Distribution Agreement

Tianli Han April 15, 2015

Automatic Strut Detection of Bioresorbable Stent from Optical Coherence Tomography Images

by

Tianli Han

Dr. Alessandro Veneziani
Adviser

Department of Mathematics and Computer Science

Dr. Alessandro Veneziani

Adviser

Dr. Jed Brody

Committee Member

Dr. Jinho Choi

Committee Member

2015

Automatic Strut Detection of Bioresorbable Stent from Optical Coherence Tomography Images

By

Tianli Han

Dr. Alessandro Veneziani

Adviser

An abstract of
a thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Sciences with Honors

Department of Mathematics and Computer Science

2015

Abstract

Automatic Strut Detection of Bioresorbable Stent from Optical Coherence Tomography Images

By Tianli Han

Detection of stents is used for 3D reconstruction of vessel lumens which is important to provide reliable medical analysis. The detection is mainly based on medical images such as OCT. A huge set of image data from hundreds of patients make tedious manual detection unaffordable. In this paper, we propose a new automatic detection algorithm which has a considerable improvement. In most cases, the algorithm can automatically find all stent structures without any error.

Automatic Strut Detection of Bioresorbable Stent from Optical Coherence Tomography Images

By

Tianli Han

Dr. Alessandro Veneziani

Adviser

A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Sciences with Honors

Department of Mathematics and Computer Science

2015

Acknowledgements

I would like to acknowledge Ph.D. candidate Boyi Yang, who acted as a co-adviser on this project and spent many hours teaching me through the research. I would like to thank my thesis adviser, Dr. Alessandro Veneziani, who provided generous help through the year. I also would like thank Dr. Steve Batterson, who offered me the great opportunity. I appreciate the support from Dr. Jinho Choi and Dr. Jed Brody for being my committee members. I appreciate the Department of Mathematics and Computer Science as a whole for helping me explore the unknown world of science. I would like to thank Emory University for providing such an interdisciplinary environment and excellent faculties and staffs.

Table of Contents

Automatic Strut Detection of Bioresorbable Stent from Optical Coherence Tomography Images

Tianli Han

Adviser: Dr. Alessandro Veneziani

April 15, 2015

1. Introduction

1.1 Bioresorable Stents

Angioplasty is an endovascular procedure to widen narrowed or obstructed arteries or veins, typically to treat arterial atherosclerosis. An empty, collapsed balloon, known as a balloon catheter, is passed over a wire into the narrowed locations and then inflated to a fixed size. The balloon forces expansion of the stenosis within the vessel and the surrounding muscular wall, opening up the blood vessel for improved flow, and the balloon is then deflated and withdrawn. A stent may be inserted at the time of ballooning to ensure the vessel remains open.

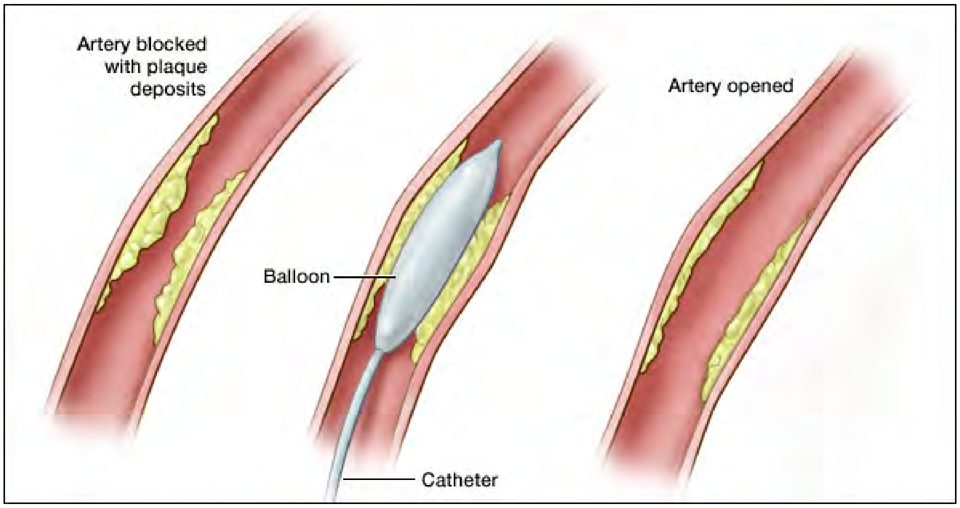


Figure 1: Balloon angioplasty

(Image source: http://virchicago.com/peripheral-artery-disease/)

In the 1970s and 1980s, interventional cardiologists confronted two of the main problems associated with angioplasty: elastic recoil and neointimal hyperplasia. Implantation of the stent appeared to reduce the occurrence of these potential symptoms. The concept of using an implantable prosthetic device to maintain the luminal integrity of diseased vessels came much earlier when laureate Alexis Carrel delivered his 1912 Nobel lecture where he described experiments with glass and metal tubes covered with paraffin that were inserted into canine thoracic aortae.[1] The idea was reintroduced by Charles Dotter in 1964, when he suggested that the temporary use of a silicone elastomer endovascular splint could maintain an adequate lumen after the creation of a pathway in a previously occluded vessel.[2]

However, various challenges suppressed any optimism that early stents might have a role to play in atherosclerotic vascular disease. One of them was the material chosen for the stent. The advent of metallic stents reduced restenosis to 20-30% from 50% noted with plain old balloon angioplasty. In the 1990s, metallic drug-eluting stents constrained restenosis within 10%. A drug-eluting stent (DES) is a scaffold placed into narrowed, diseased coronary arteries that slowly releases a drug to block cell proliferation. This prevents fibrosis that, together with thrombi, could otherwise block the stented artery, a process called restenosis. Nonetheless, metal stents had their downside as well. Metal stents are associated with acute and late thrombosis, need for long-term dual antiplatelet therapy and lasting impact of a metal scaffold in the artery interfering with adequate computed tomography (CT) or magnetic resonance imaging (MRI), vascular remodeling, and coronary vasomotion (including endothelial function).

Extensive stenting also makes future coronary artery bypass graft (CABG) procedures difficult, if not impossible.[3] CABG is one treatment for coronary heart disease (CHD). During CABG, a healthy artery or vein from the body is connected, or grafted, to the blocked coronary artery. The grafted artery bypasses the blocked portion of the coronary artery. This creates a new path for oxygen-rich blood to flow to the heart muscle.

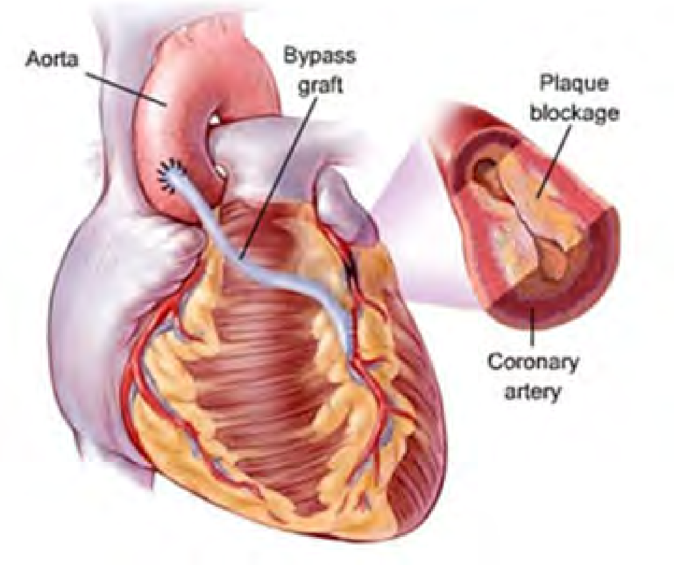


Figure 2: Coronary artery bypass graft

(Image source: http://www.kulvinderlall.com/coronary-artery-bypass-grafting.php)

For the last decade, initial experiences with different types of bioresorbable stents to replace metal stents have been under intense study. A bioresorbable stent is a stent manufactured from a material that may dissolve or be absorbed in the body. Bioresorbable stents have potential advantages including reduced or abolished late stent thrombosis, improved lesion imaging with computed tomography or magnetic resonance, facilitation of repeat treatments (surgical or percutaneous) to the same site, restoration of vasomotion, and freedom from side-branch obstruction by struts and from strut fracture-induced restenosis.[4] The BVS (Bioresorbable Vascular Scaffold) everolimus-eluting stent from Abbott Vascular is the first bioabsorbable stent to have clinical and imaging outcomes similar to those metallic drug-eluting stent (DES) implantation for more than 5 years but with the potential possibility of full-stent absorption. One of the Abbott's products, the Absorb^{TM} Bioresorbable Vascular Scaffold was used in the patients from whom the OCT images in our research were attained.

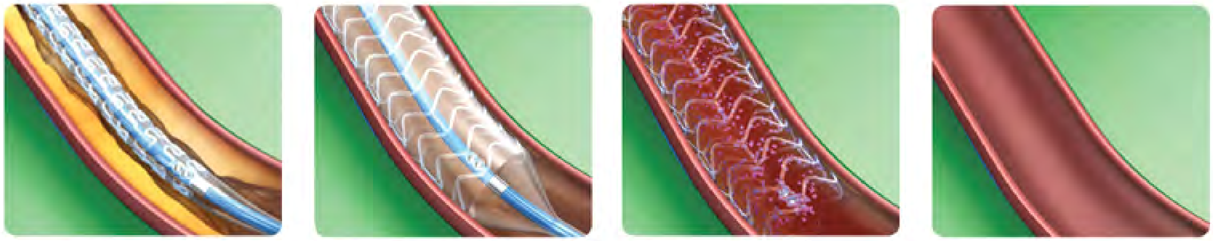


Figure 3: Bioresorbable Vascular Scaffold

(Image source: http://hkhearts.com/en/coronary-heart-surgery.html)

1.2 OCT

Optical coherence tomography (OCT) is a tissue-imaging modality of imaging the internal cross-sectional microstructure of tissues using measurements of optical backscattering or backreflection. It was first demonstrated in 1991.[5] OCT was initially applied for the eye as an example of transparent, weakly scattering media in 1993.[6] In late 1990s, advances in OCT technology have made it possible to image nontransparent tissues, thus enabling OCT to be applied in a wide range of medical specialties.[7] One of the most important advances was the use of longer wavelengths where optical scattering is reduced. By performing OCT imaging at 1300-nm wavelengths, image penetration depth of 2 to 3 mm can be achieved in most tissues. Studies have indicated that OCT is a useful method to assess both blood vessel morphology and the response of a vessel to a deployed stent.[8] OCT enables the noncontact, noninvasive imaging for a segment of blood vessel. These 2-dimension images can be reconstructed to form a 3-dimension model which is more intuitive for medical diagnosis.

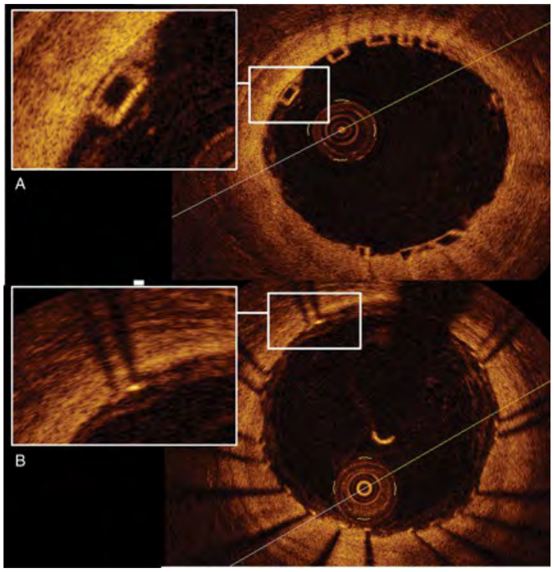
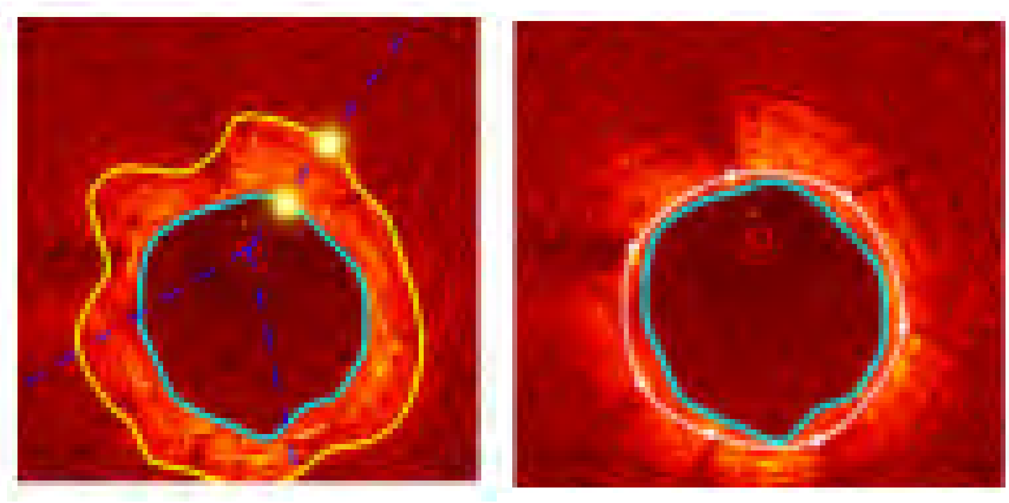


Figure 4: OCT imaging of the bioresorbable scaffolds(A) and metallic platform stents(B). (Image source: http://eurheartj.oxfordjournals.org/content/32/3/294)

1.3 Purpose of This Research

Detection of stent struts imaged in vivo by optical coherence tomography (OCT) after percutaneous coronary interventions (PCI) and quantification of in-stent neointimal hyperplasia (NIH) are important. In a recent project led by Habib Samady from Emory University and Georgia Tech, the results of the detection are used to reconstruct the 3D geometry of a deployed stent in a vessel lumen. An accurate 3D model improves computational fluid dynamics (CFD) simulations which can describe the wall shear stress (WSS) and other quantities inside a stented coronary artery. This is useful to analyze the potential risk of the stent. The large set of OCT image data makes tedious manual detection unaffordable. Thus an auto-detection algorithm is indispensable.

Researchers from Europe have observed that OCT cross-section images which contain metallic stent struts, generally include shadows behind the struts (see Figure 4(B)).[9-10] According to their discovery, analyzing the angular intensity energy distribution in the region of interest (ROI) provides clues to these shadows. To build such an energy map, rays from the image center are extended and the intersection points of these rays and the splines are computed. Interpolation of these two intersection coordinates (Figure 5(a)) and summation of corresponding image intensity values define the energy on a ray. Energy distribution over all the rays are analyzed: falling and rising of energy on these rays indicates the presence and absence of strut shadows. A strut on a shadow ray is the maximum bright intensity pixel group (Figure 5(b)).



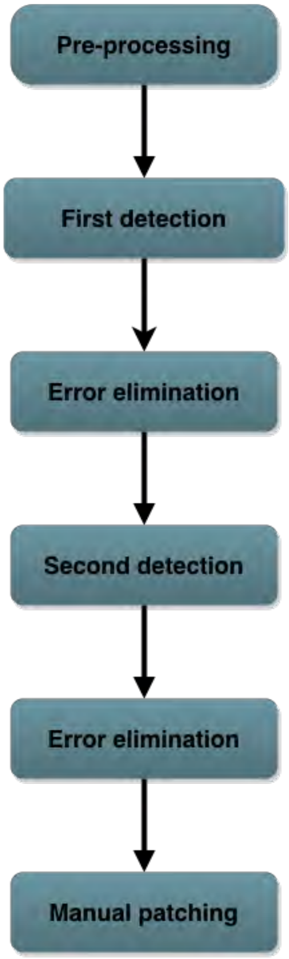
(a) The ROI with start and end points

(b) Detected struts

Figure 5: Related research from Europe[9-10]

However, in our OCT images of bioresorable stents, the kind of shadows exist only behind a few struts and are generally obscure (see Figure 4(a)) which means the method mentioned above may have a low accuracy in detecting struts in our cases. As a result, a new detection algorithm aimed at bioresorable stents is necessary to be created. A first attempt is the work by Shannon Buckley from Emory University. He created a MATLAB® algorithm that uses the image processing toolbox to automatically identify bioresorable stent structures in the OCT images. Although the algorithm provided an innovative idea to use binary images for the detection, it is not stable from image to image and does not eliminate error detection. In this research, we improve the effectiveness of Buckley's method.

From section 2, we first discuss the pre-processing stage. Then a first time and a second time detection algorithm are explained in details. After each of them, a different elimination method is necessary to provide to correct the errors. Before showing the results, a brief manual patching is presented as the last remediation. The following is a concise flow chart for our algorithm.



1.4 Related Work

Buckley's work does not rely on the shadows typically in OCT images of metallic stents. His code is modified from a publicly shared shape recognition program first written by Ahmed Samieh Abd El-Wahab from Al-Azhar University. There are eight steps to identify the stent structure locations in the artery.

Step 1: Read image into program

Step 2: Convert image from RGB into grayscale

Step 3: Find grayscale threshold and convert the image to binary

Step 4: Invert the binary image

Step 5: Find boundaries

Step 6: Determine object properties

Step 7: Plot stent boundaries and return the arrays of the boundaries' coordination

Step 8: Plot centroids of the stent and return the arrays of centroid values

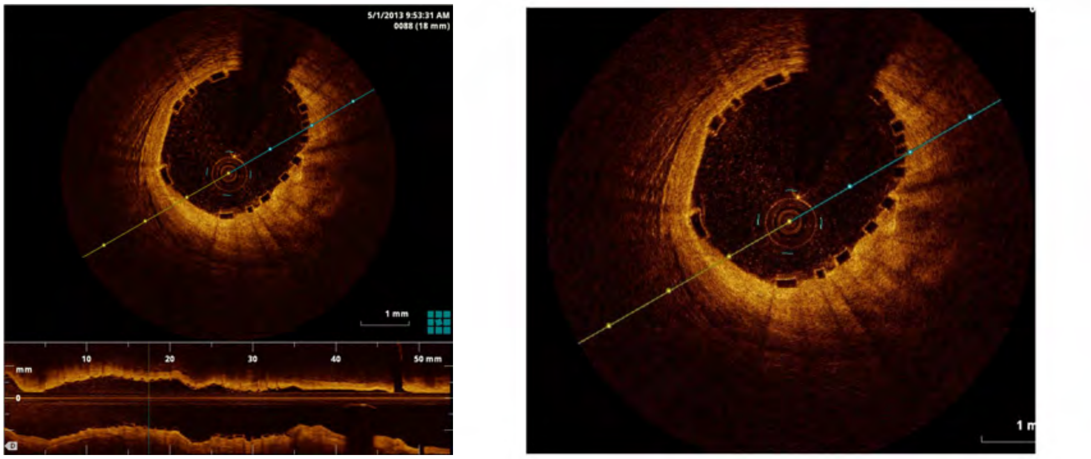
All these steps are part of the new algorithm and details will be discussed in Section 3. Thanks to Buckley's idea, our research can overcome the obstacles at an early stage.

2. Pre-processing

Not all of an OCT image is the region of interest. The most important part lies along the vessel wall where all the stents are located (see Figure 4(a)). Restricting the zone to a small but critical area can avoid meaningless search and immensely reduce the uncorrected detection.

2.1 Lower Part

The first step of our algorithm is to read in an image. One of the parameters of our auto-detection function is ImageFile, which is the variable name of the OCT image to detect. Then the function imread() saves the image as an RGB image named RGB_ORIGINAL. We have observed that in most images, the vessel wall is positioned at the upper center part. Thus the lower part of the image as well as the margin, can be cut out at the pre-possessing stage. We can easily use MATLAB® built-in function imcrop(RGB_ORIGINAL,[100 50 800 700]) to realize it. Here, 100 and 50 are the x and y values respectively as the coordination of the upper left corner point for the cropped image. 800 and 700 are the length and width of the new image.



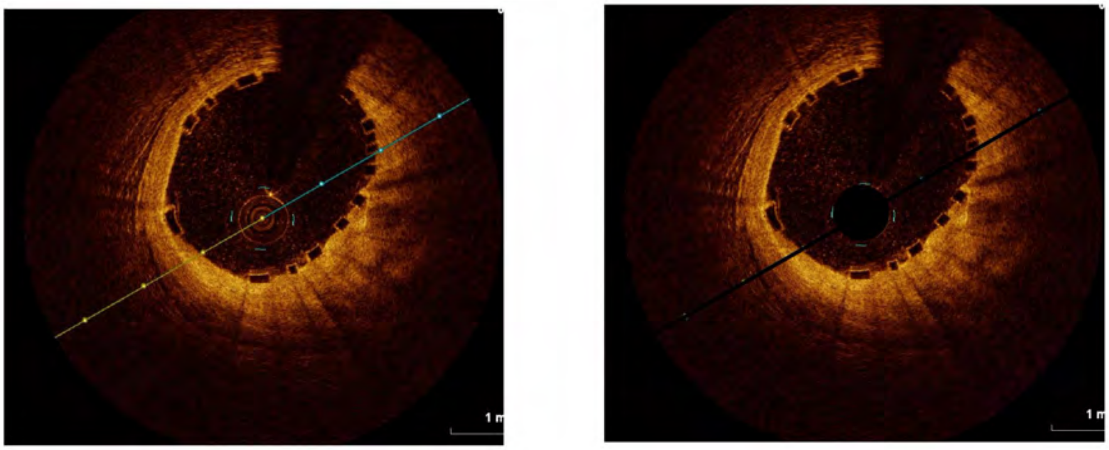
(a) Before cut

(b) After cut

Figure 6: Lower Part Cut

2.2 Circle and Line

The center of every OCT image is also the center of a circle which is the image of the catheter. The circle's radius varies in different data sets of OCT images and remains the same in one set. A typical OCT image comes with a line which enables the catheter to align all the cross-sectional images. Similarly, in one set of images, the line keeps unchanged. Since the circle and the line are definitely not part of the stents structure, they can be covered by setting the areas black. Later in the main procedure, we only recognize the bright area. Consequently, the circle and line region will be ignored.



(a) Before cut

(b) After cut

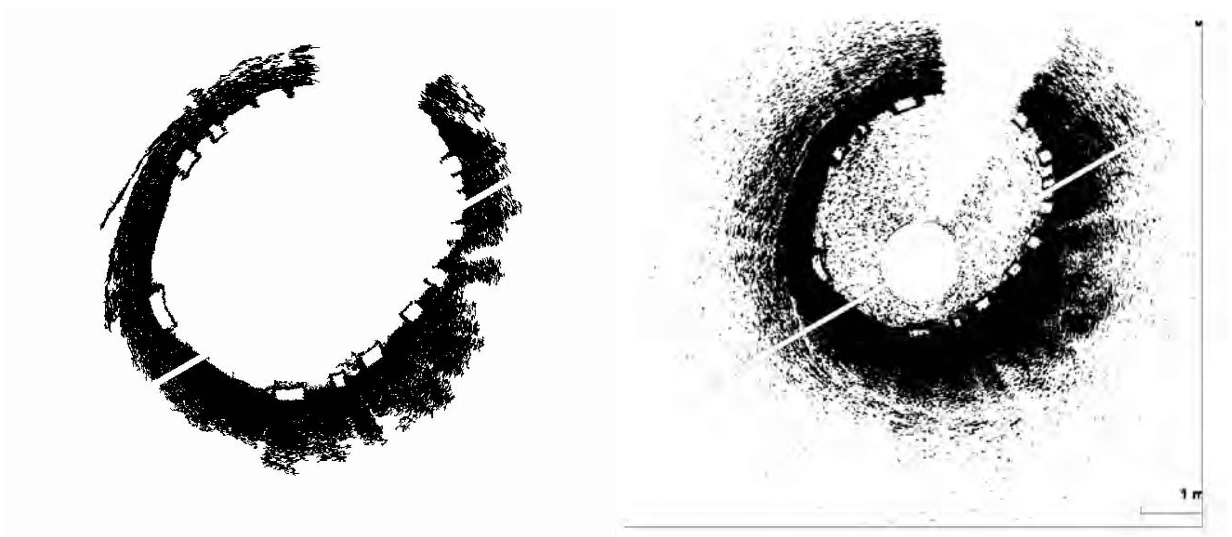
Figure 7: Circle and line cut

2.3 Bright Region

One important observation is that the blood vessel wall is always the most bright area in an OCT image. By detecting the bright region, we can precisely extract the wall and thus the stents nearby into the region of interest. First, the original image is transferred into the gray scale. Then the grayscale image is changed to a binary image using the built-in function im2bw(). im2bw() takes in a grayscale image and a threshold. The threshold is automatically derived by the MATLAB® function graythresh().[11] The input of the function should be a grayscale image.

We can manipulate the grayscale image by MATLAB® function imadjust(GRAY_CROP,[0

0.5],[0.5 1],1). The function has four input parameters and returns the adjusted grayscale image as a variable. The first parameter is the image variable that a user wishes to adjust. Here, it is GRAY_CROP. The next two parameters are the range of intensity values from the input image GRAY_CROP and the the range of intensity values the user wishes to map them to for the output image. For example, in our code we wish to map the values in the range 0 to 0.5 to values in the range 0.5 to 1. These ranges were chosen through a trial and error approach in order to adjust the image to a brighter contrast and bring out finer details that were previously lost due to lower intensity values. The final parameter is a gamma value that specifies the shape of the curve used to map the input intensity values to the output intensity values. If gamma is equal to 1, the curve is linear. If gamma is less than 1, the curve is weighted towards higher intensity values. If gamma is greater than 1, the curve is weighted towards lower intensity values. A high gamma value helps to keep only the most bright part and discard the rest. The resulted image eliminates all the noise and focuses exclusively on the vessel wall region. Figure 8 compares the binary image of a high gamma value with that of a low gamma value.

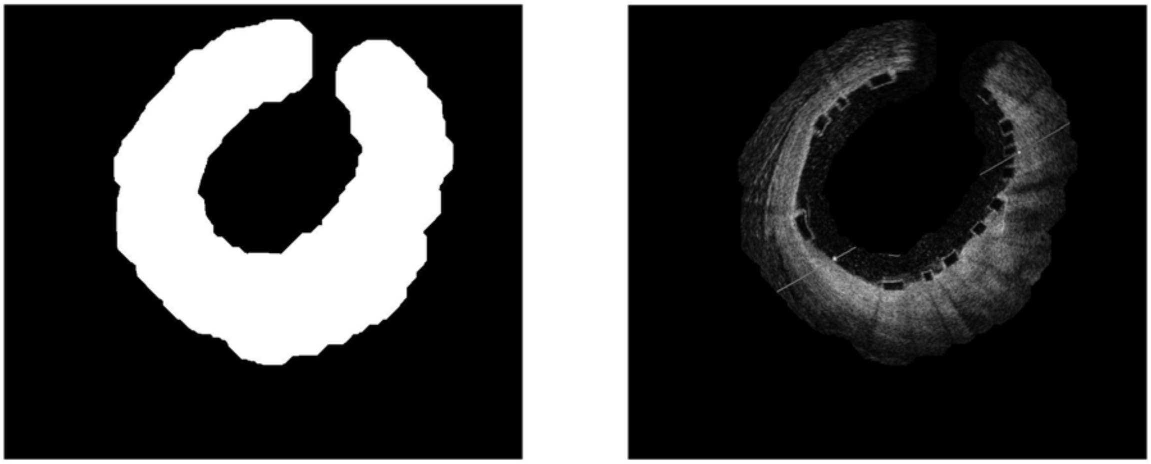


(a) High Gamma Value (here used)

(b) Low Gamma Value

Figure 8: Binary Image

Based on the binary image so far, we want to build a mask to form a region of interest (ROI) for our stent detection. Because the first time detection is relatively accurate and devoid of error, the black part can be increased as large as possible to enclose more area into our ROI. The MATLAB$^{®}$ function strel() creates a morphological structuring element. It can be used as a handle for the built-in function imerode() to expand the binary image. The first parameter of strel() is the shape of the structuring elements. Our code uses 'disk' because it is most suitable for the vessel's shape. The second parameter specifies the radius. Here we use a relatively large number, 36 through a trial and error approach. The region is then inverted to form a white ring which is used as the mask for the grayscale image.



(a) The binary ring mask

(b) Masked grayscale image

Figure 9: Mask of ROI

3. Detection Algorithm

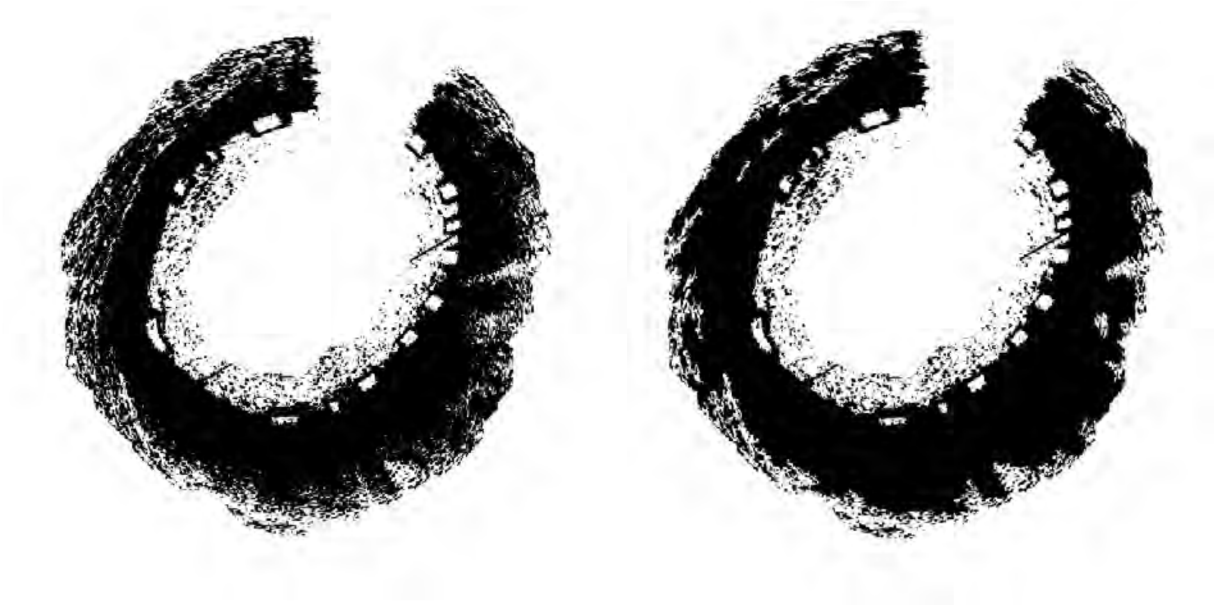
3.1 From RGB to Binary

In an RGB image, each pixel contains three sets of information: hue, saturation and intensity values. Hue values are the property of light by which the color of an object is classified as red, green, or blue in reference to the color spectrum. Saturation values are the colorfulness of a color. A high saturation value means the color looks bright whereas a low value means the color appears dark. Intensity values are given in a range from 0 to 1, where 0 represents a complete absence of light(black) and 1 represents a complete presence of light(white).

RGB image contains abundant information which makes the graph vivid. However, when it comes to auto-detection, the majority of the information is redundant. We can simplify the problem by converting a RGB image to a binary image in which each pixel is either 1(white) or 0(black). The simplification reduces to a great extent the uncontrollability of the diverse parameters in processing the image data obtained from assorted sources of patients.

In the pre-processing stage, a masked grayscale image called GRAY_BOUNDARY has already been generated. We can use imadjust() and im2bw() on it directly to construct a masked binary image. We found through trial and error approach that an input intensity range from 0 to 0.5, an output intensity range from 0.8 to 1 and a gamma value of 1 brings out the best mapping for the image recognition purposes in the first time detection. The new image variable ADJUST_NOISE is the adjusted grayscale image that is converted to a binary image BW_NOISE_N in the next step. Now the stent outlines have intensity values of 1 so for the algorithm to recognize them, we need to invert the image and give the values of 0. The ∼ symbol inverts all of the pixels' intensity values turning 1's into 0's and 0's into 1's.

Furthermore, the bwareaopen() function is used to reduce background noise in our image. It takes in a binary image and a threshold number. It uses this threshold to convert any grouping of contiguous white pixels, whose total number is less than the threshold, to black. Our code uses the inverted binary image BW_NOISE, and a threshold of 50 pixels. Thus any white areas composed of fewer than 50 pixels will be converted to black. We know that a stent structure's interior will generally be composed of more than 50 pixels so any object composed of fewer pixels is unnecessary for our computations and is just background noise.



(a) Before noise reduction

(b) After noise reduction

Figure 10: Noise reduction

3.2 Edge Detection

At this point we now have the image in the format of binary and can apply the MATLAB® boundary finding function bwboundaries():

[boundary_stent,region_stent] = bwboundaries(BW_FINAL, 'noholes');

It uses a binary image as its input parameter. The image must have the pixels of objects contain the value 1 and background pixels contain the value 0. In our code, this is the finalized inverted binary image BW_FINAL. The function then returns two matrices containing the boundary information. The first matrix is a cell array with Z cells, where Z is the number of objects found in the image. Each cell contains a Y-by-2 matrix where each row contains the row and column coordinates of a boundary pixel. Y is the number of boundary pixels found for an object. In the code this matrix is designated as *boundary_stent*. The second matrix is a label matrix, which is designated as *region_stent*. This is a two-dimensional array of nonnegative integers that represents contiguous regions in the image. Region k is the set of all elements of *region_stent* that have the value k. This array is then used in the next step to compute the properties of the objects that were found. The 'noholes' parameter of this function is used to specify for the algorithm not to search for objects within other objects since there will be no stents inside other stents in our images.

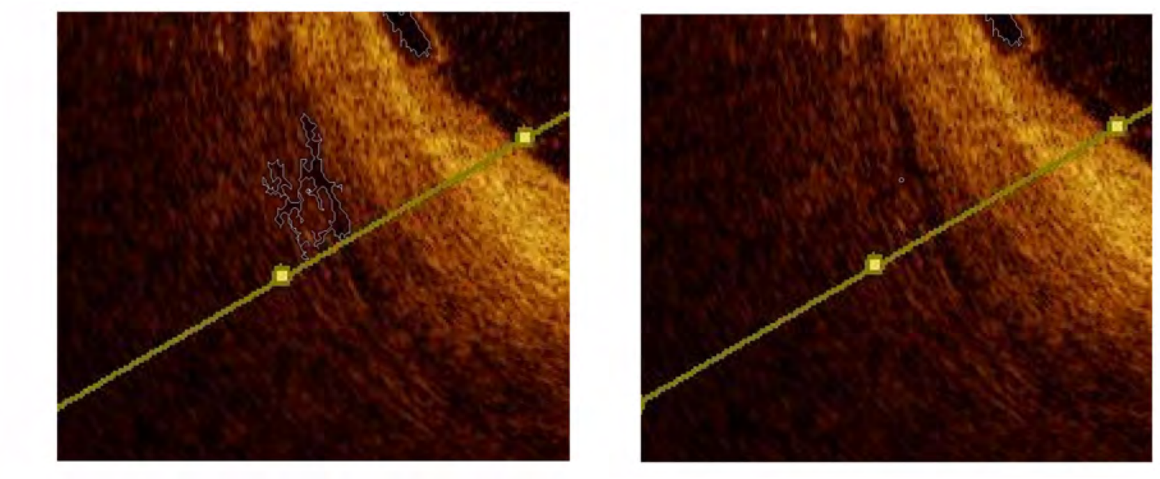
The original *boundary_stent* captures the outline of the entire image as its first object so we need to remove it via the instruction:

```
boundary_stent = boundary_stent(2:length(boundary_stent));
```

Our code plots the outline of each labeled object which is neither too large nor too small and returns the array of the outline's coordination. This is realized by a simple for loop:

```
for i = 1 : length(boundary_stent)
        G = boundary_stent{i};
        if length(G) <= 250 && length(G) > 10
                X = G(:,2);
                Y = G(:,1);
                plot(X, Y);
        end
end
```

Any object with an edge longer than 250 or shorter than 10 will be eliminated from our potential candidates. From Figure 6, it can be discovered that the code is helpful in removing some wrong candidates but can not eradicate them. Moreover, by examining the whole picture, as shown in Figure 7, there still exist numerous errors which cause the severe inaccuracy of the auto-detection. This can be solved in the error elimination process.



(a) Before elimination

(b) After elimination

Figure 11: Edge Elimination

3.3 Centroid Plot

MATLAB$^{®}$ offers a built-in function regionprops() that takes a label matrix, *region_stent*, and computes a set of properties about each labeled object:

STATS = regionprops(region_stent, 'all');

The function returns the properties in a P-by-1 array, STATS, where P is the number of objects labeled in *region_stent*. The property that is concerned here is the centroid of the object, which will be used as the location of the stent structure for the creation of the 3D geometry. It can be attained by the following for loop:

```
for i = 2 : length(STATS)
          centroid = STATS(i).Centroid;
end
```

Again, the first element of STATS contains the properties of the whole image so it is necessary to skip it. From Figure 12, it can be observed that besides the centroids of the correct objects, copious wrong results are plotted as well. In the next critical stage, all the errors can be automatically rejected.

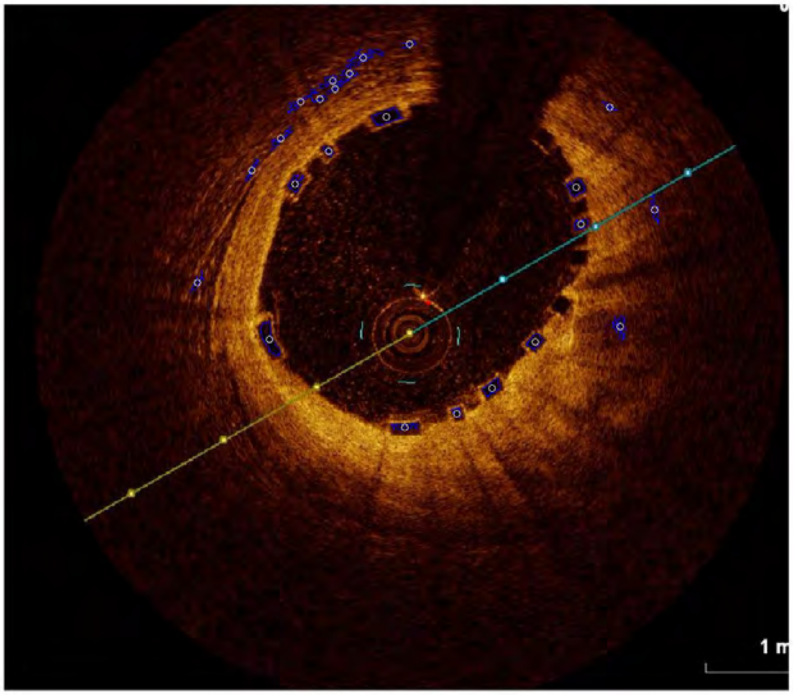


Figure 12: Centroids from first time detection

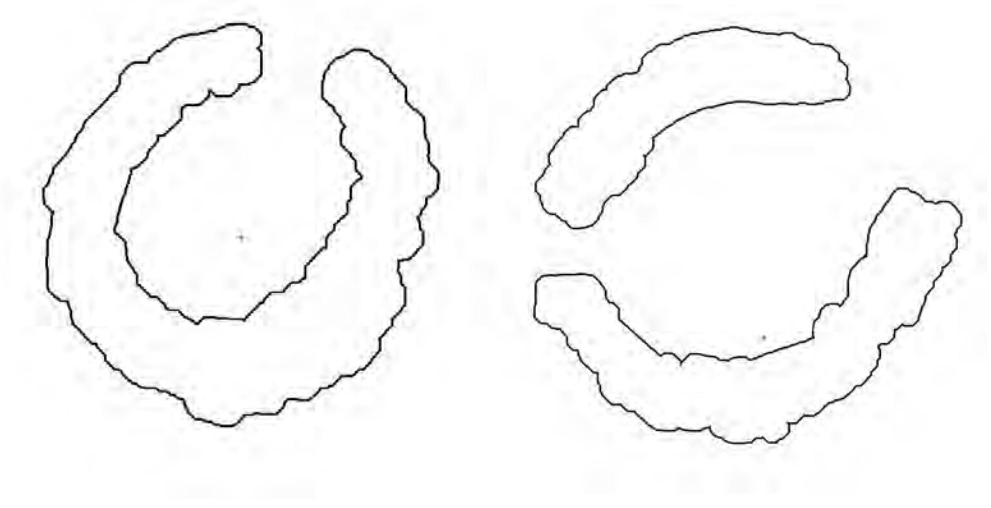
4. Error Elimination

4.1 Relative Distance

From Figure 12, it is clear that all objects of the stent structure are attached to the vessel wall. If the wall is expanded towards inside slightly to include all the stent objects, these objects are the nearest to the expanded inner vessel wall. All the wrong objects in the figure are comparably far away from the inner wall. Thus we can compute the distance between the centroid and the wall and eliminate those centroids whose distances are relatively longer than others'. We define the distance to be the relative distance and the distance between the centroid and the center of the vessel to be the absolute distance.

The trivial expansion of the vessel wall can be realized by the same method discussed in the pre-possessing, imerode(). But this time, a small number is used for the radius in the function strel(). Furthermore, the difference between a smaller eroded region and a larger one can generate an artificial binary vessel wall area(Figure 13(a)). Our code uses number 15 and 18 in strel() to form such two eroded regions. The consequent man-made wall can be constructed via the MATLAB® function imshowpair(A, B, method). The first two parameters are the two eroded regions and the third is 'diff' which displays the difference of A and B. The thickness of the wall is perfect to make the relative distance stable and reliable.

The absolute distance depends on the choice of the center of the vessel. In a normal OCT image of the blood vessel, the artificial wall is a integral part. It is reasonable to attain the center of the wall as the center of the vessel. However, due to the irregularity and diversity of the sources of the images, some walls may be divided into several parts(Figure 13(b)). This brings the uncertainty to using the center of the artificial wall. Indeed, the code can be modified to check whether the wall is an integral or not but the process is really verbose compared to another way. We recall that the center position of the arthroscope in one set of images remains unchanged. Fortunately, this center position is always inside the blood vessel. Consequently, to simplify the problem, it can be assumed to be the center of the vessel for the computation of the absolute distance and later the relative distance.



(a) Integral artificial vessel wall (b) Two-parts artificial vessel wall

Figure 13: Artificial vessel wall

In our code, the relative distance is the overlapped part of the absolute distance within the slightly expanded vessel wall. To compute it, numerous points are drawn along the line between a centroid and the center of the wall. In figure 14, these points form several red lines. The first point reaches the artificial wall from the center is recorded as a small red circle in the graph. The relative distance is then calculated between the point and the centroid. The choice of the center may affect the absolute distance. However, as long as the chosen center is inside the expanded wall, the relative distance just suffers minor changes which can be ignored. Centroids with long relative distances are abandoned. We decide to use 45 as a threshold through the trial and error approach to keep only the centroids of the strut.

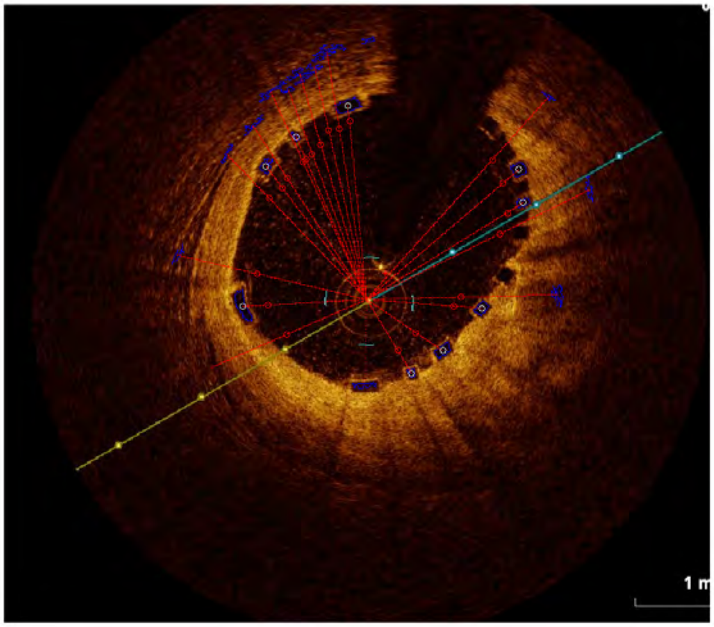


Figure 14: Distance from center to centroid

4.2 Center Modification

In figure 14, the strut directly under the center has not been detected. In figure 15, we zoom in that part and change the point color to white to make it more clear. This is because the x or y coordinates of the center and the centroid are too close to each other which causes loss of precision and the sparseness of the points. The group of points may skip the artificial wall so that no intersection point is found.

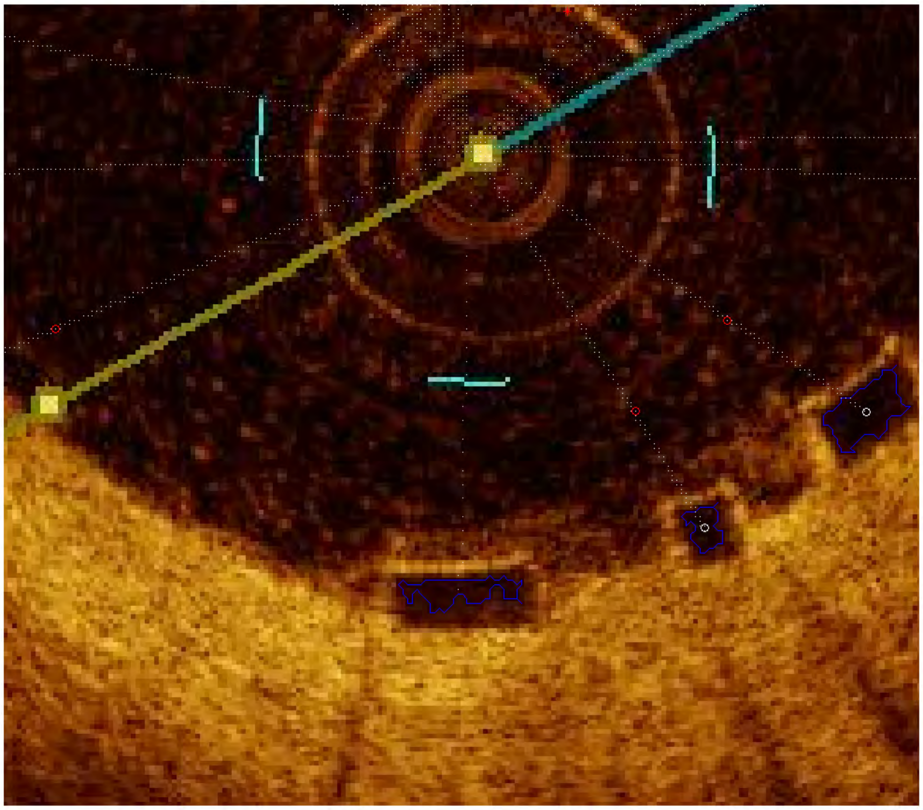


Figure 15: Failure of the method

Once the problem is discovered, it can be solved by changing the center position and increasing the number of points. Since the relative distance barely changes due to such procedures, the aforementioned threshold is still valid to produce the same result.

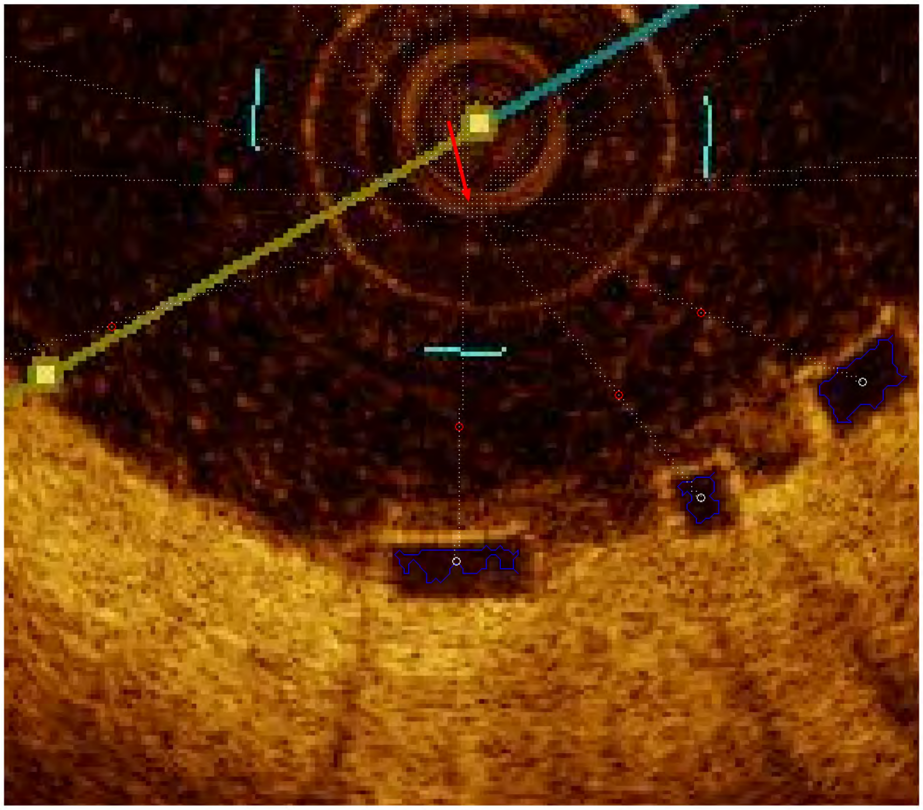


Figure 16: Amended result

5. Second Detection to Obtain More Struts

As may be noticed in figure 14, there are several struts not successfully being detected. In order to obtain more struts, a second detection is required. First, those already detected should be "white" out in the masked grayscale image before it is transfered into binary because they should not be detected again (Figure 17). The white color is used here since all the bright areas will be black after the transformation.

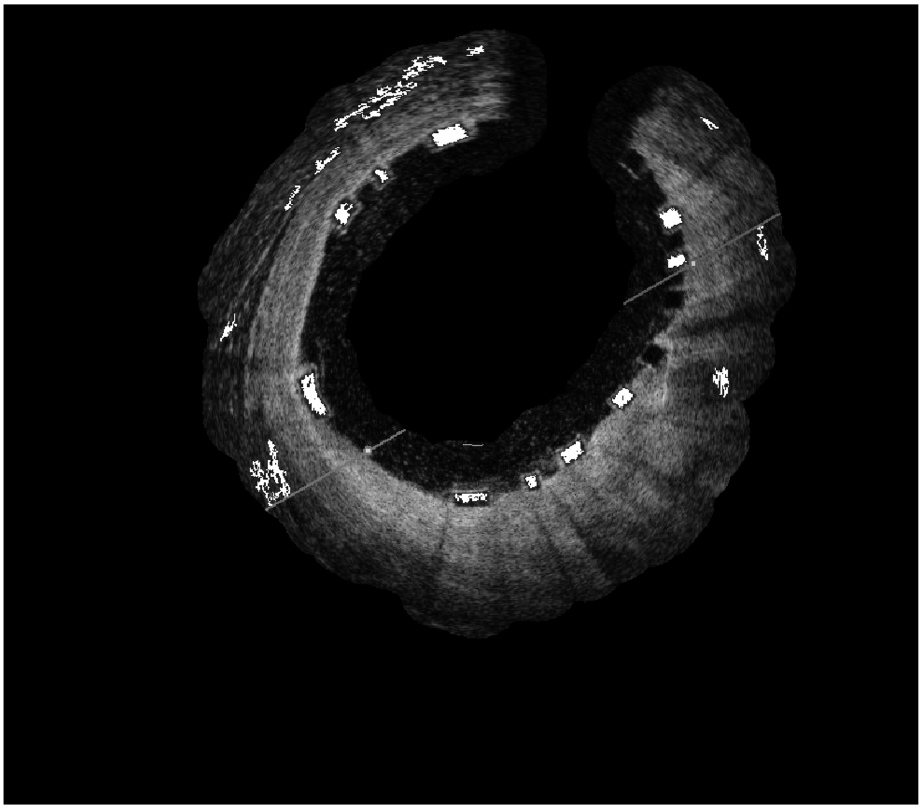
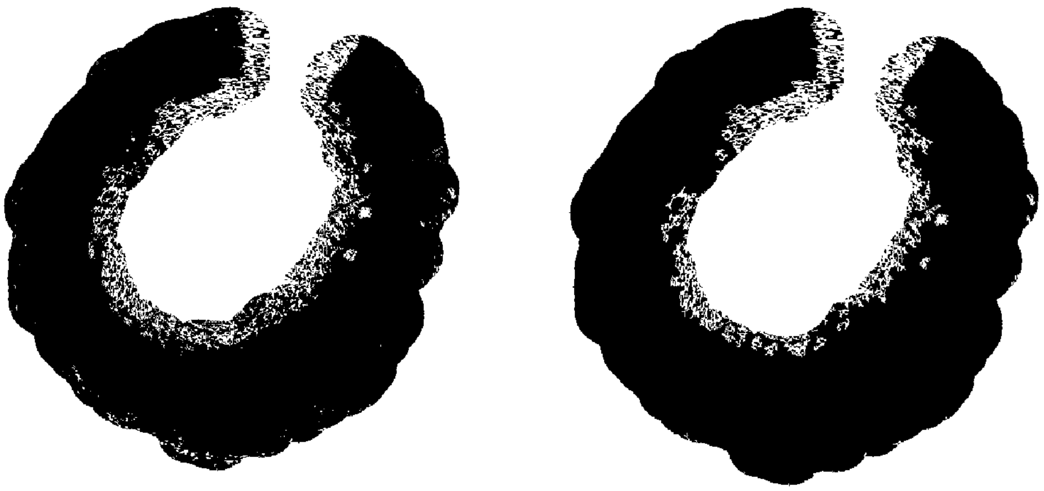


Figure 17: "Whited" out grayscale image

For the binary image conversion, a new set of inputs is used in function imadjust() to bring out more obscure parts. More darker areas are transfered into black so that the ambiguous strut boundaries can be catched. Through a trial and error approach, an original intensity range from 0 to 0.5, an output intensity range from 0.5 to 1 and a gamma value of 0.4 are the most suitable. As in the first detection process, the bwareaopen() function is used to reduce background noise in our image. Again, a threshold of 50 pixels is applied here.



(a) Before noise reduction

(b) After noise reduction

Figure 18: Noise reduction

As we may see from figure 19, although the three struts on the upper-right part which were not detected at the first time are discovered now, more erroneous edges are brought out as well. Most of these incorrect edges are more difficult to remove by the elimination procedure discussed above because they have the same property as the correct ones, alongside the vessel wall. Hence, new elimination methods should be designed.

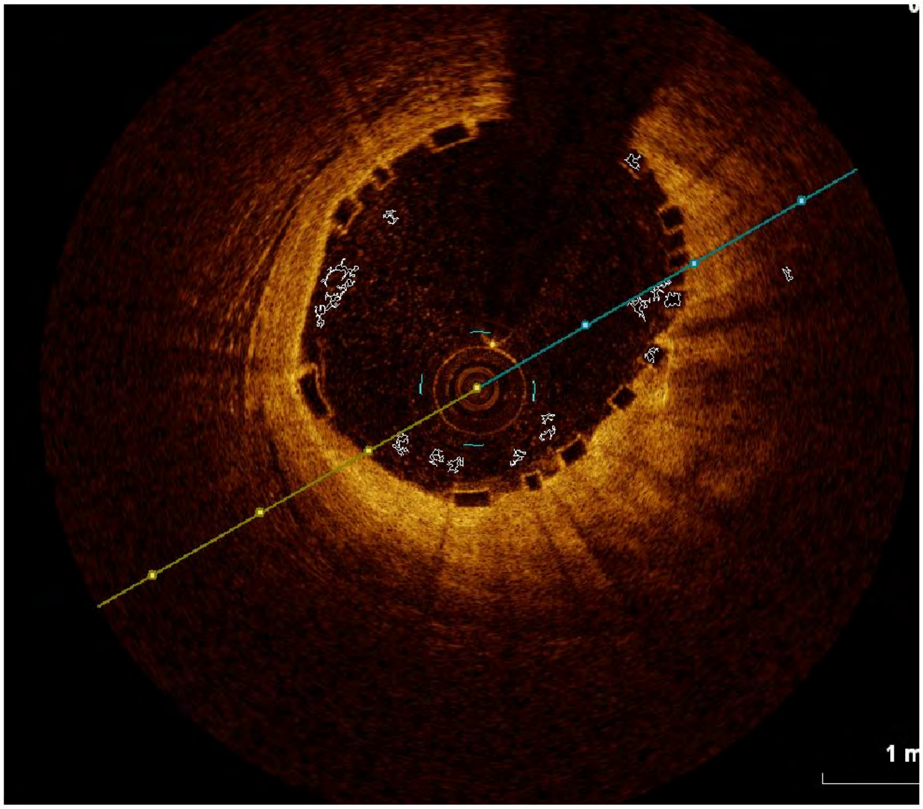


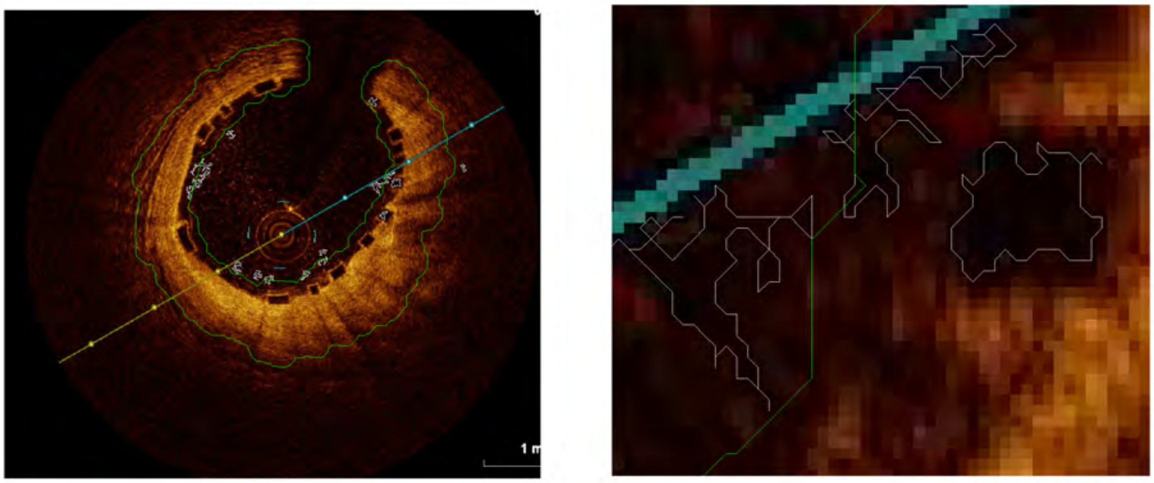
Figure 19: Edges from second time detection

6. Final Error Elimination

6.1 Covered Edges

By examining more details from figure 19, we may find that the erroneous edges are slightly farther from the vessel wall than the correct ones. Especially, if we check the farthest point of these edges, there exists a comparably more obvious discrepancy between the incorrect one and the valid ones. Thus, we can expand the vessel wall to some extent to cover all the correct edges while keeping out part of the unwanted ones. The expanded wall can be built by the boundary of a binary image. Based on the image in Figure 8(a), we use imerode() function again to enlarge it. Through test, we find strel('disk', 15) is the fittest handle for imerode() here.

In figure 20(b), three examples are provided. The left edge is totally beyond the boundary. The center edge crosses over the boundary. The right edge, also the correct one, is wholly covered by the boundary. Only those covered by the boundary are reserved as the candidates for the struts. The rest are discarded as errors. In figure 21, we can see that only the edges of the three struts on the upper-right part are left after the elimination.



(a) Whole picture of the boundary

(b) Detailed picture of the boundary

Figure 20: Expanded wall's boundary and edges

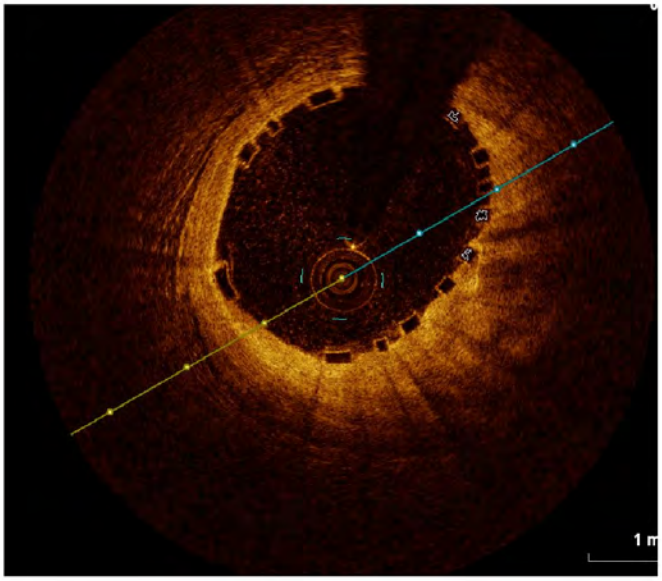


Figure 21: After elimination

6.2 Nearby Area

In some images, there may be still some errors unremoved. In figure 22, the red points are erroneous centroids after all the aforementioned elimination processes. A further elimination method is to compare the nearby area of the detected centroid in a binary image (Figure 23) created in section 2.3.

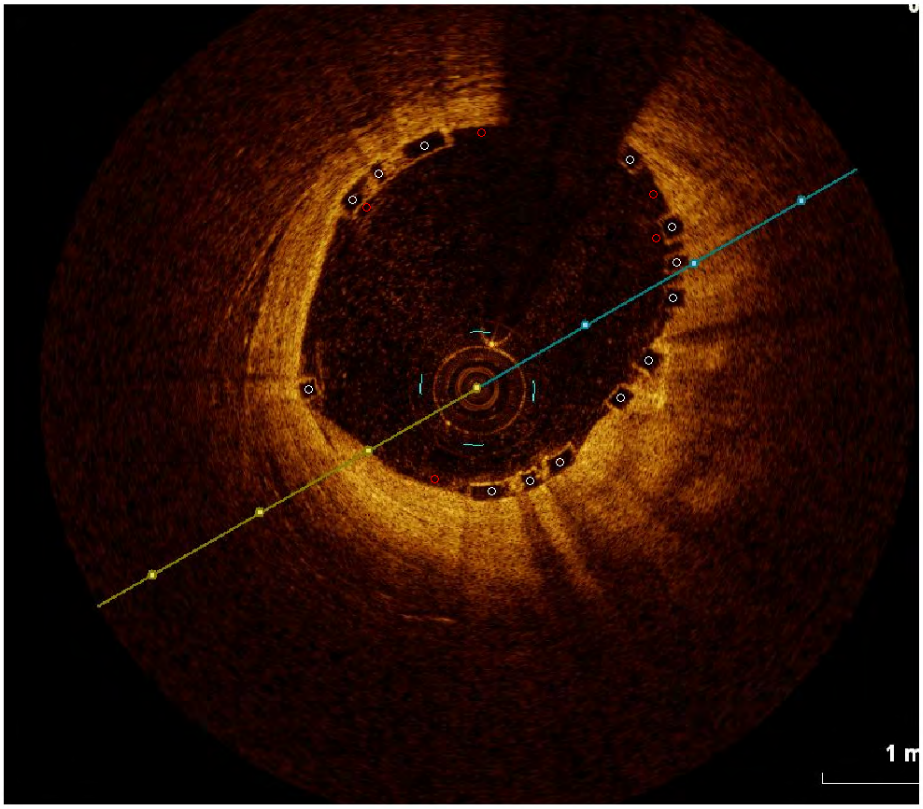


Figure 22: Errors represented by red points

In figure 23, one important observation is that the nearby areas of the red points, particularly those towards the inner side are lack of white pixels compared to the right points, here the green ones. Consequently, by setting a threshold of the white pixels in such areas, we can effectively remove some of the errors. The yellow parts are the white pixels enclosed in the nearby areas of the points. We see near most red points, four out of five, there are few yellow regions, if exist.

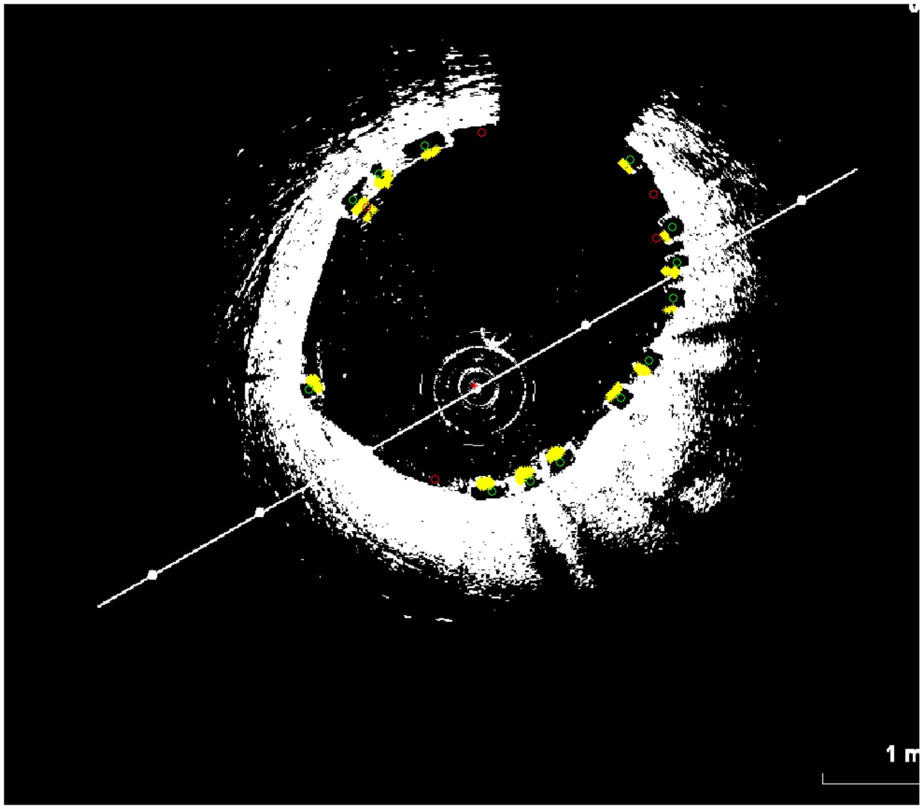
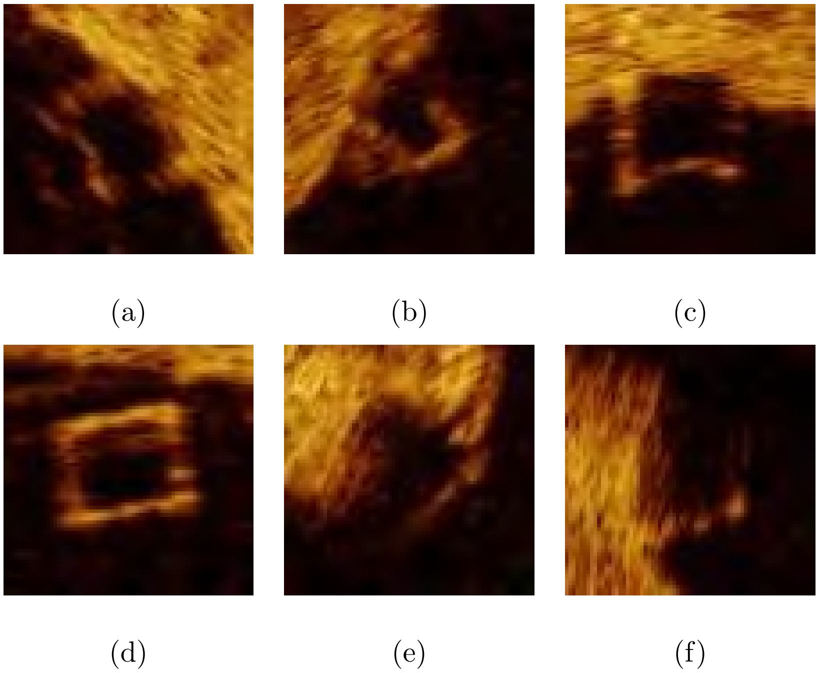


Figure 23: Nearby area error elimination

7. Manual Patching

The condition of an OCT image varies widely which causes algorithms with static parameters impossible to detect all the struts in the data set. We have to manually record the missed struts to generate a 100% accurate result for 3D reconstruction. Importantly, when we record a strut, a small image is saved to a "missed struts" image date pool. The data pool is used for the future machine learning algorithm to automatically find all the struts position in the whole OCT images data set.



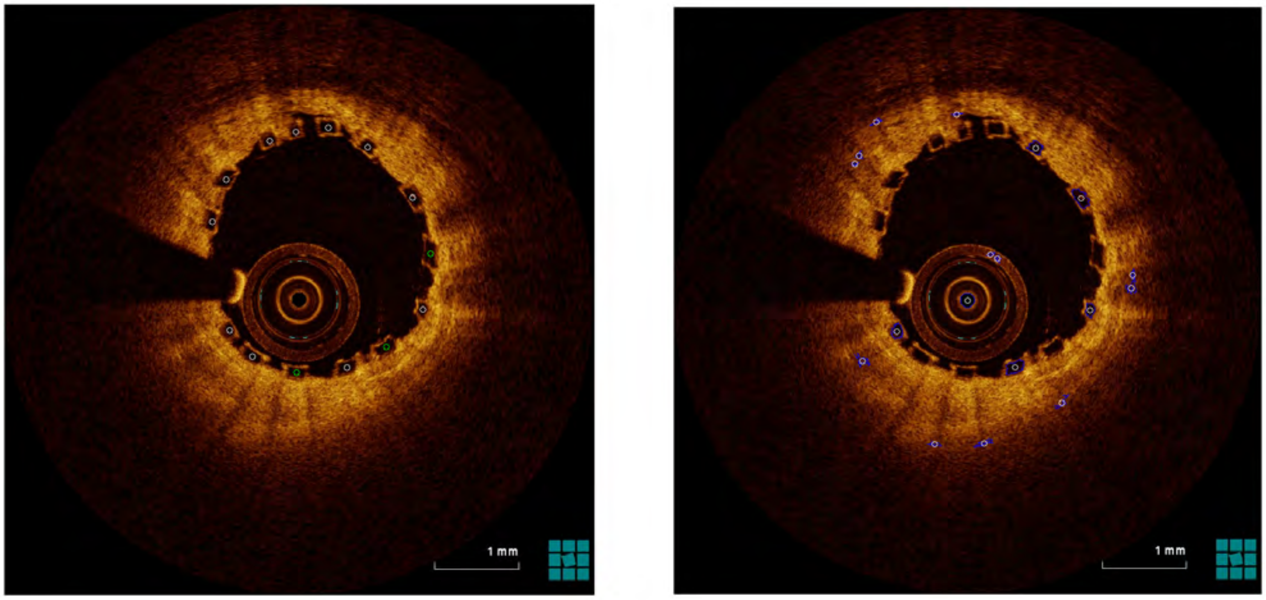
(a) (b) (c)

(d) (e) (f)

Figure 24: Examples of missed struts

8. Results

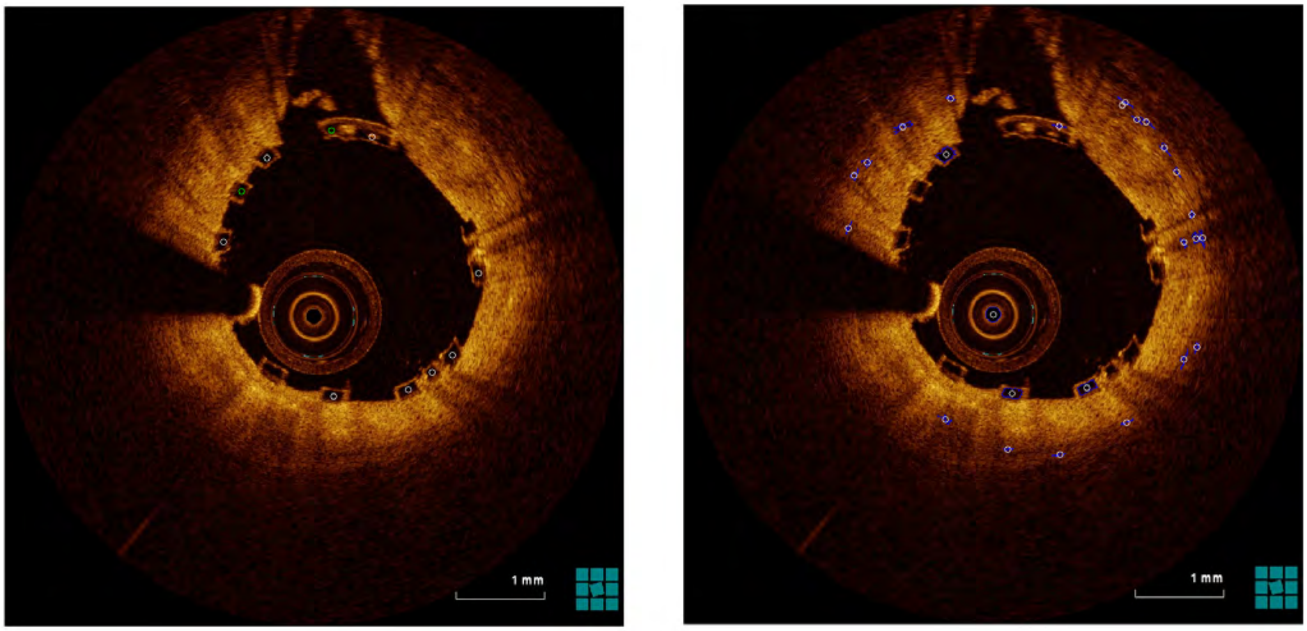
Here we show three pairs of pictures to briefly compare previous work with ours to measure the improvement.



(a) Our result

(b) Previous result

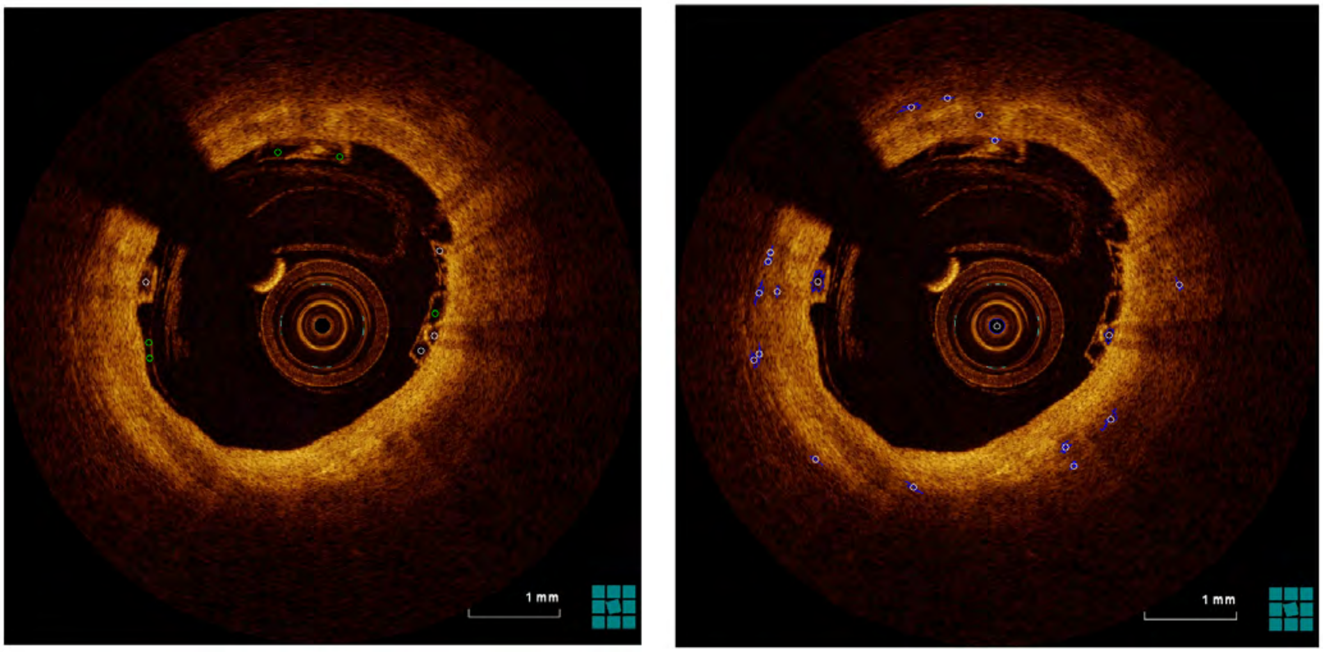
Figure 25: Pair 1



(a) Our result

(b) Previous result

Figure 26: Pair 2



(a) Our result

(b) Previous result

Figure 27: Pair 3

The white points in pictures of our algorithm are the struts detected at th first time. The green points are the ones found during the second time. In pair 1, our algorithm succeeded in plotting all the struts without error. In pair 2, our algorithm missed two struts without error. In pair 3, all struts were detected but with two errors. Most our results are like pair 1 and pair 3 where errors occurred is really a rare case. On the other hand, in all pictures from the previous algorithm, a limited number of struts were discovered. Moreover, multiple errors were incurred at the same time. To further confirm the improvement, we constructed a

3D point cloud with our results in a full image data set (Figure 28). Since Buckley's method provides many errors, it is meaningless to built a comparable 3D point cloud by his code here. Our algorithm automatically detected 564 correct struts with 105 struts missed. The accuracy rate computed by the ratio of the detected struts to the total struts is 84.30%.

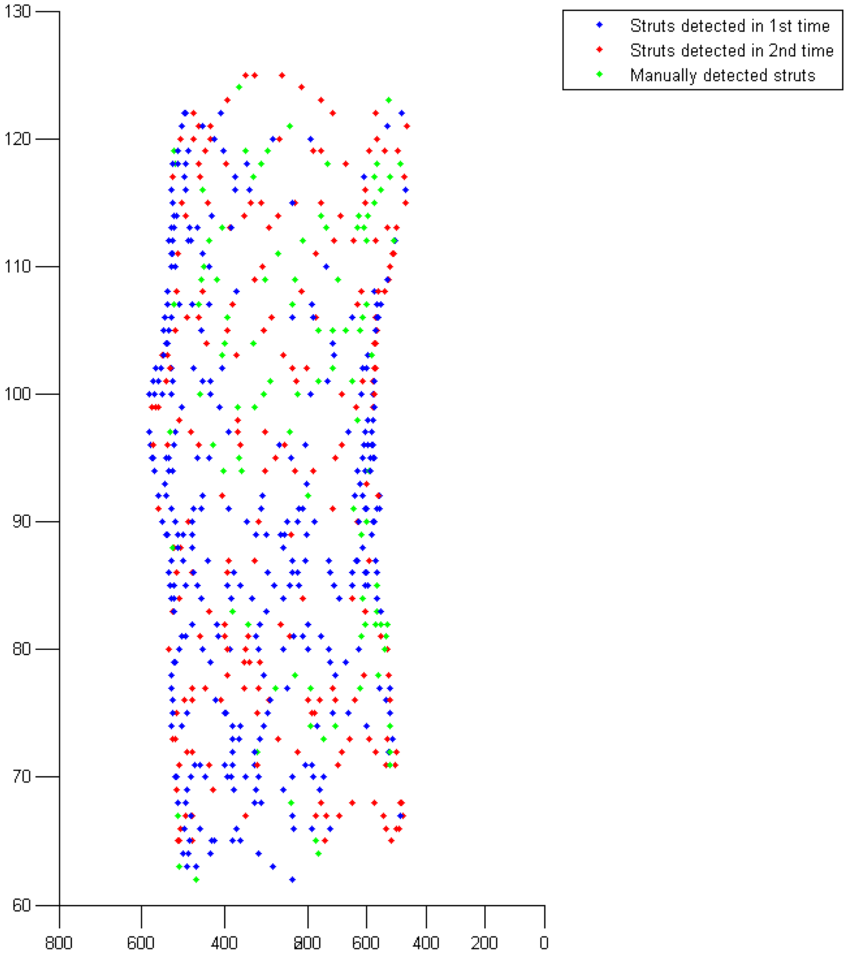


Figure 28: 3D point cloud

9. The Usage of The Strut Points in Stent Reconstruction And Blood Flow Simulation

In the project of Boyi Yang from Emory University, he is working on reconstructing the 3D geometry of the stented vessel lumen to quantitively simulate fluid dynamics inside it and access the relationship between wall shear stress and neointimal thickness after stenting. Without the stent geometry, true local hemodynamics conditions around the strut are unavailable and research result would be very weak (see Figure 29). Part of Boyi's work is focused on using the point cloud to automatically generate a full stent structure. Then the computed stent will be combined with the vessel lumen to produce a fluid dynamic simulation. The simulation can provide a more reliable and useful tool to analyze the risk of the stent and other concerned problems.

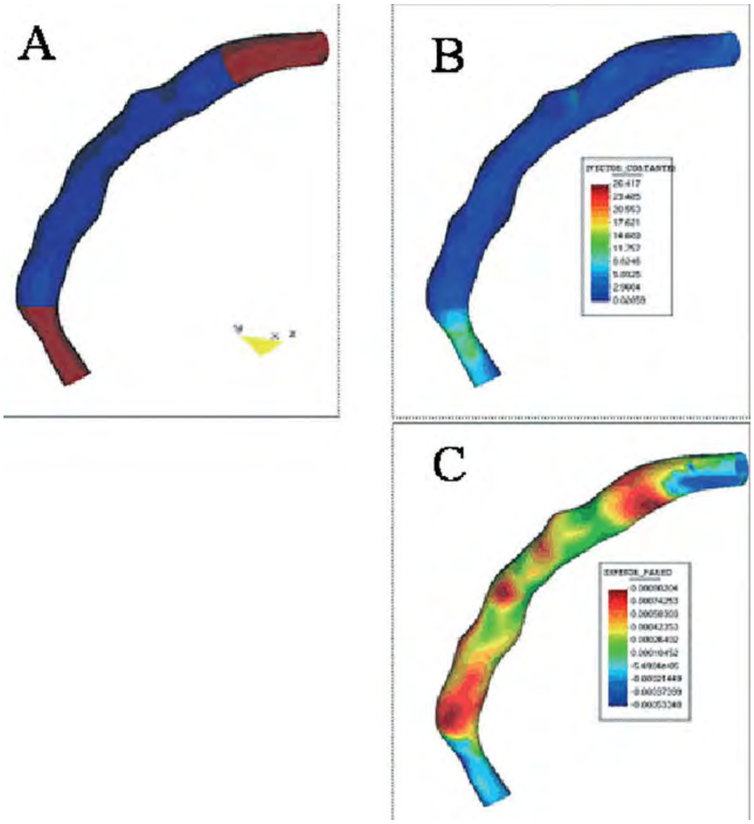


Figure 29: Fluid simulation without stent[12]

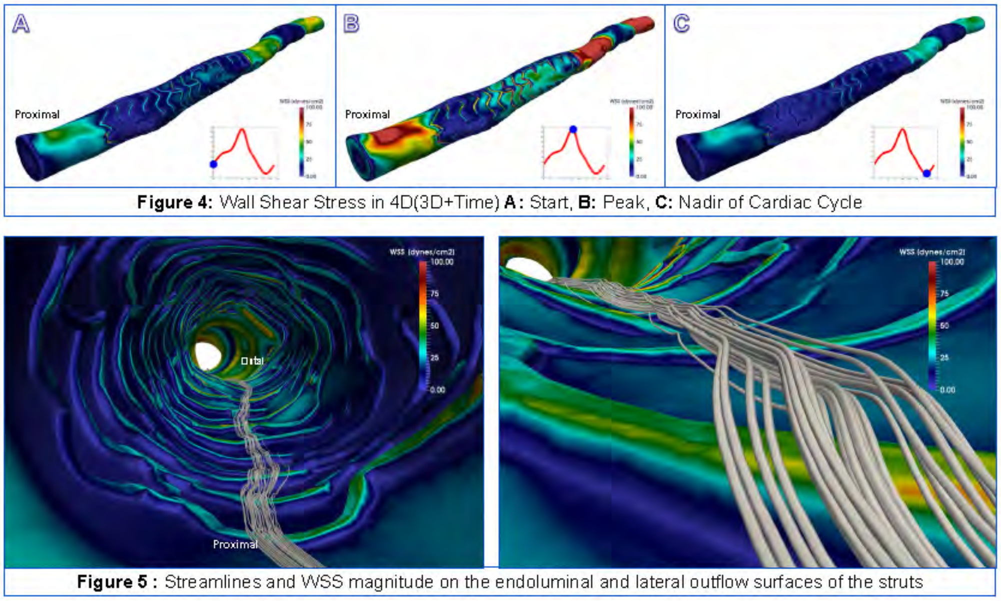


Figure 4: Wall Shear Stress in 4D(3D+Time) **A:** Start, **B:** Peak, **C:** Nadir of Cardiac Cycle

Figure 5 : Streamlines and WSS magnitude on the endoluminal and lateral outflow surfaces of the struts

Figure 30: Fluid and WSS simulation with stent

10. Reference

[1] Peter N. Ruygrok, Patrick W. Serruys. Intracoronary Stenting From Concept to Custom. Circulation. 1996;94: 882-890.

[2] Charles T. Dotter, Melvin P. Judkins. Transluminal treatment of arteriosclerotic obstruction. Circulation. 1964;30: 654-670.

[3] Morton Kern. Bioabsorbable Stents – Where Are We Now? Cath Lab Digest. 2012;Volume 20 - Issue 6.

[4] John A. Ormiston, Patrick W.S. Serruys. Contemporary Reviews in Interventional Car-

diology: Bioabsorbable Coronary Stents. Circulation. 2009;2: 255-260.

[5] Huang D, Swanson EA, Lin CP, Schuman JS, Stinson WG, Chang W, Hee MR, Flotte T, Gregory K, Puliafito CA, Fujimoto JG. Optical coherence tomography. Science. 1991;254:1178–1181.

[6] Fercher AF, Hitzenberger CK, Drexler W, Kamp G, Sattmann H. In vivo optical coherence tomography. Am J Ophthalmol. 1993;116:113–114.

[7] Schmitt JM, Knuttel A, Yadlowsky M, Eckhaus MA. Optical coherence tomography of a dense tissue — statistics of attenuation and backscattering. Phys Med Biol. 1994;38:1705–1720.

[8] Garret T. Bonnema, Kristen O. Cardinal, James B. McNally, Stuart K. Williams, Jennifer K. Barton. Assessment of blood vessel mimics with optical coherence tomography. J. Biomed. Opt. 12(2), 024018 (March 30, 2007). doi:10.1117/1.2718555

[9] Gurmeric S, Isguder GG, Carlier S, Unal G. A new 3-D automated computational method to evaluate in-stent neointimal hyperplasia in in-vivo intravascular optical coherence tomography pullbacks. MICCAI. 2009;12(Pt 2):776-85.

[10] Nico Bruining, Kenji Sihan, Jurgen Ligthart, Sebastiaan de Winter, Evelyn Regar Erasmus MC, Thoraxcenter, Rotterdam, The Netherlands. Automated Three-Dimensional Detection of Intracoronary Stent Struts in Optical Coherence Tomography Images. Computing in Cardiology, 2011;221-224

[11] Otsu N. A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man, and Cybernetics. 1979;Vol.9, No.1, pp. 62-66.

[12] Sanmartín M, Goicolea J, García C, García J, Crespo A, Rodríguez J, Goicolea JM.,2006. “Influence of shear stress on in-stent restenosis: in vivo study using 3d reconstruction and-computational fluid dynamics”. Rev Esp Cardiol. 2006 Jan;59(1):20-7.