptation of some parts of the curve to the EEL but some errors caused by the same problem as the previous image.

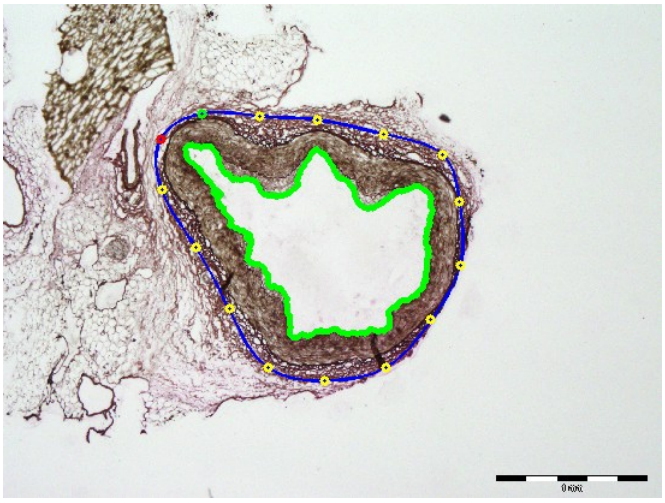


Figure 8.3: Automatic result classified as good

This image has an error measure of 17.3%.The difference from the previous image

is that we can consider this result to be good.

The fourth image, figure 8.4, shows a very good behavior of our algorithm but fails to adapt the curve perfectly in some points due to the high curvature of the shape of the artery in these points. In this case the problem of the upper right region of the image can be fixed by adding one point manually.

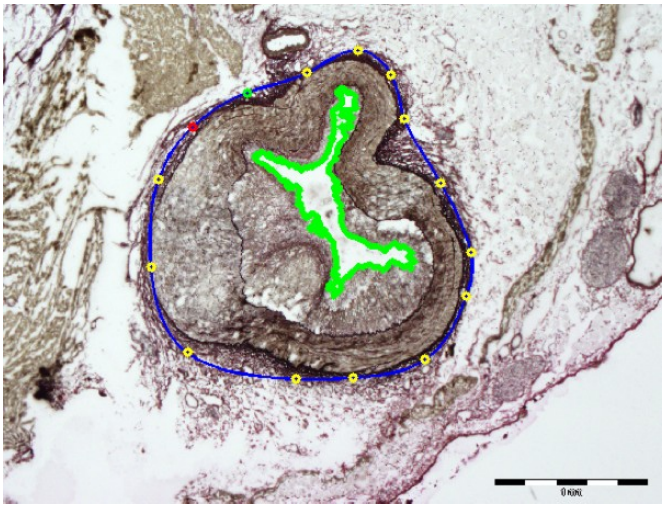


Figure 8.4: Automatic result classified as good

This image has an error measure of 9.5%. It is a very low error value which reflects what we see in the image, that we just have a few points which are not correct, but generally the result is pretty good. We only need to make few small adjustments.

Finally, figures 8.5 and 8.6 show how the algorithm finds both the inner and the outer boundary perfectly well.

These images have error measures of 6.9% and 2.3%, respectively. When we find these error values we can definitely consider that the automatic segmentation is visually perfect. As we can see in figure 8.6, inside the artery there are some calcifications that could affect our segmentation. For now we have been able to avoid this problem.

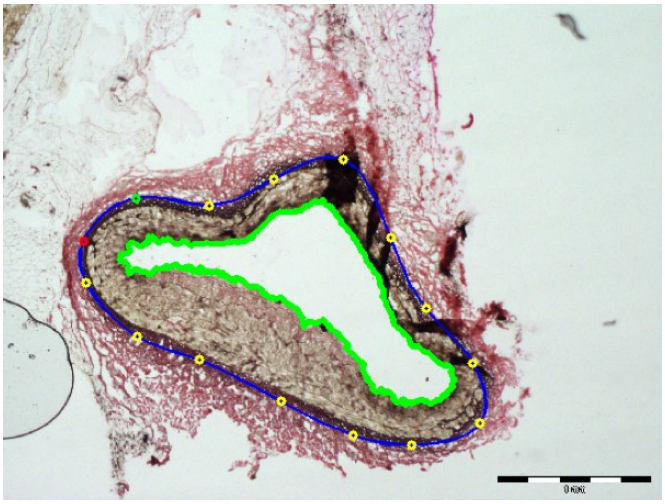


Figure 8.5: Automatic result classified as very good

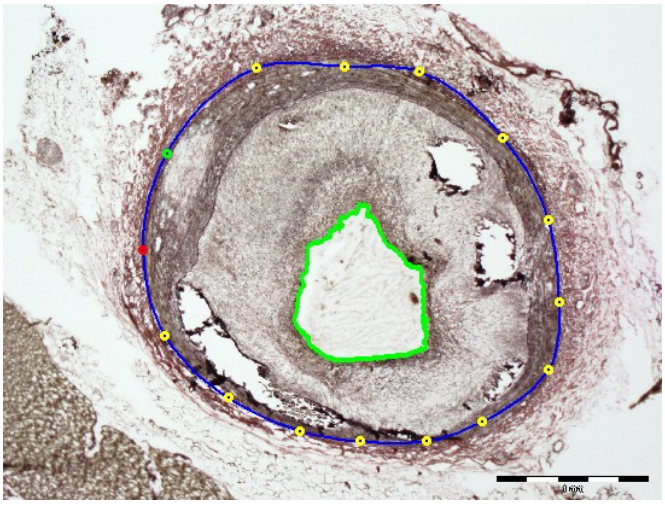


Figure 8.6: Automatic result classified as very good

Now that we know how to interpret the error measures, we can see the statistics of the results for all the 42 analyzed images. In figure 8.7 we plot the number of images against the percentage errors of the outer boundaries.

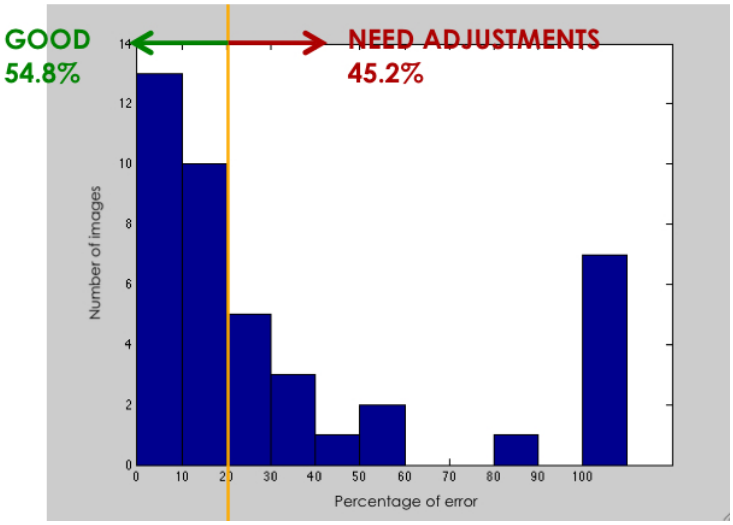


Figure 8.7: Statistics of the error measures

As we marked in the plot itself, and based on what we have seen before, we consider all the images with an error measure of 20% or less to be correct (54.8% of the images).

If we analyze this further as we present in figure 8.8, we can say that 31% of the images have an error of 10% or less and therefore are visually well-segmented. Another 23.8% have an error between 10% and 20% and we consider them well-segmented except a few points which are slightly misplaced. The 45.2% of the images have incorrect segmentations and need many adjustments. Of these 45.2%, a 23.8% has an incorrectly detected inner boundary.

We can make small manual adjustments to the 54.8% of good images to reduce their error measures, but these automatic segmentations are useful to make the calculations of lipids and other characteristics of the artery.

Now let's focus on the 45.2% of incorrect images and look at the problems that our algorithm has.

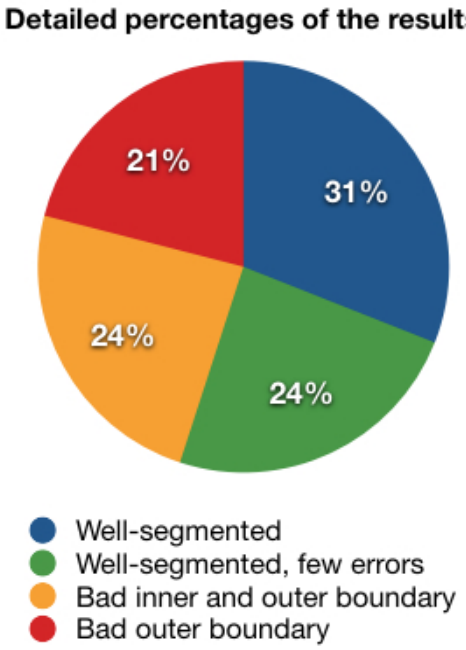


Figure 8.8: Statistics of the error measures

8.3 Analysis of the problems

Studying the specific problems of our algorithm is specially important both to see if there is an error that appears repeatedly and to find the type of errors that can be reduced with small adjustments.

A quite common problem that affects 21.4% of our images is that the lumen is divided in several separate regions and so the region growing algorithm is only able to find the closed region that contains the seed. An example of this problem is shown in figure 8.9 where we see that the artery's lumen is divided into 3 regions and therefore we need 3 separate seeds for the region growing algorithm to work properly.

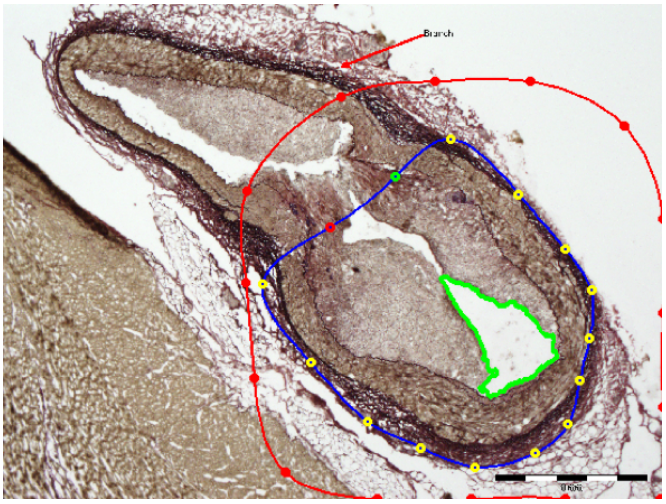


Figure 8.9: Problem 1: Lumen divided

If the inner boundary is not well-segmented, the starting points for the snake are also likely to be incorrect and so the curve can never reach the whole EEL. We can see in figure 8.9 that the lower part of the snake has, in fact, adapted to the boundary of the artery.

Another problem, which we commented previously, is that the probability image is not correct. Either it has too many white artifacts or there are not enough white pixels to correctly draw the artery contour. This is sometimes due to the wrong value of the erosion parameter. A wrong probability image, as shown in figure 8.10, affects all the process and we have results like the one shown in figure 8.11.

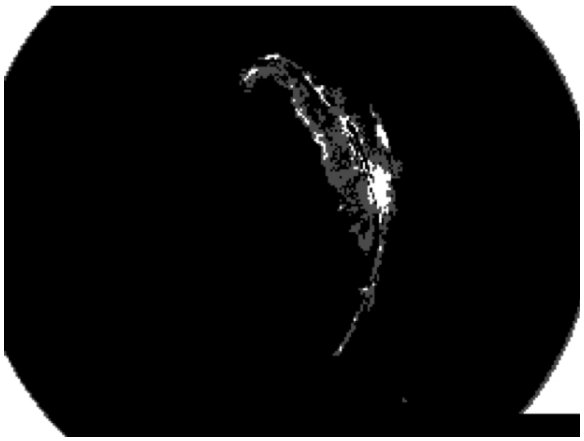


Figure 8.10: Problem 2: Wrong probability image

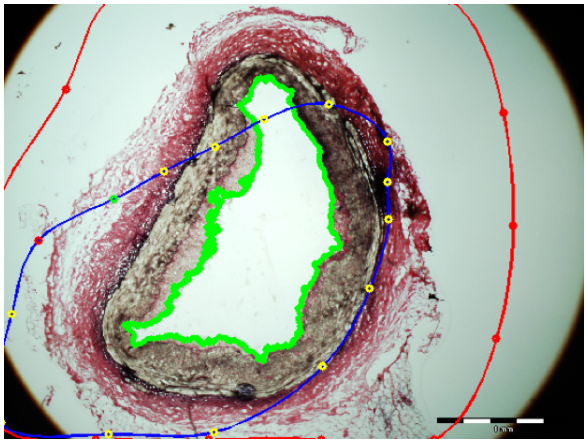


Figure 8.11: Problem 2: Wrong segmentation

As we can see, the snake cannot adapt to the EEL in places where there are no white pixels in the probability image. We will need to look at the erosion parameters to see how many images can be fixed.

The third problem affects only 1 image, shown in figure 8.12. It is the incorrect detection of the inner boundary due to the incorrect value of the threshold. Having an incorrect inner boundary also affects the outer boundary.

There is also a small problem that we have in 4 of our images, one being figure 8.12. In the corners of these images we can see the lens of the microscope. This should not appear in the photograph and it creates a big problem for the algorithm, because the corners are dark, just like the EEL, and in the probability image they appear as white. We will not discuss this problem any further, as we talked to the professionals that took the pictures and they say it usually does not happen, so the algorithm does not need to take it into account.

The last and most difficult problem to solve is the groups of dark pixels that are formed outside the artery adventitia and that are quite common in the process of staining. We can see an example in figure 8.13, where there is a mass of stain on the left side of the image. Our algorithm detects it as boundary pixels and in the

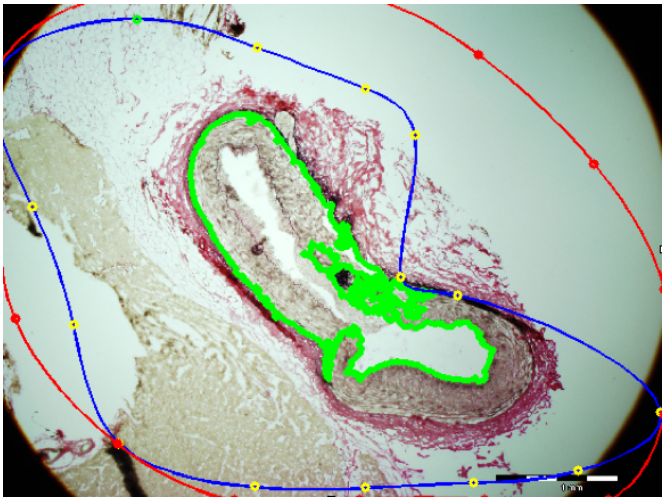


Figure 8.12: Problem 3: Lumen incorrectly detected

probability image they appear as white, as we can see in figure 8.14. The snake has to start after these masses and so the points are stuck in the white pixels and cannot move towards the real EEL.

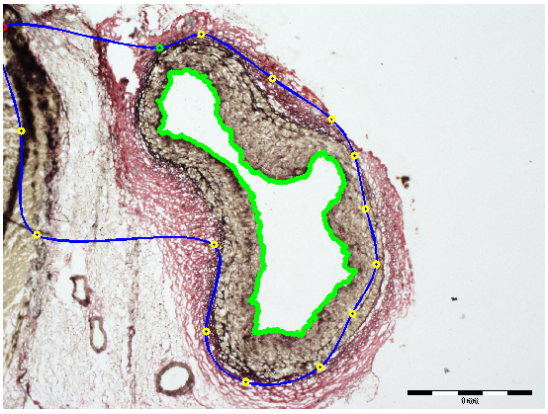


Figure 8.13: Problem 4: Staining problem

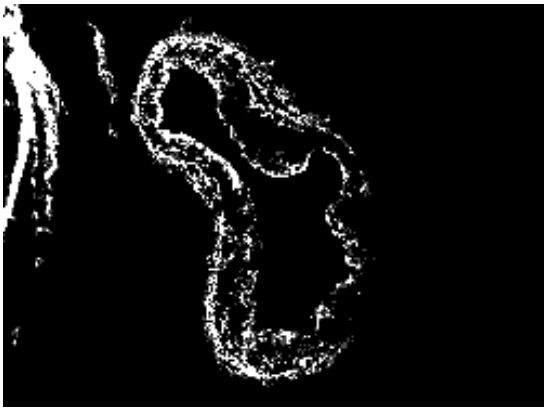


Figure 8.14: Problem 4: Staining problem

Now that we have presented the main problems of our algorithm, it is time to see which of these problems can be easily solved.

8.4 Images we can fix with small adjustments

First of all, we have to say that out of the 10 images that had incorrect inner boundaries, 9 can be fixed by adding seeds and 1 can be fixed by changing the threshold value. Both are fairly easy to do with our GUI, therefore, with small adjustments, we have 100% of the inner boundaries well-segmented.

Now we can focus on our biggest problem: the detection of the outer boundary. Just by adding more seeds we can improve the segmentation of 2 images. By changing the erosion parameter, 3 segmentations are improved.

So let's see what are the error measures with these new small adjustments.

8.5 Final results

If we compute the statistics of the error measure again, we obtain the plot that appears in figure 8.15.

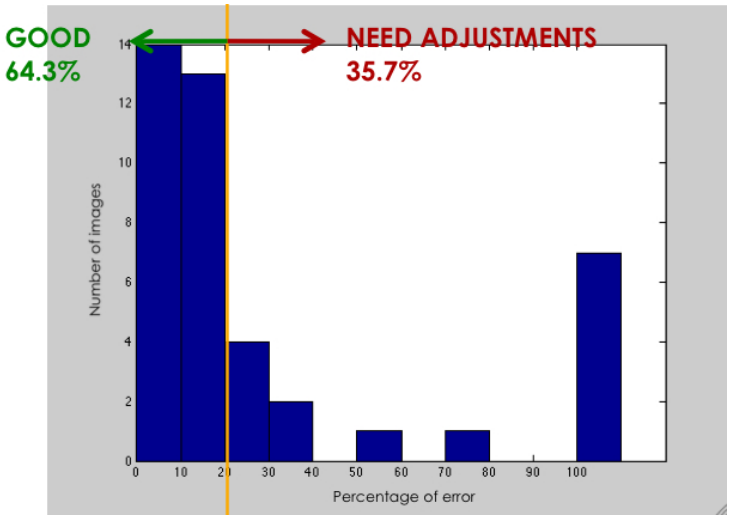


Figure 8.15: Final statistics of the error measures

In conclusion, we have obtained statistically **54.8% of fully-automatically segmented outer boundaries**. If we now include a little user interaction, with our GUI it just requires 1-2 clicks to improve the results to **64.3% well-segmented images**.

The 35.7% of incorrectly segmented images can be also easily manually corrected using our GUI, requiring less than 10 clicks for each correction. A more visual presen-

tation of these results is shown in figure 8.16. Note that we have added to the results the initial click for the region growing algorithm.

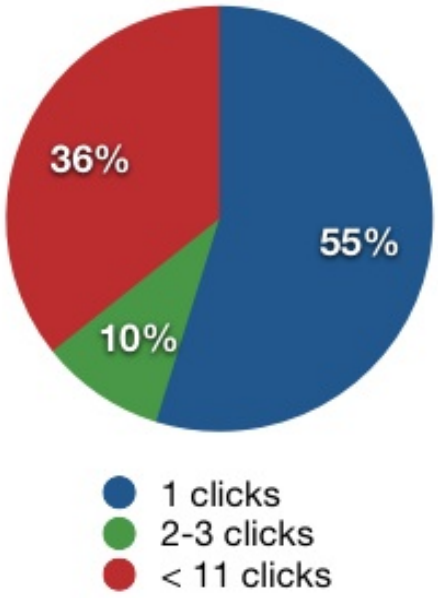


Figure 8.16: Number of clicks needed to segment the images

Chapter 9

Future work

Although the method is still not automatic, we definitely see this project as a good starting point. Writing about the characteristics of possible future work is as important as writing the algorithm itself. Now that we have been working for some months on this project, we know perfectly well the problems that we face, how we can improve the algorithms and the new ideas that are more likely to work. It is important to pass all this information so another person can continue our work.

9.1 Texture information

If we take a look at the problems that we cannot solve yet with our algorithm, we can see the two big issues are: the points stuck outside the adventitia and the points stuck far from the artery in stain masses.

So, what if we left color information aside for a moment and focused on texture information? We started to work on texture but since we could not finish the study due to time constrains, we decided to include the results in the chapter of future work. There are several techniques to analyze texture (20) and it has been applied many times on medical images (21; 22; 23; 24).

The adventitia has a texture that is indeed very different from the media as shown in figure 9.1.

The media has a more uniform texture, without the white spots that clearly appear in the adventitia.

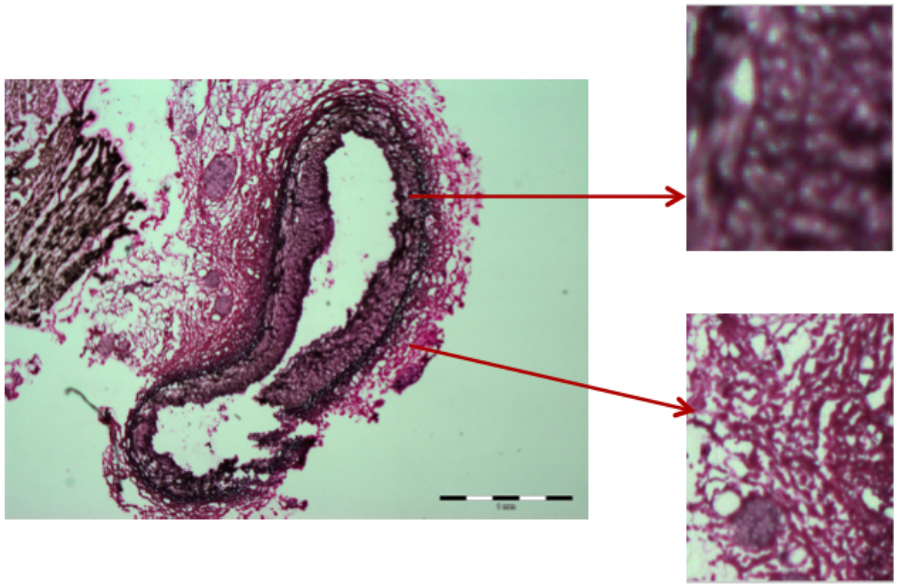


Figure 9.1: Difference in texture: adventitia vs. media

A good measure of texture is the **Gray-level co-occurrence matrix or GLCM**. This matrix shows the distribution of co-occurring values at a given offset (25).

An example is shown in figure 9.2.

The entry (i,j) of the matrix corresponds to the number of occurrences of the pair of gray level values (i,j) which are a d distance apart. For every distance d, one co-occurrence matrix is created. In our case we have 256 different level values, so our matrices have a dimension of 256x256.

The mathematical expression for the calculation of the matrix elements is shown in equation 9.1.

$$C(i,j) = \sum_{p=1}^{n} \sum_{q=1}^{m} \begin{cases} 1 & \text{if } I(p,q) = i \text{ and } I(p+\Delta\, x, q+\Delta\, y) = j \\ 0 & \text{otherwise} \end{cases} \tag{9.1}$$

Once we have calculated the matrix for the original image, we can calculate 4 different properties to see the differences between the media and adventitia:

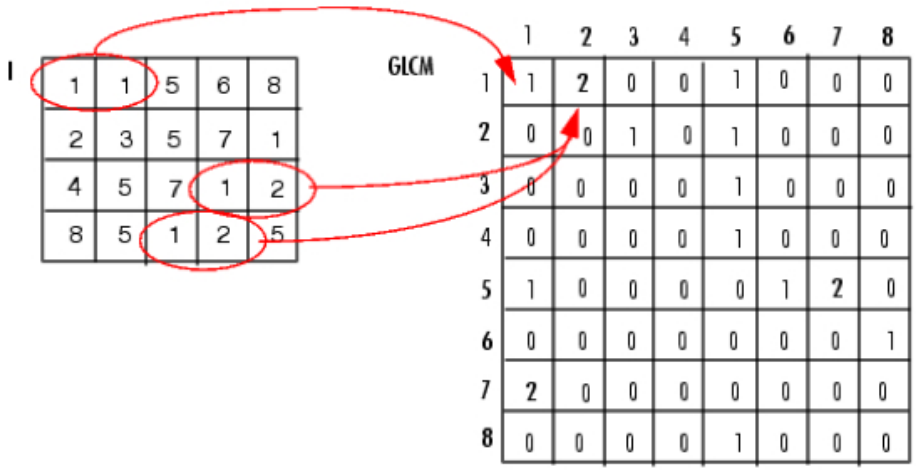


Figure 9.2: Gray-level co-occurrence matrix

- **Contrast**: returns a measure of the intensity contrast between a pixel and its neighbor over the whole image. Its mathematical expression appears in equation 9.2.

$$CON = \sum_{i,j} |i-j|^2 * A(i,j) \tag{9.2}$$

- **Correlation**: returns a measure of how correlated a pixel is to its neighbor over the whole image. Its mathematical expression appears in equation 9.3.

$$COR = \sum_{i,j} \frac{(i-\mu_i) * (j-\mu_j) * A(i,j)}{\sigma_i * \sigma_j} \tag{9.3}$$

- **Energy**: returns the sum of squared elements in the GLCM. Its mathematical expression appears in equation 9.5.

$$EN = \sum_{i,j} A(i,j)^2 \tag{9.4}$$

- **Homogeneity**: returns a value that measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal. Its mathematical expression appears in equation 9.5.

$$EN = \sum_{i,j} \frac{A(i,j)}{1+|i-j|} \tag{9.5}$$

We create the matrix with the MatLab function *graycomatrix* and calculate the properties with the function *graycoprops*.

We calculate the properties of blocks of 9x9 and we advance in steps of 3. The resulting values of the 4 properties are combined as described in equation 9.6 to obtain a unique value.

$$IM(i,j) = CON(i,j) * COR(i,j) * imcomplement(EN(i,j)) * imcomplement(H(i,j)) \tag{9.6}$$

We create a new image by assigning this IM value to the region of 3x3. As it can be seen, these new images have less resolution. This is done to reduce the computational time of the algorithm, which is unviable if we compute the properties in blocks of 9x9 but advance in steps of 1. We also normalize the new image by the maximum IM value, in order to create an image similar to the probability image created with color information.

After experimenting and comparing the result using these properties, we conclude that the best distances to see the texture differences between the adventitia and the media are: **d=[-2,2] and d=[-3,3]**.

With these distances, we obtain results such as the ones shown in figures 9.3, 9.4 and 9.5.

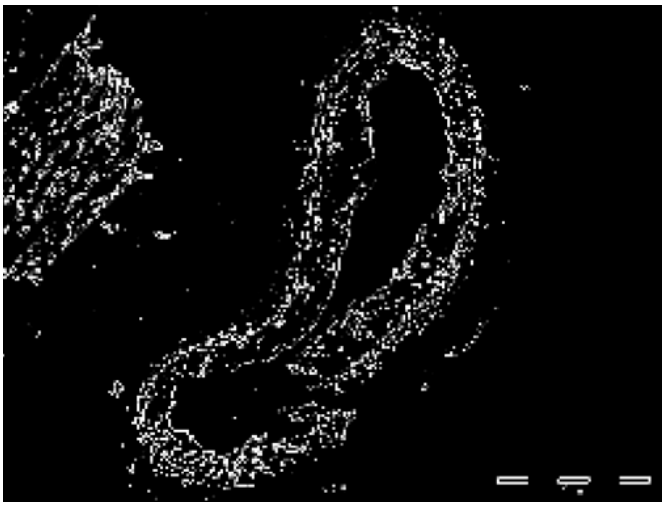


Figure 9.3: Image using texture

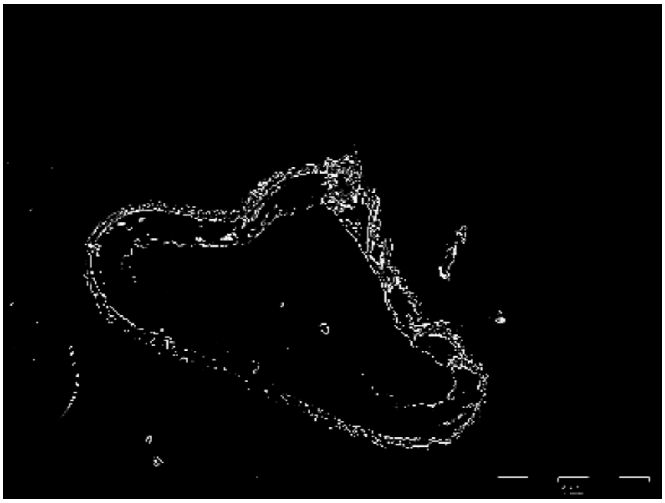


Figure 9.4: Image using texture 2

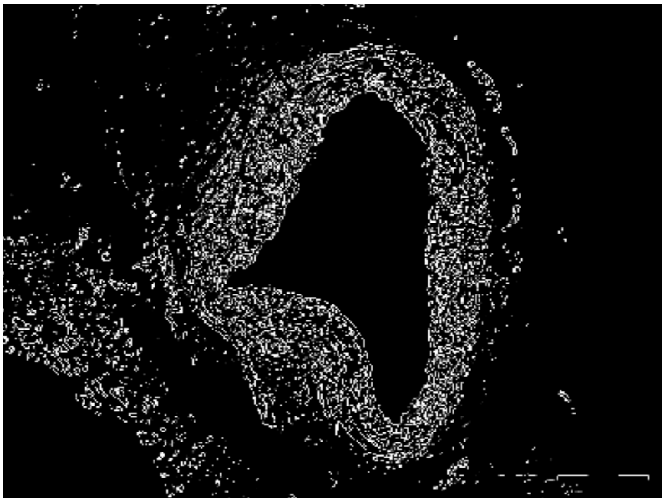


Figure 9.5: Image using texture 3

It is obvious that the contours can be seen in these new images, although with a little bit less resolution than our probability images. In the case of figure 9.5, the result is clearly bad because we have too many white artifacts.

Though it is not a perfected method yet, we see a bright future in the use of texture information. This is just a small study of texture using one technique which is GLCM, but there can be other techniques to improve these results. Texture can, in fact, differentiate the artery from the background, so I definitely think that it is a good next step to take.

9.2 Segmenting the Oil Red O images

The second objective of the analysis of these images was, once the ROI is found, using it on the Oil Red O stained images to find the amount of lipids found in the IEL, and that can cause the disease called atherosclerosis.

As we mentioned, it is much easier to find the ROI in the Verhoeff images because the stain brings out the IEL and EEL. Once we have the ROI in these images, we can translate it to the Oil Red O images to perform the second task of lipid counting.

In order to do that, we would need to apply a **Projective transformation** which is a combination of **translation + rotation + scaling + perspective warp**.

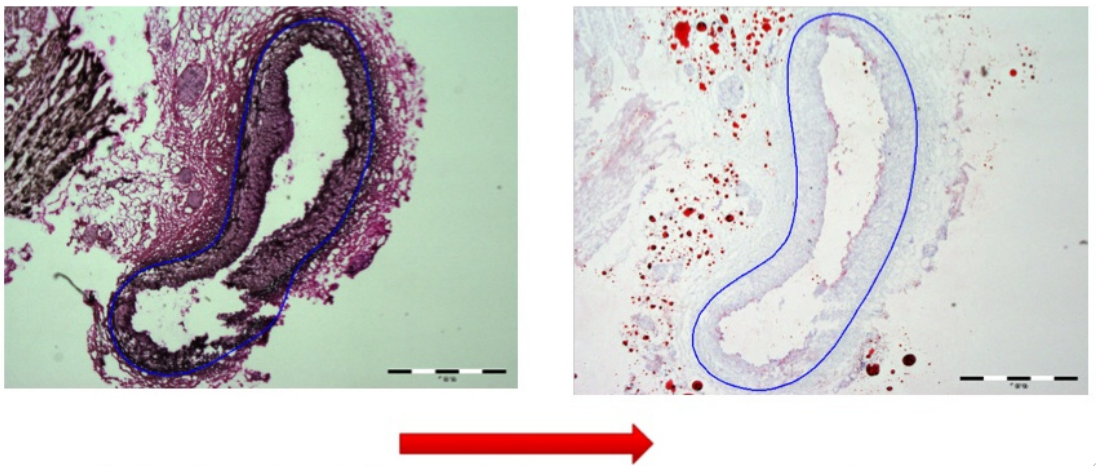


Figure 9.6: Projective transformation

Chapter 10

Conclusions

We started this project with the idea of getting closer to the automatic segmentation of the microscopic histology images of stained arteries. After seeing the different types of stain, we decided to work with the Verhoeff images, since this stain has the function of showing the IEL and EEL, exactly what we needed to detect.

We started with the work of George Masganas, who found a very good algorithm to segment the lumen of the images by finding the IEL but did not develop an automatic tool to segment the EEL. He worked on the livewire method and with that he was able to detect the outer boundary with about 10-20 clicks. This was still a highly user-dependent method, so we focused our work on developing a more automatic method for the EEL, while keeping the region growing for the IEL.

We started by increasing the size of our data set from 12 images to 42 images. This way we could have a broader view of the characteristics of the images. We soon realized that, due to the staining and freezing process, **the arteries had different shapes and sizes**. Therefore, we could not apply shape-based segmentation algorithms.

After looking at the images, we decided to start working with **color information**. The EEL had clearly a darker color than the rest of the artery, so we started by **modeling this color** and **creating a probability image**. In this image we obtained clear contours of the arteries in white pixels, while the rest of the background was black. The second step was to find the outer contour on the probability images, given the problems that they presented. We decided to use **snakes or active contours** as the tool to find these EEL. **Splines** were also used to simplify the snake representation.

After developing and correcting the algorithm, we created a GUI to simplify the work for the user.

In the end, we have obtained a **100% of well-segmented inner boundaries** by using region growing. The user still has to participate in the beginning of the segmentation process by choosing the initial seeds. Some images require more than one seed.

If we now take a look at the most challenging part of our project, the segmentation of the outer boundary, we see a big improvement in comparison with the livewire algorithm. The results shown earlier show that more than half of the images can be segmented in a fully-automatic way, we can add another 10% if we use a maximum of 2 clicks. Now we are only left with a third of the images that need a more manual segmentation. Even for these, with the help of our GUI, the user only needs 10 clicks or less to segment them.

In conclusion, we started from a method that required 10-20 clicks to segment the outer boundary plus the initial click to segment the inner boundary. That's an average of 16 clicks per image. By using our algorithm, we have reduced that to an **average of 4.76 clicks** per image in the worst cases. This shows an incredible improvement and reduction of the work that the user has to do to segment these images.

Although the method is still not automatic, we definitely see this project as a good starting point. If we take a look at the problems that we cannot solve yet with our algorithm, we can see the two big issues are: the points stuck outside the adventitia and the points stuck far from the artery in stain masses. In order to solve these problems, we tried using the the texture information of the image. We could not complete the work due to time constraints, but we definitely think that it is a good approach to be continued in the future.

After finding the ROI in the Verhoeff images, the next step is to transfer these results to the Oil Red O images and then quantify the amount of lipids inside this ROI, which is the ultimate goal of the segmentation.

In conclusion, we started our project with the goal of automatically segmenting the IEL and EEL of an artery. This task has proved complicated due mainly to the

variations that can occur during staining and freezing. Every image and every artery is different, therefore we need an adaptive method. We have been able to segment all the inner boundaries without a problem and almost 65% of the images with minimal user intervention. On average, the user goes from needing 10-20 clicks to segment every image to needing just 4.76 clicks.

There is still work to do, and we have detailed some ideas that we think would be good approaches to improving the method. We definitely think we have taken a step forward towards the fully-automatic segmentation of multi-stain histology images of arteries.

Appendix A

Verhoeff and probability images

In this appendix we show the 42 original images and their respective probability images. These results are absolutely automatic and do not include any manual adjustment.

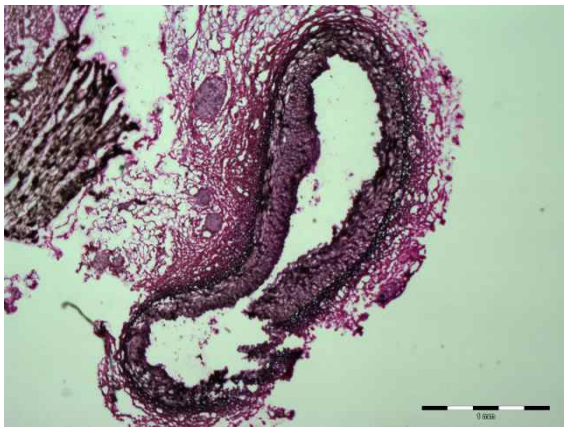


Figure A.1: 10346 LAD A Ver 2X

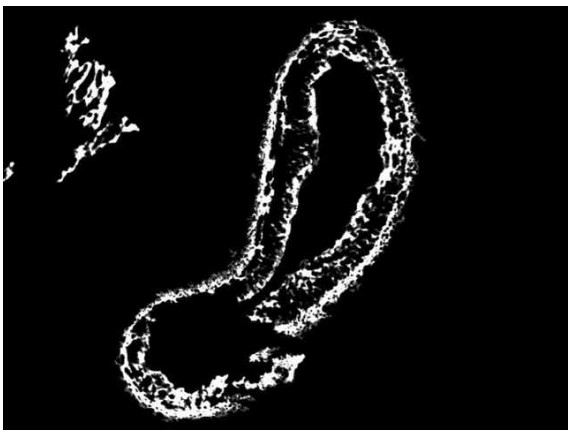


Figure A.2: 10346 LAD A Ver 2X

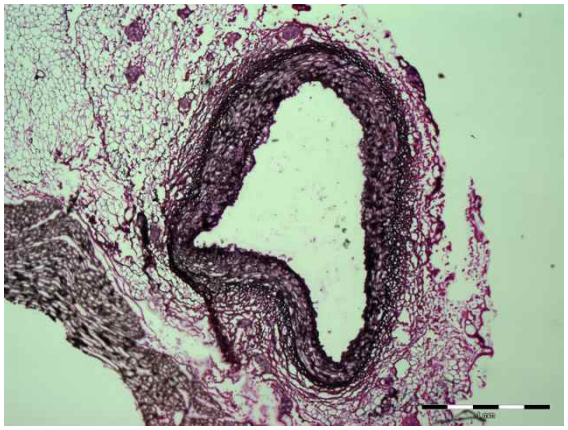


Figure A.3: 10346 LAD B Ver 2X

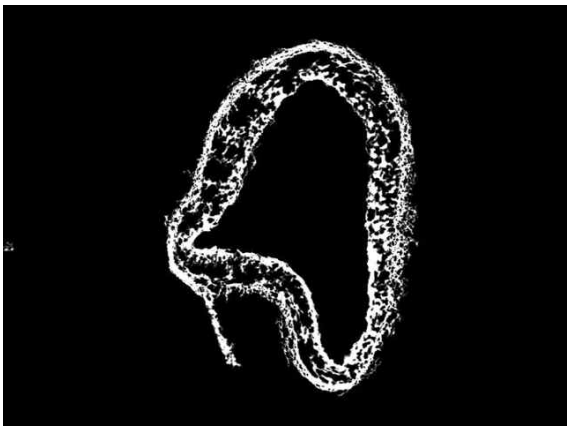


Figure A.4: 10346 LAD B Ver 2X

Figure A.5: 10346 LCX A Ver 2X

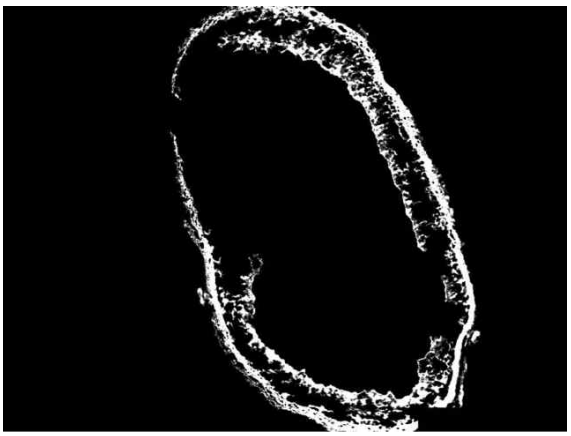


Figure A.6: 10346 LCX A Ver 2X

Figure A.7: 10346 LCX B Ver 2X

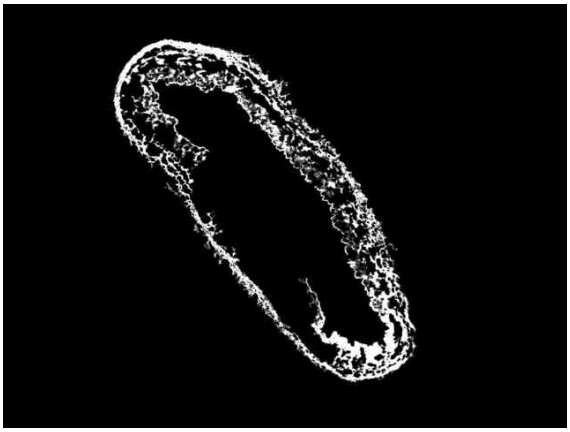


Figure A.8: 10348 LCX B Ver 2X

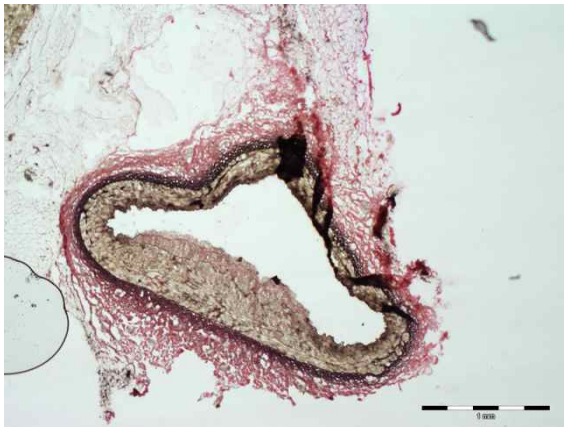


Figure A.9: 10346 RCA A Ver 2X

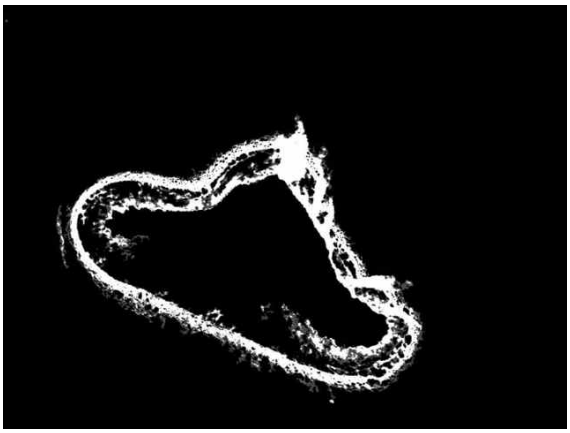


Figure A.10: 10346 RCA A Ver 2X

Figure A.11: 10346 RCA B Ver 2X

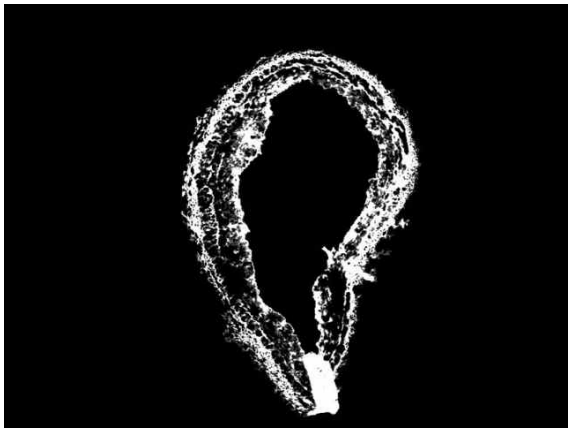


Figure A.12: 10346 RCA B Ver 2X

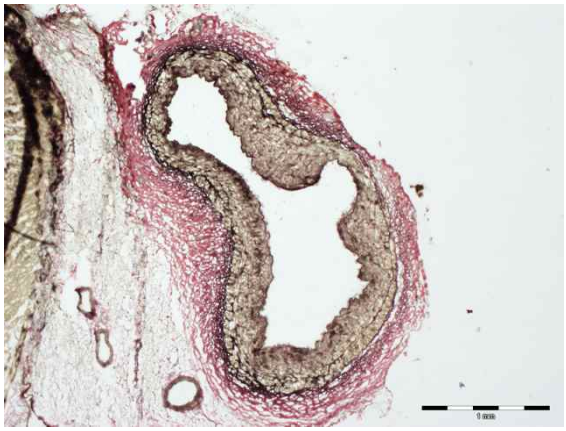


Figure A.13: 10346 RCA BC Ver 2X

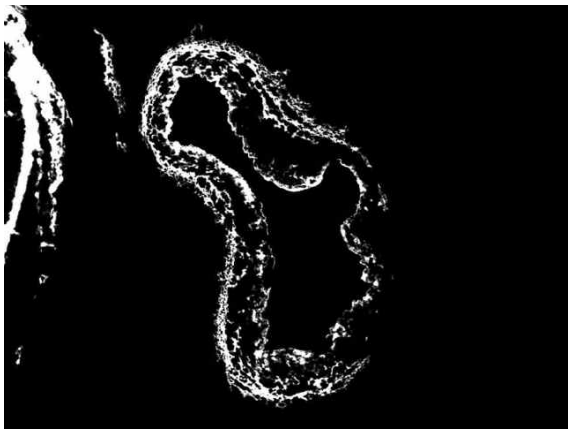


Figure A.14: 10346 RCA C Ver 2X

Figure A.15: 10348 LAD E1 Ver 2X

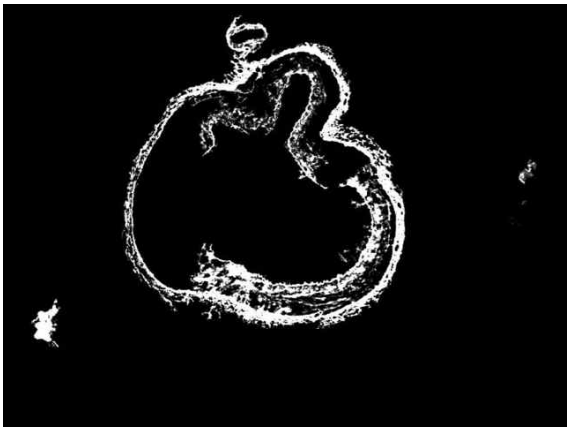


Figure A.16: 10348 LAD E1 Ver 2X

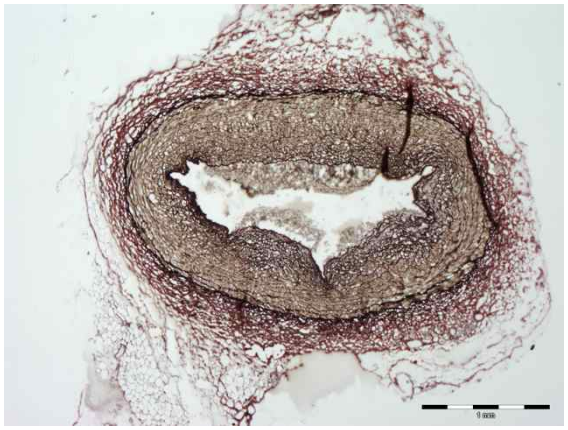


Figure A.17: 10348 LCX A Ver 2X

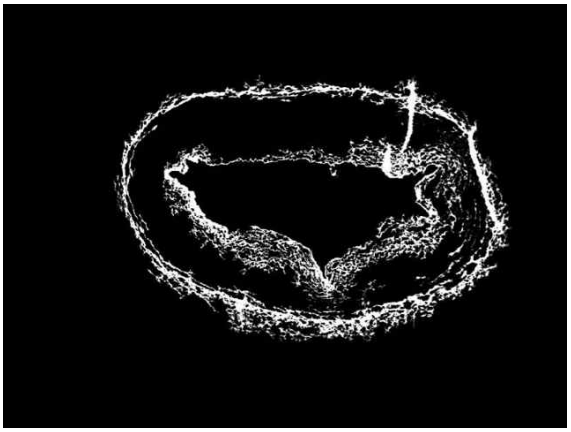


Figure A.18: 10348 LCX A Ver 2X

Figure A.19: 10348 LCX B Ver 2X

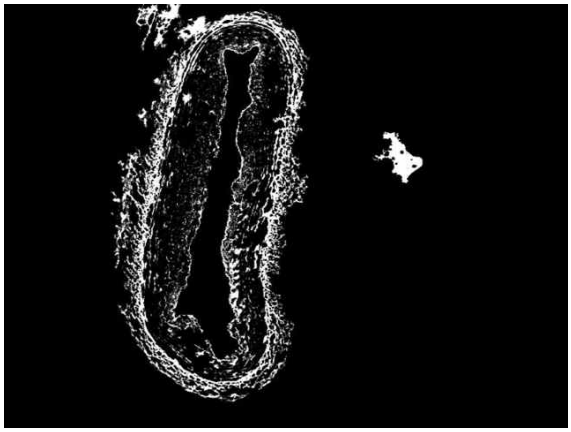


Figure A.20: 10348 LCX B Ver 2X

Figure A.21: 10351 LCX C Ver 2X

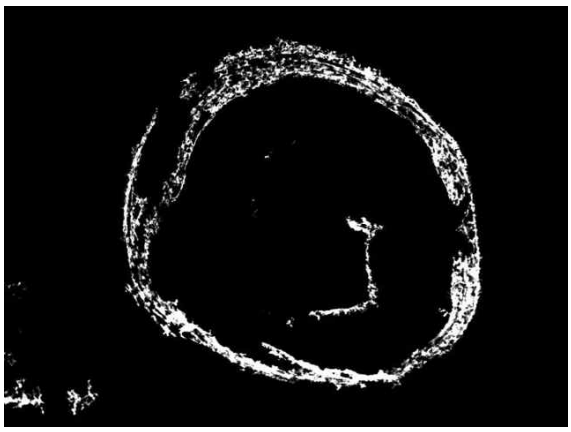


Figure A.22: 10351 LCX C Ver 2X

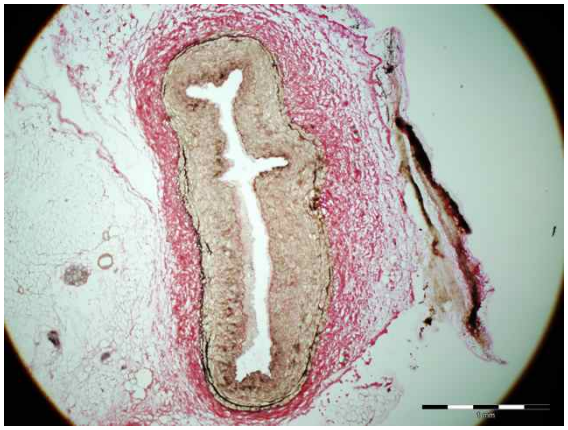


Figure A.23: 10352 RCA A Ver 2X

Figure A.24: 10352 RCA A Ver 2X

Figure A.25: 10352 RCA B Ver 2X

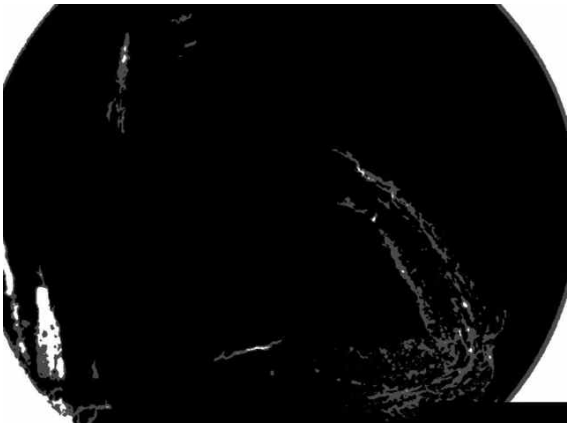


Figure A.26: 10352 RCA B Ver 2X

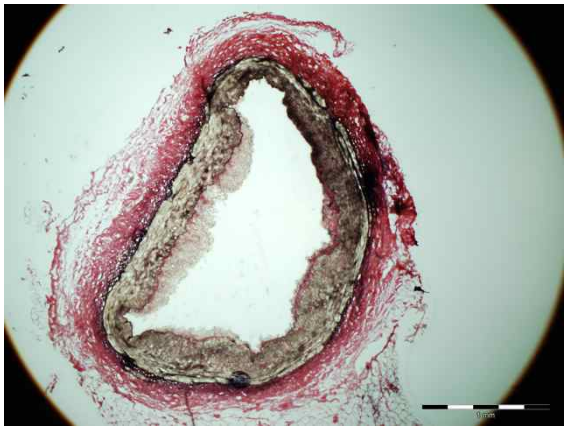


Figure A.27: 10352 RCA C Ver 2X

Figure A.28: 10352 RCA C Ver 2X

Figure A.29: 10352 RCA D Ver 2X

Figure A.30: 10352 RCA D Ver 2X

Figure A.31: 10356 RCA A Ver 1.25X

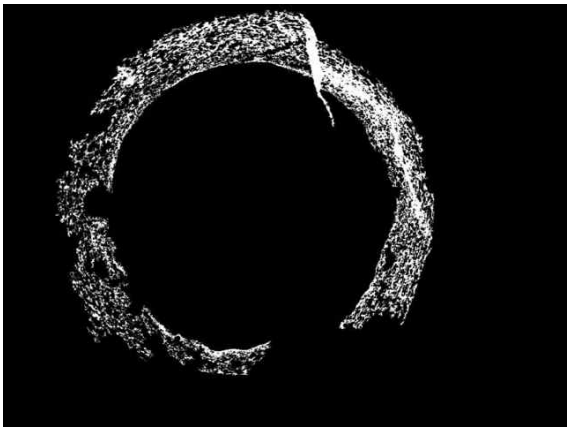


Figure A.32: 10356 RCA A Ver 1.25X

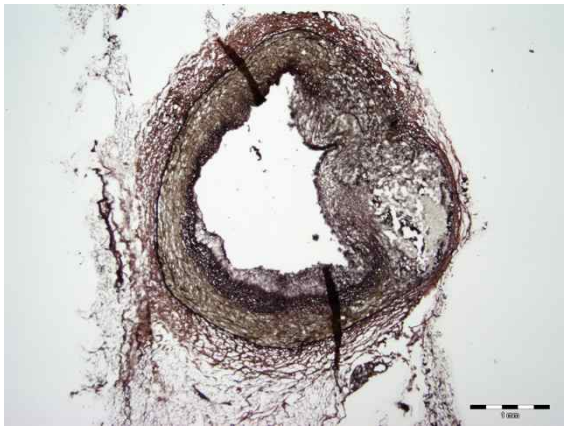


Figure A.33: 10356 RCA B Ver 1.25X

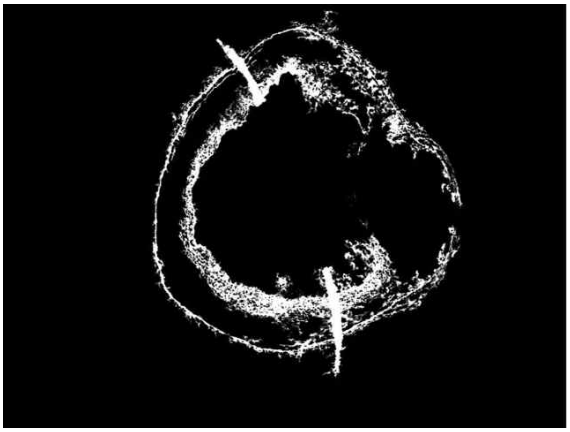


Figure A.34: 10356 RCA B Ver 1.25X

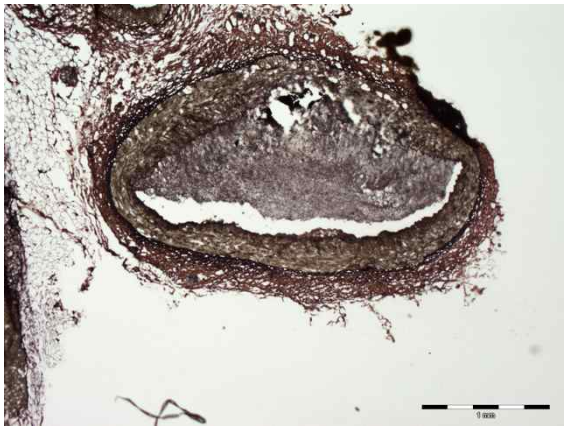


Figure A.35: 10356 RCA B1 Ver 2X

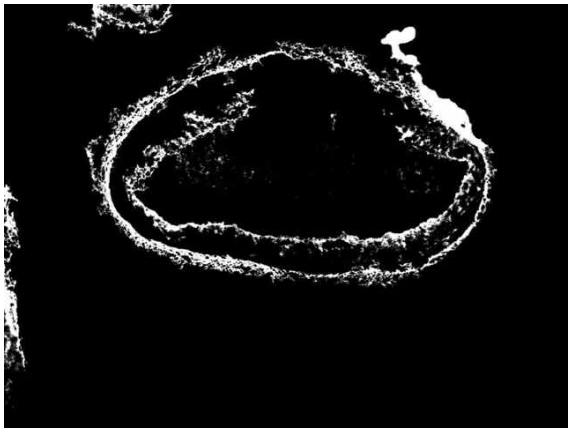


Figure A.36: 10356 RCA B1 Ver 2X

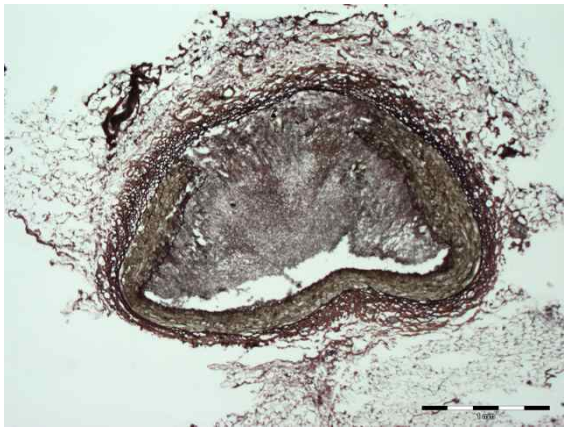


Figure A.37: 10356 RCA C Ver 2X

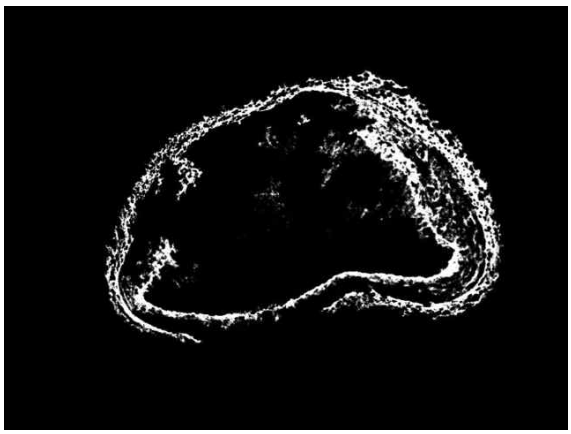


Figure A.38: 10356 RCA C Ver 2X

Figure A.39: 10356 RCA D Ver 2X

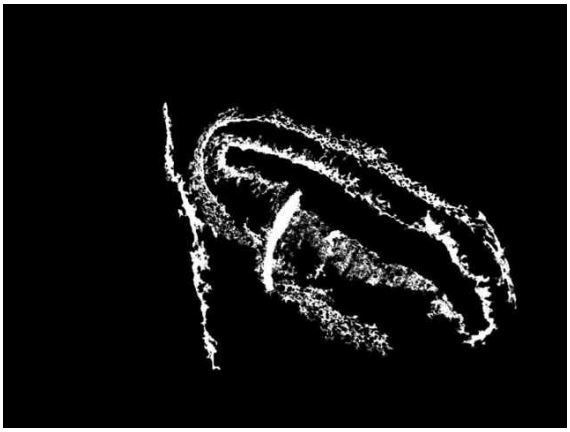


Figure A.40: 10356 RCA D Ver 2X

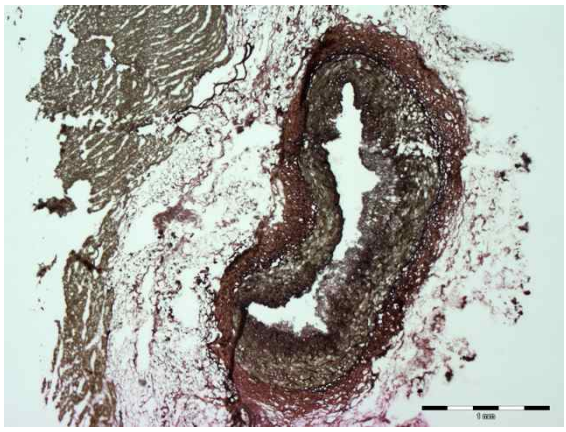


Figure A.41: 10356 RCA E Ver 2X

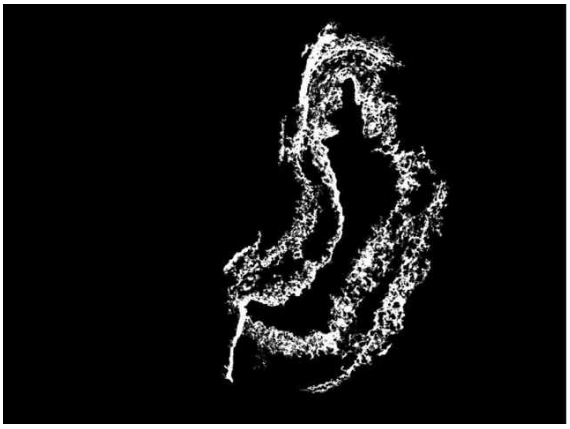


Figure A.42: 10356 RCA E Ver 2X

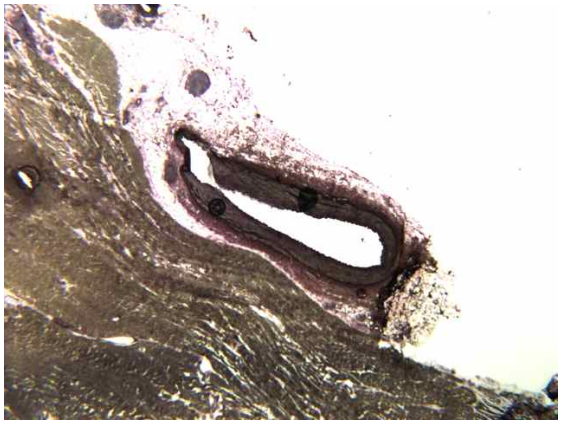


Figure A.43: 10357 LAD B Ver 2X

Figure A.44: 10357 LAD B Ver 2X

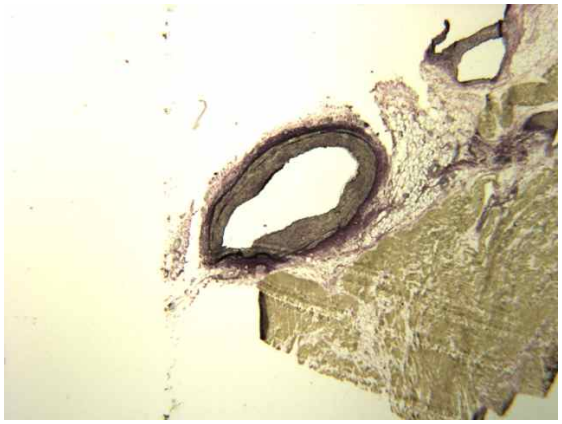


Figure A.45: 10357 LAD C Ver 2X

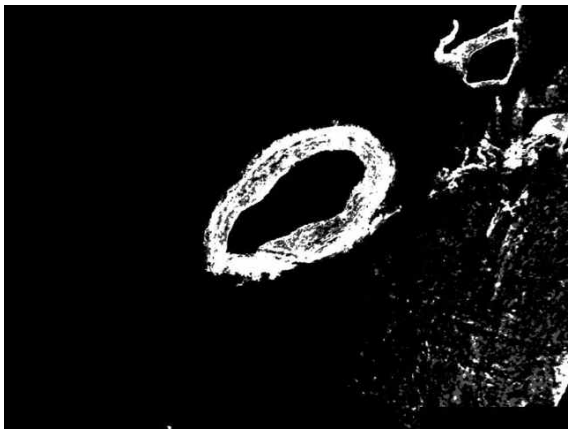


Figure A.46: 10357 LAD C Ver 2X

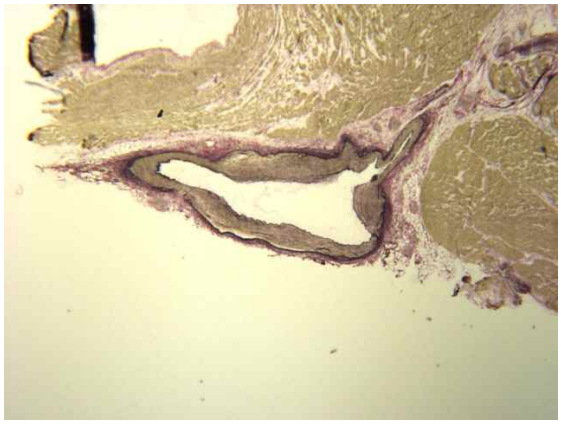


Figure A.47: 10357 LAD D Ver 2X

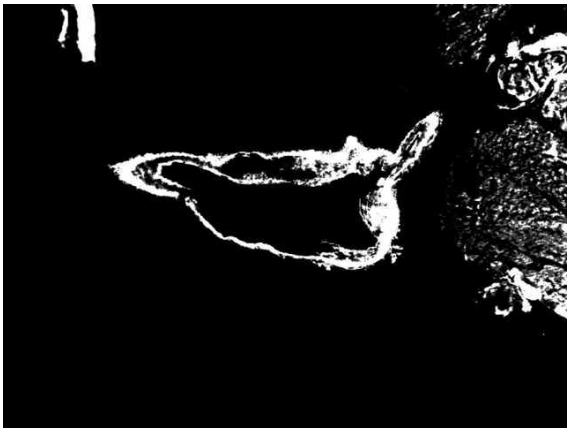


Figure A.48: 10357 LAD D Ver 2X

Figure A.49: 10359 RCA A Ver 2X

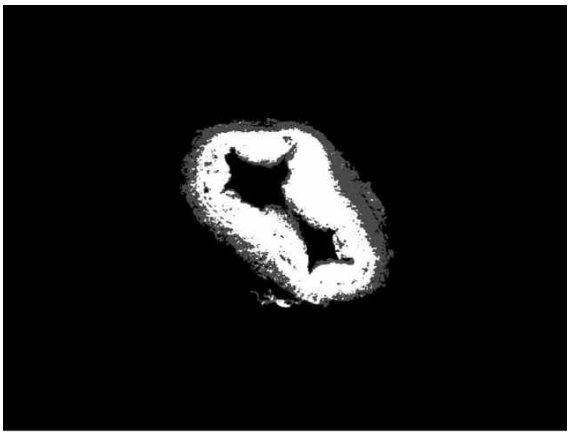


Figure A.50: 10359 RCA A Ver 2X

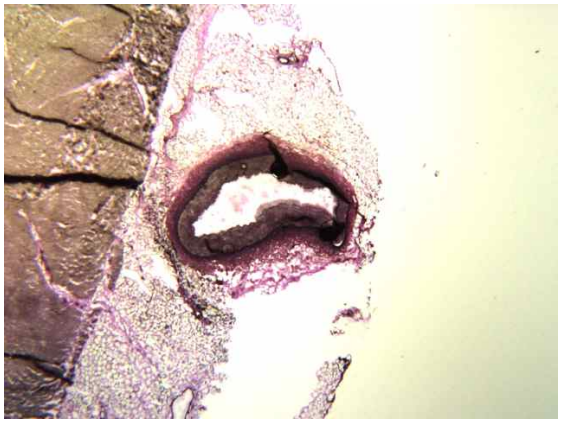


Figure A.51: 10359 RCA B Ver 2X

Figure A.52: 10359 RCA B Ver 2X

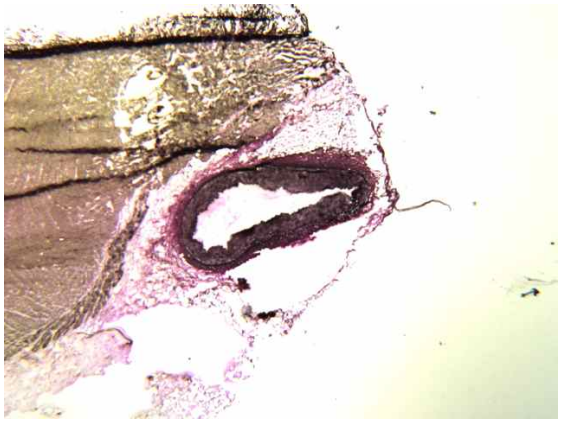


Figure A.53: 10359 RCA C Ver 2X

Figure A.54: 10359 RCA C Ver 2X

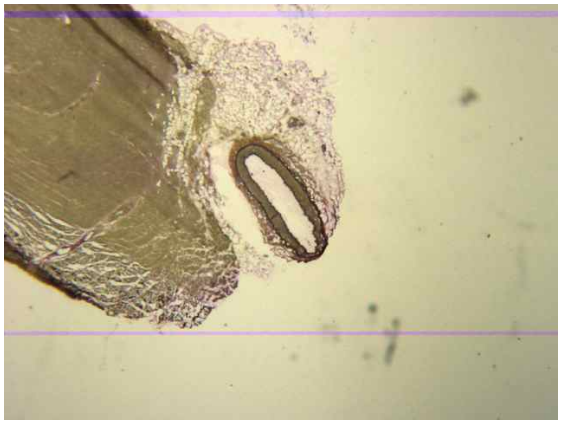


Figure A.55: 10359 RCA D Ver 2X

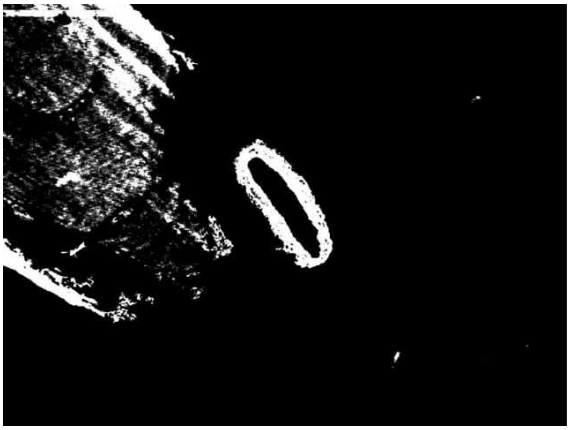


Figure A.56: 10359 RCA D Ver 2X

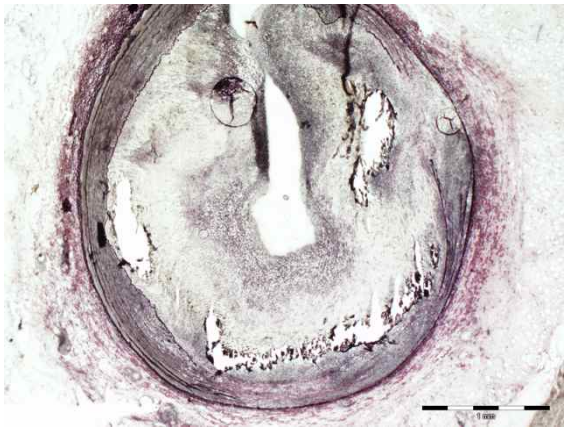


Figure A.57: 10362 LAD A Ver 2X

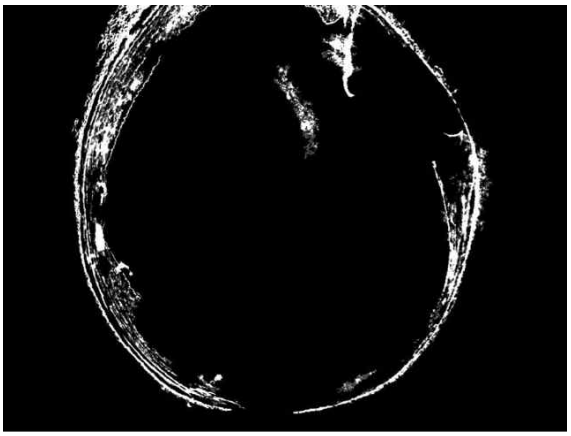


Figure A.58: 10362 LAD A Ver 2X

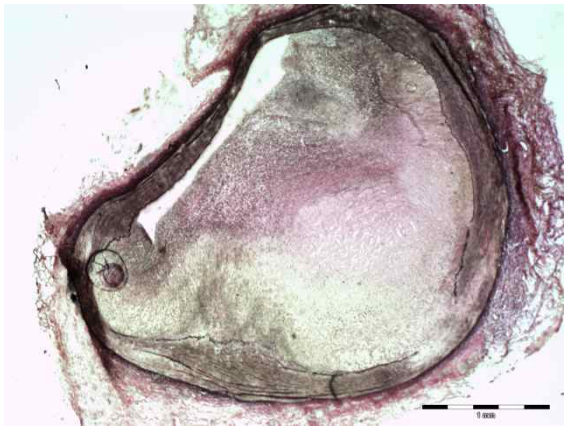


Figure A.59: 10362 LAD A2 Ver 2X

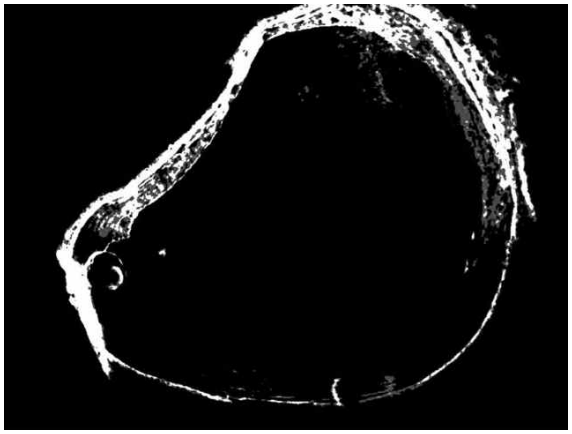


Figure A.60: 10362 LAD A2 Ver 2X

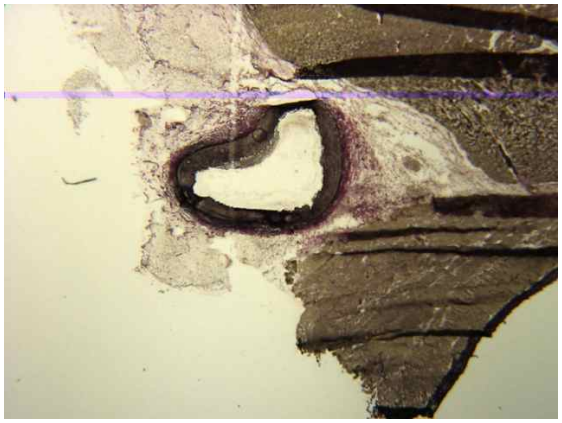


Figure A.61: 10362 LCX E Ver 2X

Figure A.62: 10362 LCX E Ver 2X

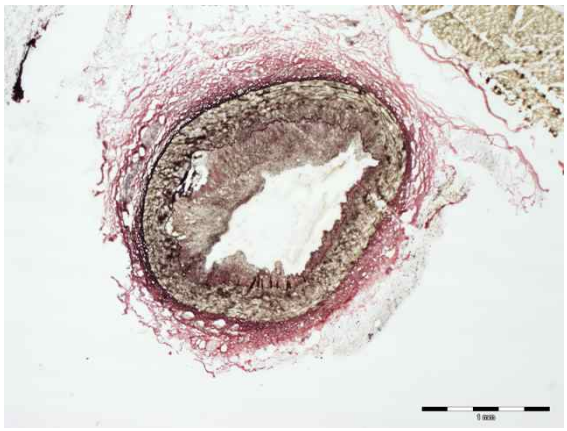


Figure A.63: 10364 OM B Ver 2X

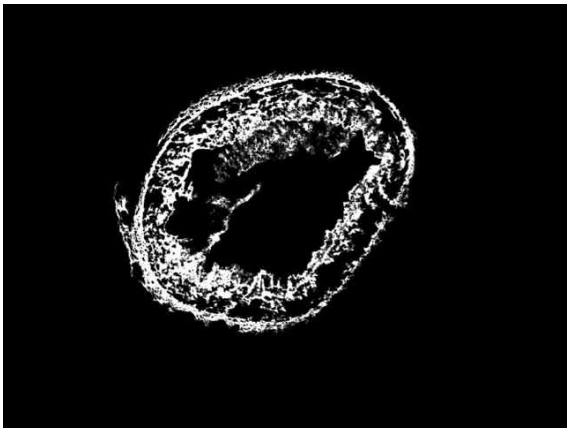


Figure A.64: 10364 OM B Ver 2X

Figure A.65: 10364 OM D Ver 2X

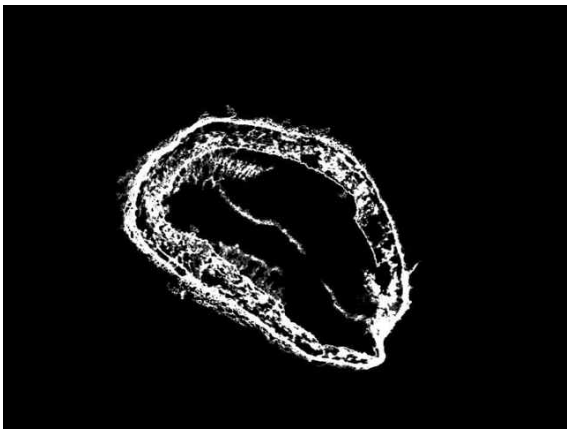


Figure A.66: 10364 OM D Ver 2X

Figure A.67: 10364 OM E Ver 2X

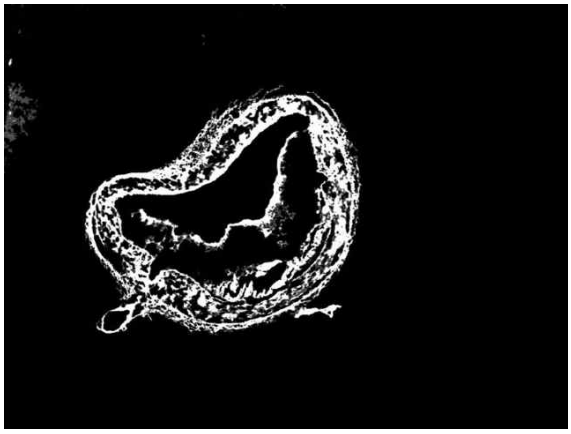


Figure A.68: 10364 OM E Ver 2X

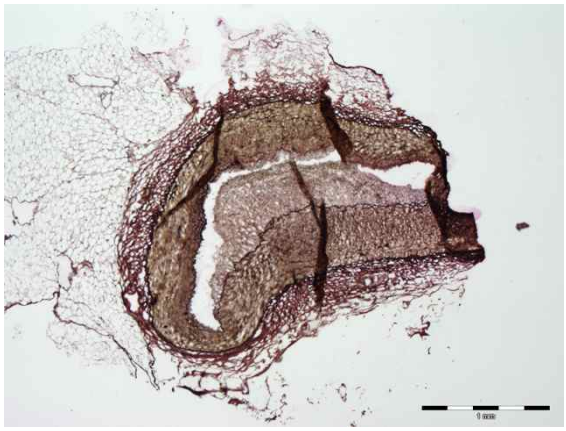


Figure A.69: 10565 LCX A Ver 2X

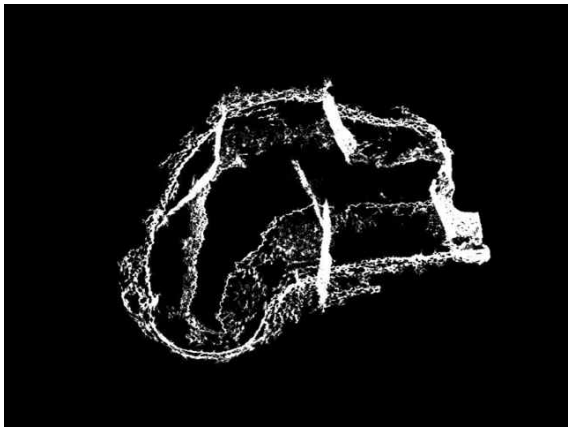


Figure A.70: 10565 LCX A Ver 2X

Figure A.71: 10565 LCX B Ver 2X

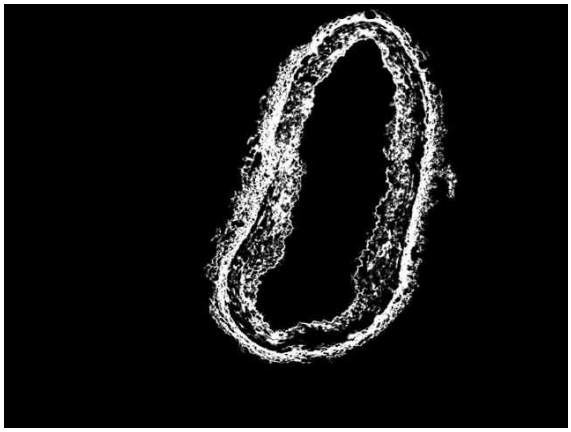


Figure A.72: 10565 LCX B Ver 2X

Figure A.73: 10565 LCX C Ver 2X

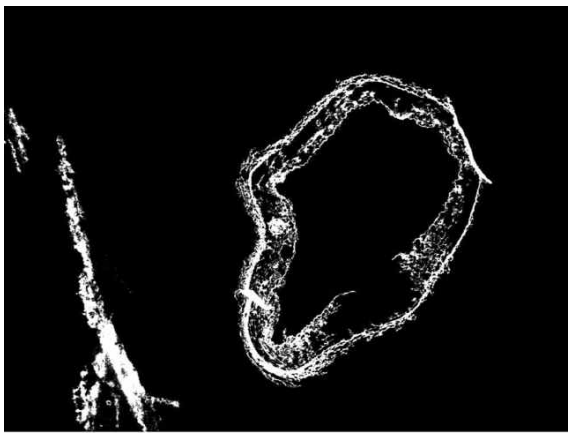


Figure A.74: 10565 LCX C Ver 2X

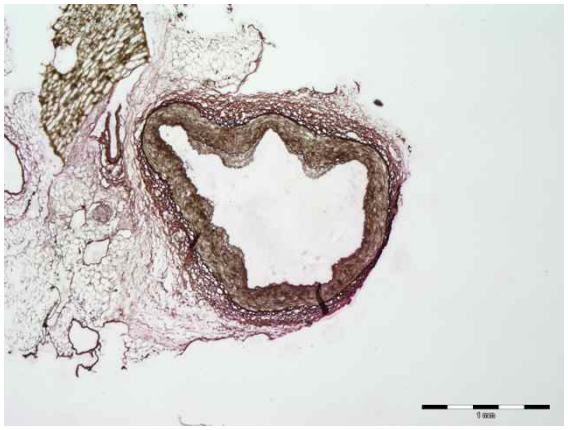


Figure A.75: 10565 LCX D Ver 2X

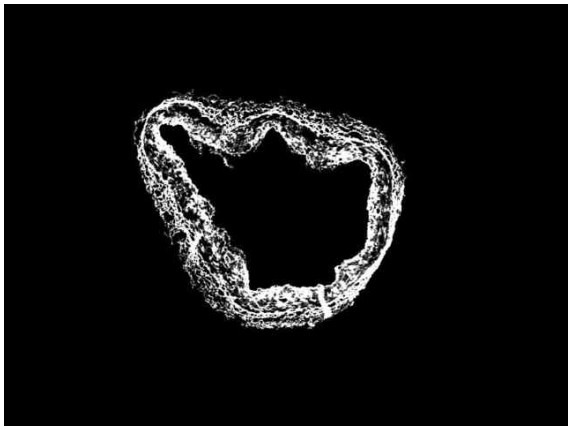


Figure A.76: 10565 LCX D Ver 2X

Figure A.77: 11105 LAD A Ver 2X

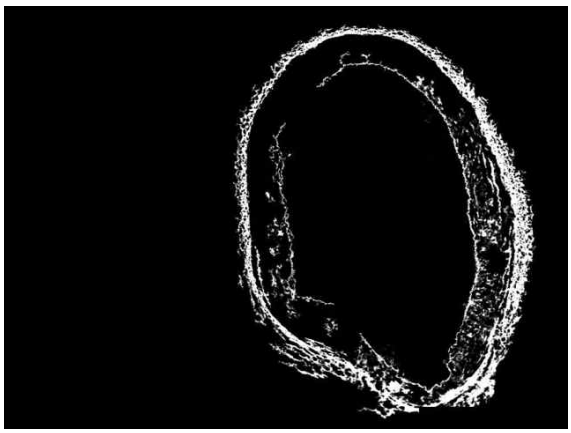


Figure A.78: 11105 LAD A Ver 2X

Figure A.79: 11105 LAD B Ver 2X

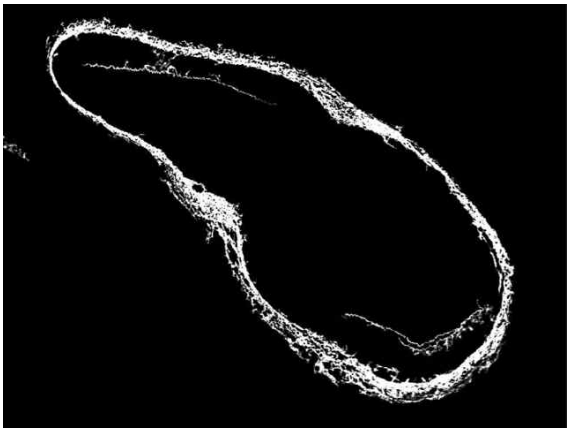


Figure A.80: 11105 LAD B Ver 2X

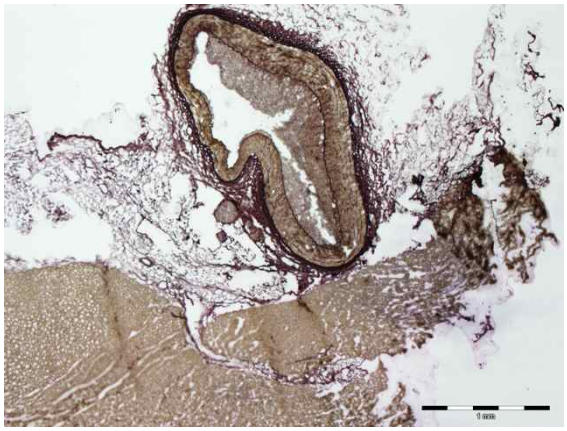


Figure A.81: 11105 LAD C Ver 2X

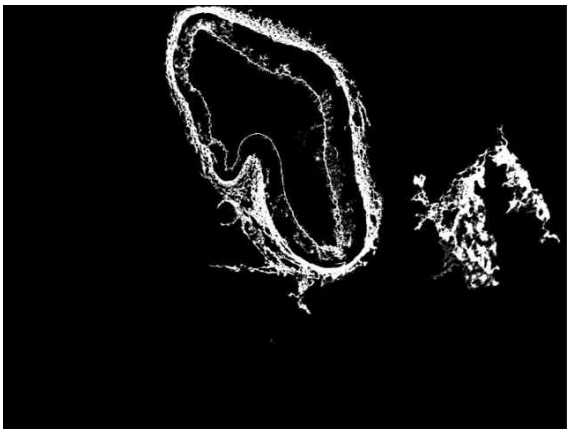


Figure A.82: 11105 LAD C Ver 2X

Figure A.83: 11105 LAD D1 Ver 2X

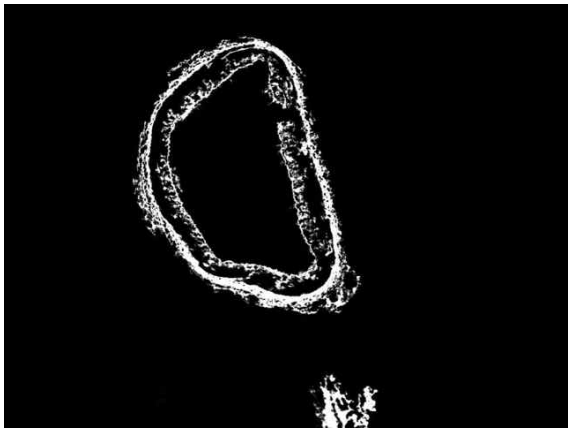
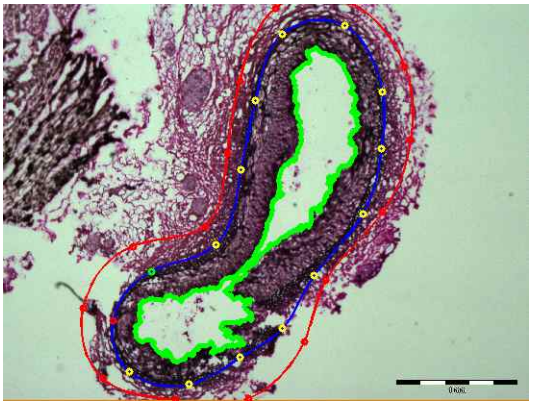


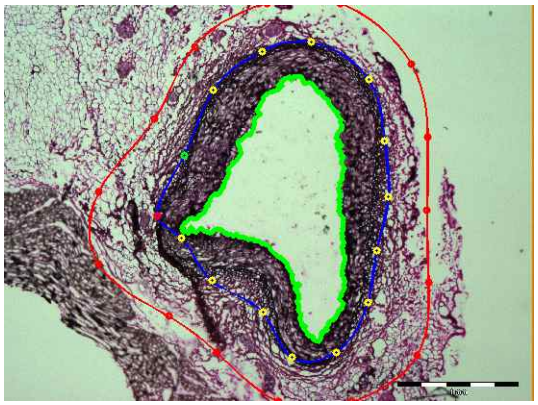
Figure A.84: 11105 LAD D1 Ver 2X

Appendix B

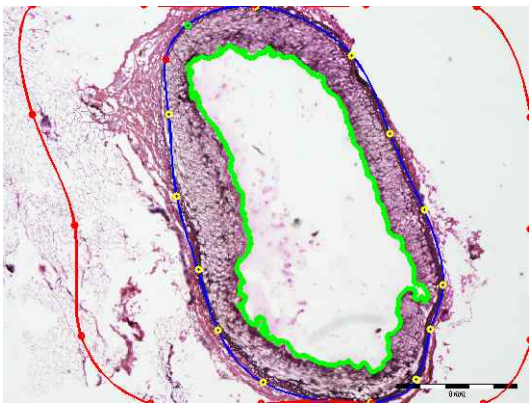
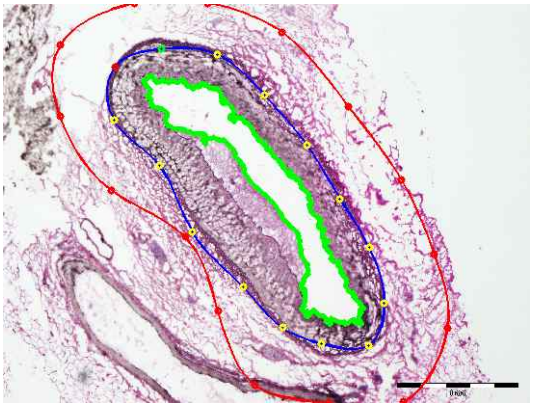
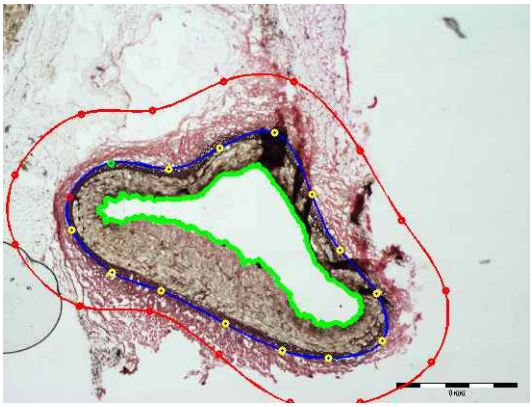
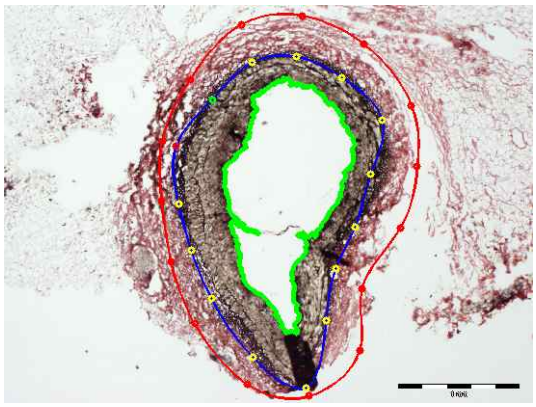
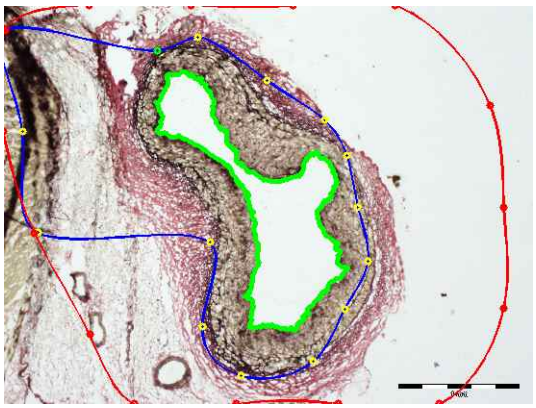
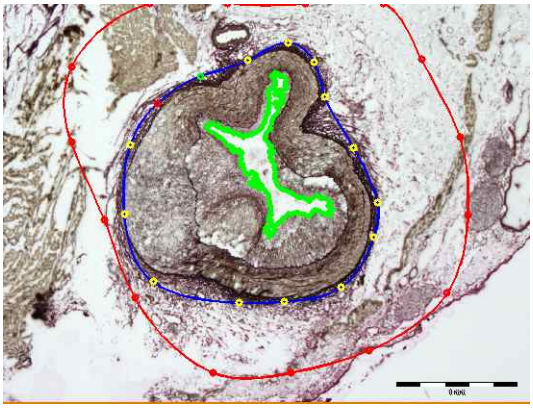
Automatic segmentation of the 42 images

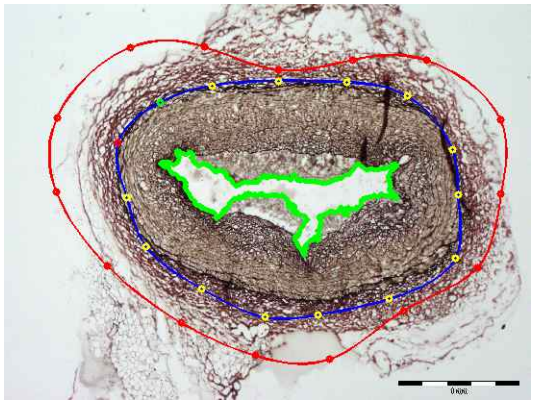
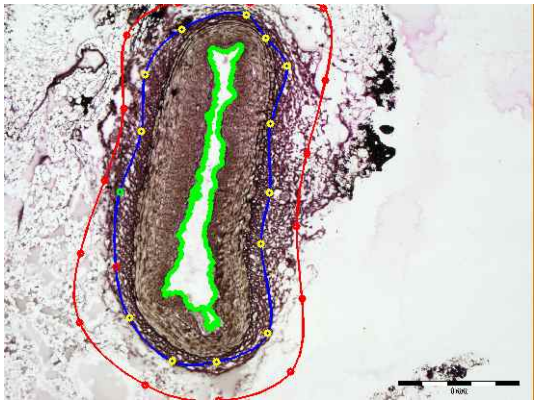
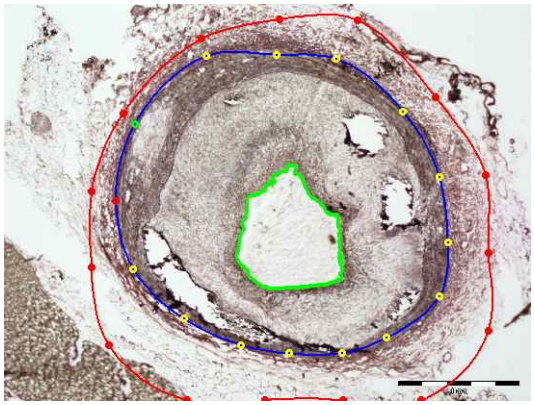
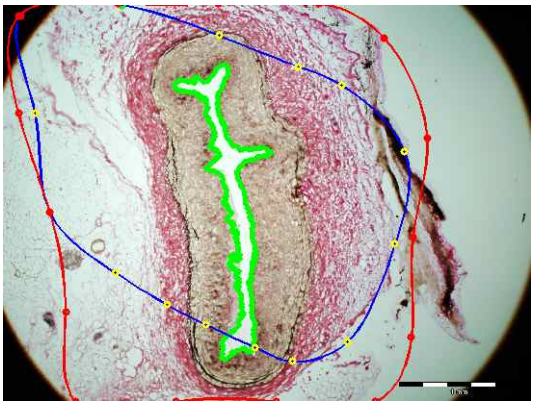
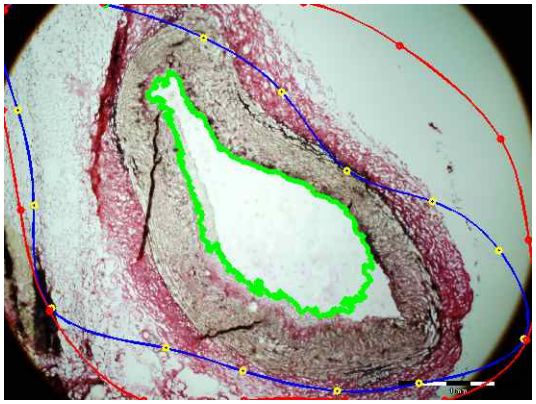
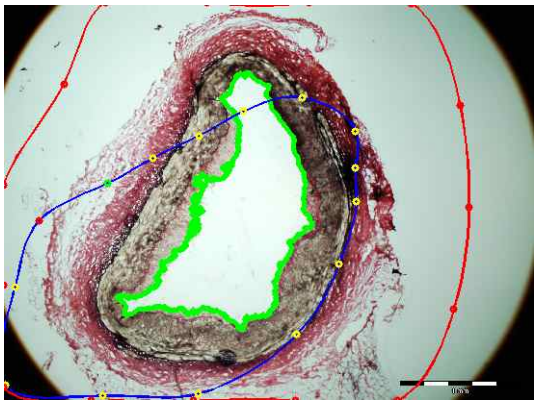
In this appendix we show the 42 original images and the automatic inner boundary, outer boundary and the snake starting position. These results are absolutely automatic and do not include any manual adjustment.



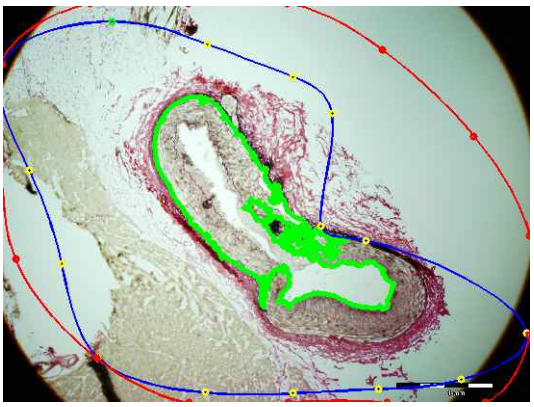


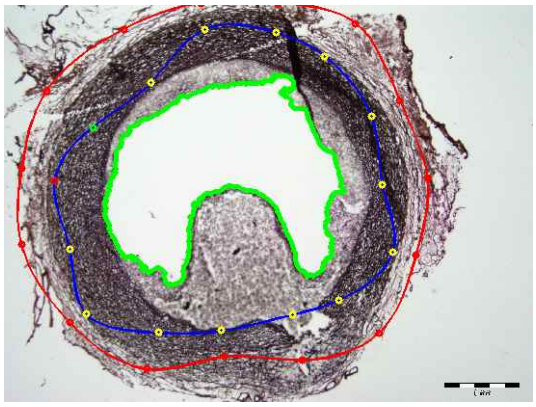
B. AUTOMATIC SEGMENTATION OF THE 42 IMAGES

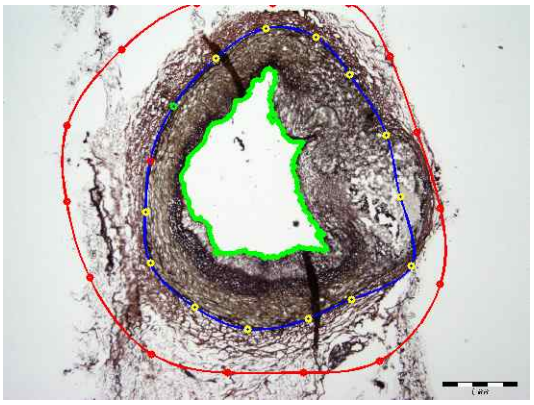







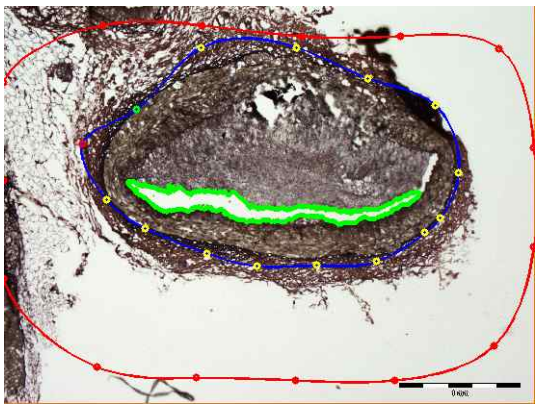







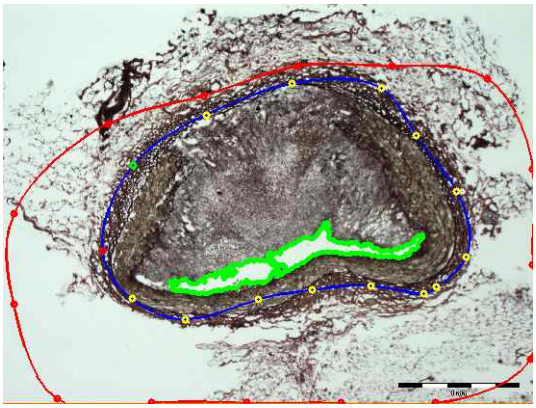
B. AUTOMATIC SEGMENTATION OF THE 42 IMAGES

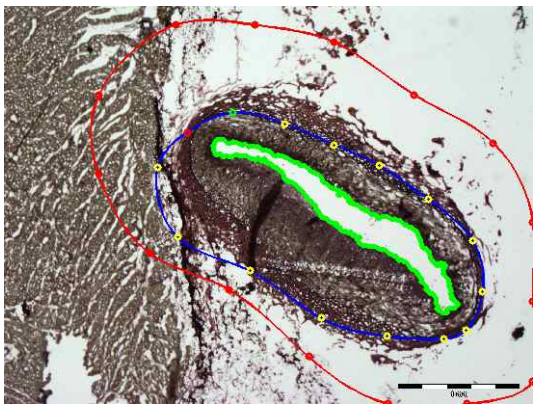


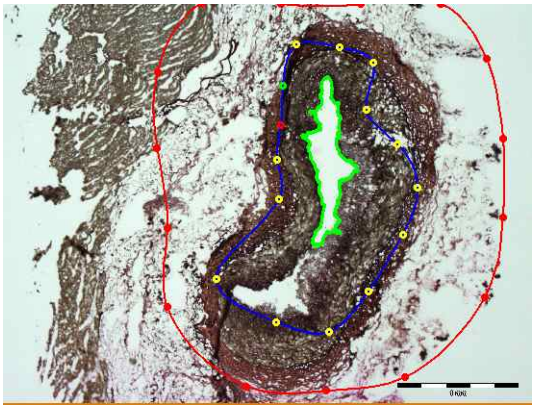
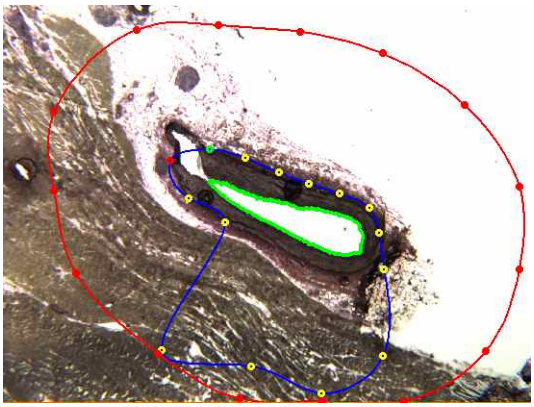
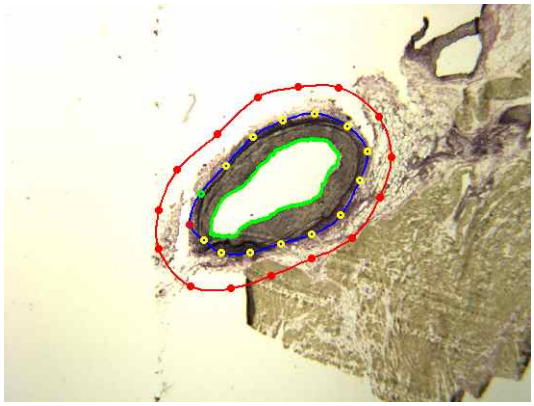
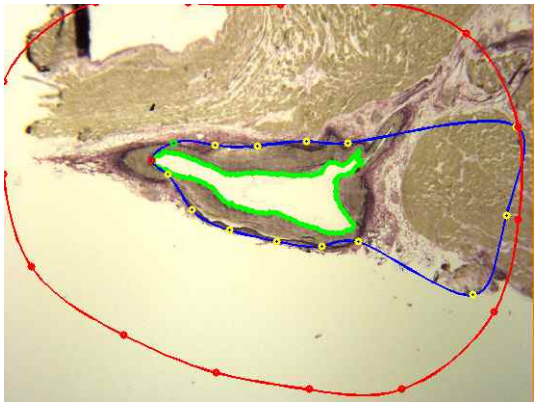




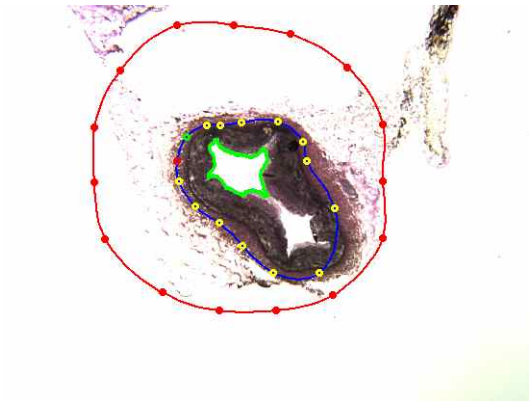
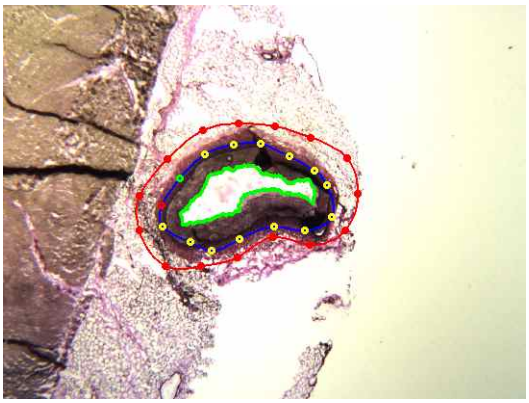
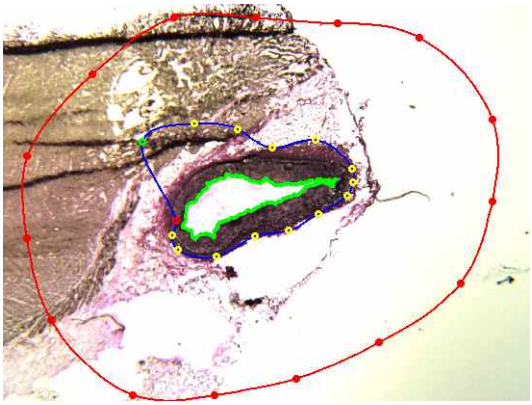
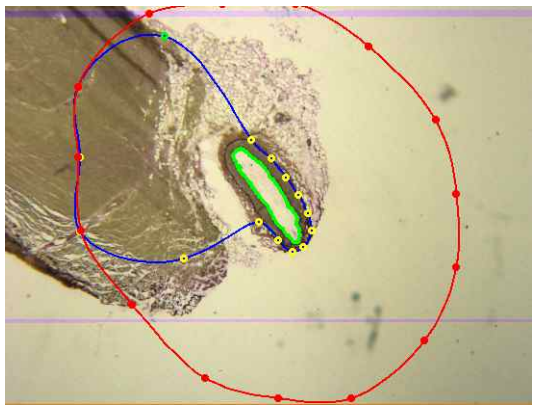
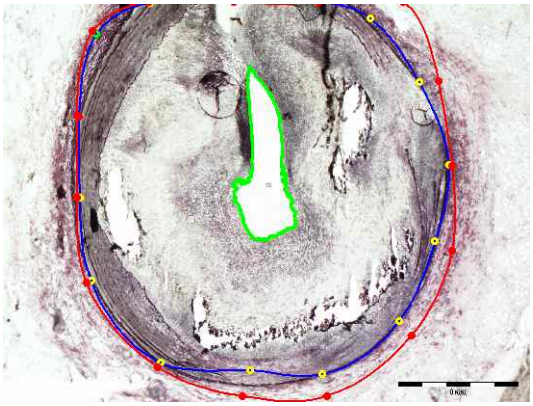
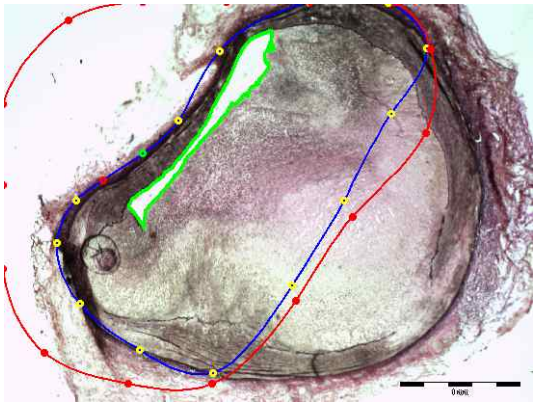


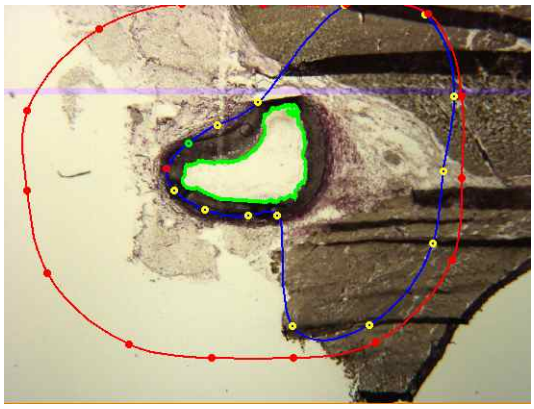
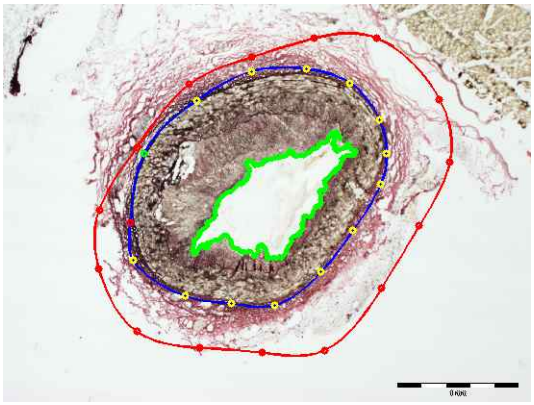
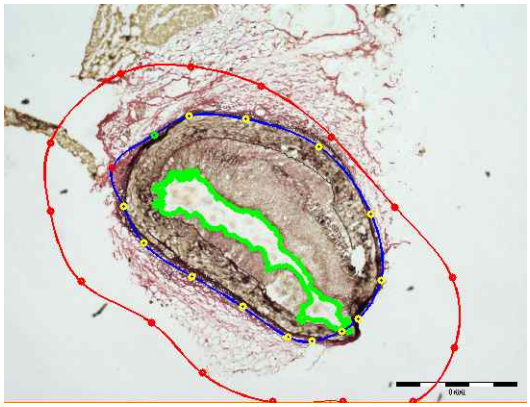
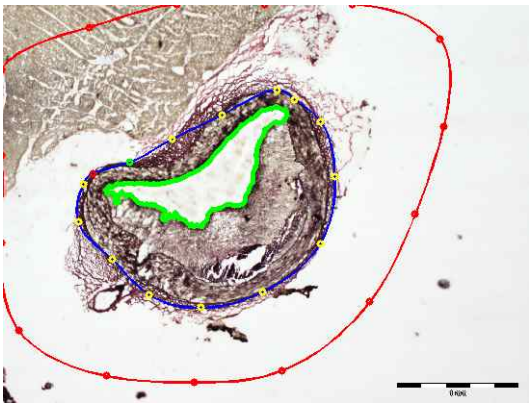
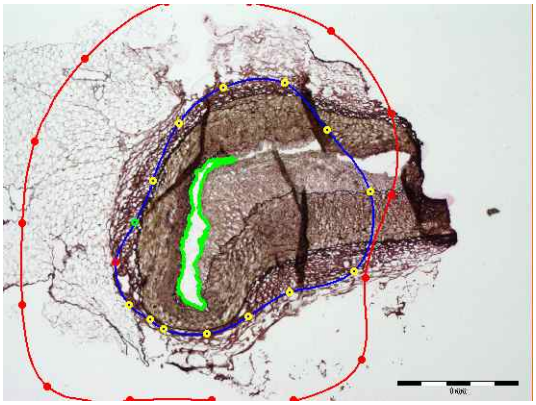
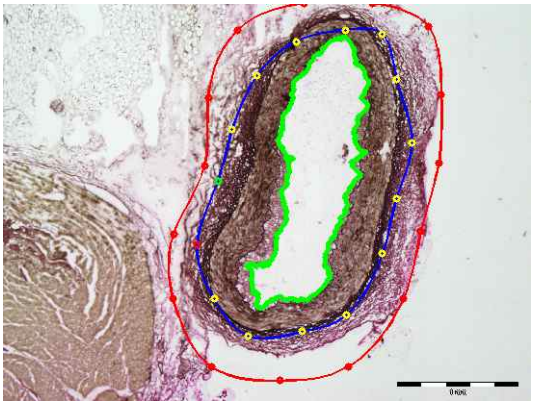




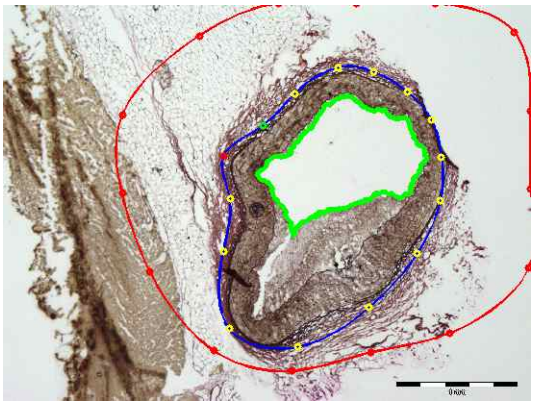





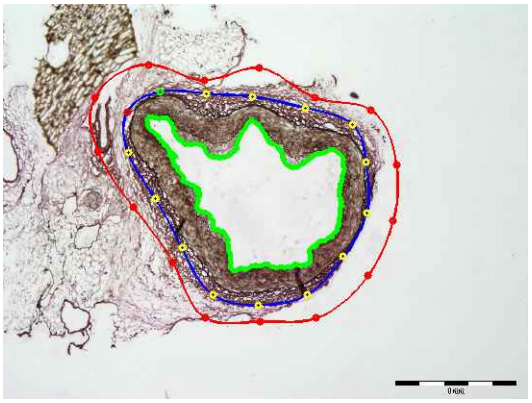
B. AUTOMATIC SEGMENTATION OF THE 42 IMAGES

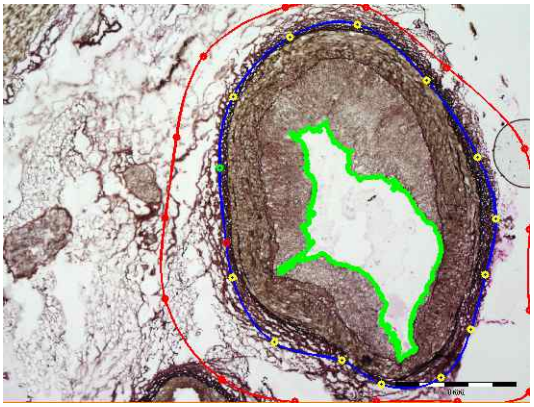







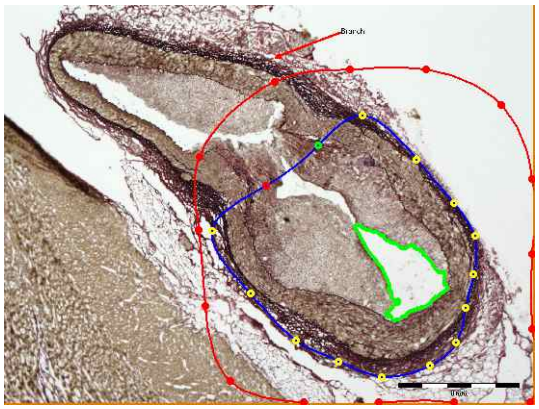







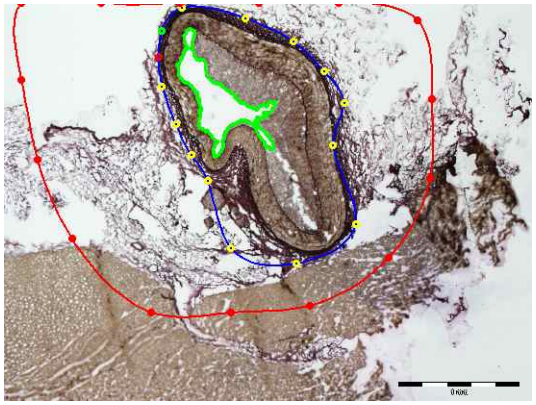
B. AUTOMATIC SEGMENTATION OF THE 42 IMAGES

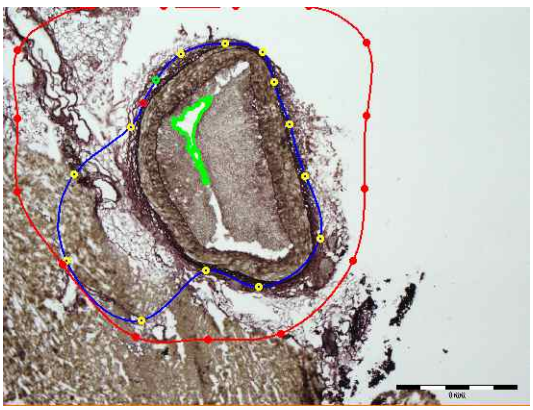












Bibliography

[1] American Heart Association. Heart disease and stroke statistics - 2008 update. Technical report, American Heart Association. Dallas, Texas., 2008. 1

[2] Cleveland Clinic. Webmd: Coronary heart disease (chd). 2

[3] M.D. Alan Berger. Medlineplus medical encyclopedia. 2

[4] Wikipedia article on atherosclerosis. 4

[5] Corwin D. Santeusanio G. Lehr HA., Mankoff DA. and Gown AM. Application of photoshop-based image analysis to quantification of hormone receptor expression in breast cancer. *Journal of Histochemistry and Cytochemistry*, 45:1559–1565, 1997. 6

[6] Teeling P. Lehr HA., van der Loos CM. and Gown AM. Complete chromogen separation and analysis in double immunohistochemical stains using photoshop-based image analysis. *Journal of Histochemistry and Cytochemistry*, 47:119–126, 1999. 6

[7] George Masganas. Image processing for semi-automatic segmentation and area determination in multi-stain histology slices from porcine coronary arteries. Master's thesis, Northeastern University, Boston, Massachusetts, March 2007. 11

[8] Gonzalez R.C. and Woods R.E. *Digital Image Processing*. Prentice Hall, 2007. 16, 23

[9] Woods R.E. Gonzalez R.C. and Eddins S.L. *Digital Image Digital Image Processing Using MATLAB*. Prentice Hall, 2004. 23

[10] Luo M. A colour management framework for medical imaging applications. *Computerized Medical Imaging and Graphics*, 30(6-7):357–361. 27

[11] Kokate J.Y. Leland K.J. Hansen G.L., Sparrow E.M. and Iaizzo P.A. Wound status evaluation using color image processing. *IEEE Transactions on Medical Imaging*, 16(1):78–86, 1997. 27

[12] Devaux J.Y. Herbin M., Venot A. and Piette C. Color quantitation through image processing in dermatology. *IEEE Transactions on Medical Imaging*, 9(3):262–269, 1990. 27

[13] Yli-Harja O. Zhang W. Niemisto A., Hu L. and Shmulevich I. Quantification of in vitro cell invasion through image analysis. *Engineering in Medicine and Biology Society*, 1:1703–1706, 2004. 32

[14] Peters R.A. Thomas J.G. and Jeanty P. Automatic segmentation of ultrasound images using morphological operators. *IEEE Transactions on Medical Imaging*, 10(2):180–186, 1991. 32

[15] Khan S. Di Ruberto C., Dempster A. and Jarra B. Segmentation of blood images using morphological operators. *15th International Conference on Pattern Recognition*, 2:397–400, 2000. 32

[16] Ablameyko S. Nedzved A. and I. Pitas. Morphological segmentation of histology cell images. *Proceedings of the International Conference on Pattern Recognition*, 1:500–503, 2000. 32

[17] Schafer R.W. Maragos, P. and M.A. Butt. *Mathematical Morphology and its applications to image and signal processing.* 1996. 32

[18] Witkin A. Kass M. and Terzopoulos D. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988. 39

[19] DeBoor C. *A Practical Guide to Splines.* Springer, 1994. 40

[20] Tuceryan M. and Jain A.K. Texture analysis. *The Handbook of Pattern Recognition and Computer Vision (2nd Edition)*, pages 207–248, 1998. 71

[21] Konofagou E.E. Katouzian A., Baseri B. and Laine A.F. Texture-driven coronary artery plaque characterization using wavelet packet signatures texture-driven coronary artery plaque characterization using wavelet packet signatures texture-driven coronary artery plaque characterization using wavelet packet signatures. *Biomedical Imaging: From Nano to Macro*, pages 14–17, May 2008. 71

[22] Li L. Castellano G., Bonilha L. and Cendes F. Texture analysis of medical images. *Clinical Radiology*, 59(12):1061–1069. 71

[23] Kontos D Megalooikonomou V, Zhang J and Bakic P.R. Analysis of texture patterns in medical images with an application to analysis of texture patterns in medical images with an application to breast imaging. *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 6514, 2007. 71

[24] Sharma S. Shukla K.K. Pradhan S. Sharma N., Ray A.K. and Aggarwai L.M. Segmentation and classification of medical images using texture-primitive features: Application of bam-type artificial neural network. *Journal of Medical Physics*, 33(3):119–126, 2008. 71

[25] Wikipedia article on gray level co-occurrence matrix. 72

[26] Herderick E. Gaddipati A., Cornhill J.F. and Yagel R. An efficient method for automated segmentation of histochemically stained slides. *17th Annual International Conference of the Engineering in Medicine and Biology Society*, 1:497–498, 1995.

[27] Wilson G.D. Vojnovic B. Loukas C.G. and Linney A. An image analysis-based approach for automated counting of cancer cell nuclei in tissue sections. *Cytometry Part A*, 55A(1):30–42, 2003.

[28] Springall D. Wootton R. and Polak J. *Image analysis in histology: conventional and confocal microscopy.* Cambridge University Press, 1995.

[29] Mokji M.M. and Bakar A. Gray level co-occurrence matrix computation based on haar wavelet. *Computer Graphics, Imaging and Visualisation*, pages 273–279, 2007.

[30] Bitter I. Van Uitert, R. and R.M. Summers. Detection of colon wall outer boundary and segmentation of the colon wall based on level set methods. *28th Annual International Conference of the Engineering in Medicine and Biology Society*, pages 3017–3020, 2006.

[31] Chassery J. M. Garbay C. and Brugal G. An iterative region growing process for segmentation based on local color similarity and global shape criteria. *Analytical and Quantitative Cytology and Histology*, 8:25–34, 1986.

Declaration

I herewith declare that I have produced this paper without the prohibited assistance of third parties and without making use of aids other than those specified; notions taken over directly or indirectly from other sources have been identified as such. This paper has not previously been presented in identical or similar form to any other English or foreign examination board.

The thesis work was conducted from February 2008 to August 2008 under the supervision of Prof. Dana Brooks and Prof. Umit A. Coskun, at Northeastern University in Boston.

Boston,