

A Study of CT-based Visualization and Analysis of White Matter in Rat Brain

Author:

Heng Wang
s161101

Supervisor:

Anders BJORHOLM Dahl
Carsten Gundlach
Henning Friis Poulsen
Matteo Busi
Ulrik Lund Olsen



Kongens Lyngby 2018

Technical University of Denmark
Department of Applied Mathematics and Computer Science

Abstract

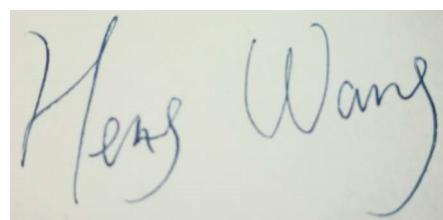
Animal brain is a very common research object in many fields. In order to understand the microstructure of it, numerous studies have been implemented, resulting a great number of visualization approaches. This research attempted to visualize the microstructure of white matter from rat brain with X-ray computed tomography techniques. To optimize the quality of the projections and reconstruction results, methods for processing the projections and reconstructed images are applied in this research. The bad pixel detection and correction, stripe deletion and off-center correction works well in optimizing the projections. Gamma correction and gray stretch could upgrade the contrast of the reconstructed slices. The image segmentation techniques are also discussed and examined in our project, and we have successfully visualized the 3-dimension structure of vessels and a brain tissue which are extracted from the volume of white matter. Finally, in order to find out the characterization of the vessels, we applied a structure tensor method for analyzing the vessel orientation, and analyzed the result we got. The results can be used to provide references about projection and reconstructed image optimization, to understand the microstructure of white matter from rat brain, and to offer suggestion to related future researches.

Preface

This thesis was prepared at DTU Compute in fulfillment of the requirements for acquiring an M.Sc. in Engineering degree at Technical University of Denmark (DTU).

This report represents 30 ECTS work from 22th Jan. 2018 to 22th Jun. 2018, which is carried out in DTU Compute and supervised by Anders Bjørholm Dahl, Carsten Gundlach, Henning Friis Poulsen, Matteo Busi and Ulrik Lund Olsen.

Lyngby, 22-June-2018

A handwritten signature in black ink on a light blue background. The signature reads "Heng Wang".

Heng Wang

Acknowledgements

I would like to thank my thesis supervisor, Anders BJORHOLM DAHL, Professor MSO, Head of Section in DTU Compute. He is always welcoming me, squeezing time from his heavy work, giving me answers and suggestions whenever I ran into a trouble spot or had a question about my project. He consistently allowed this project my own work, but ensure me running in the right direction whenever I'm missing.

I would also like to thank all co-supervisors in the project team, for their patience and support in overcoming numerous obstacles I've been faced in the project. Henning Friis Poulsen, Prof., provided me a chance of doing this project, so that I can finish my master thesis. Carsten Gundlach, allowed me to use the high performance computers in the image center, so I am able to process the big data set of this project. Ulrik Lund Olsen and Matteo Busi, gave me a great number of suggestions about the projection preprocessing and reconstruction.

I sincerely thank my cooperator, Jinning Lyu, explained the physical theories involved in this project to me, made the background of the data set comprehensible to me. We achieved a lot together during this project, and it is impossible for me to do so much without her. My sincere thanks also goes to Jan Kehres, for consistently setting up and fixing the Housing machine, making every scan in this project possible; and Mariam Andersson, gave me a brief understanding of the rat brain sample.

Last but not the least, I would like to thank my family: my husband and my parents, for saving me from influenced by depression, supporting me spiritually throughout writing this thesis and my life in general.

Contents

Abstract	i
Preface	iii
Acknowledgements	v
1 Introduction	1
1.1 Research Motivation	1
1.2 Scope of this Project	3
1.3 Outline of this Thesis	4
2 Reconstruction and CT Image Processing	5
2.1 Reconstruction with Astra-Toolbox	5
2.1.1 Study of the Reconstruction and Toolboxes	5
2.1.2 Reconstruction Results	8
2.2 Image Enhancement	12
2.2.1 Study of the Image Enhancement Methods	12
2.2.2 Application in Match Head Images	14
2.2.3 Application in Brain Images	17
3 Projection Preprocessing	19
3.1 Method to Detect Bad Pixels	19
3.2 Method to Correct Bad Pixels	24
3.2.1 Median Filter	24
3.2.2 Calibration with Referenced Projection	25
3.3 Stripe Removal in Sinograms	27
3.3.1 Processing Results	30
3.4 Off-Center Correction	32
3.4.1 Problem Analysis	32

3.4.2	Simulation of Off-center Conditions	34
3.4.3	Application in our sinogram	36
4	Analyzing of 3D Volume	39
4.1	Image Segmentation Study	39
4.1.1	Thresholding	40
4.1.2	Contextual Segmentation	41
4.2	Brain Vessel Analysis	43
4.2.1	Deleting the background	43
4.2.2	Extracting the vessel	46
4.2.3	Visualizing the Vessel Orientation	48
5	Discussion	51
5.1	Achievements and Limitations	51
5.2	Future Work	55
5.3	Conclusion	56
A	Other Figures	57
	Bibliography	61

CHAPTER 1

Introduction

1.1 Research Motivation

Computed tomography(CT) is an important and widely used tool for medical imaging. Researches with X-ray CT images have contributed a great number of scientific achievements in the past decades. Imaging of the animal (including human) organs is also a critical process for biology and medical studies, considering the visualization of an organ or components in it helps a lot to the researchers who are willing to see the small parts, like tissues, clearly in a zoomed image or volume.

Animal's brain, which is a good object for the relative researches, is also being CT-imaged in a number of projects. In 2014, Sahar Ghanavati's team published a research [GLLS14], cast a perfusion procedure for imaging the mouse cerebral vasculature (the small vessel system in the brain) by X-ray micro-CT. In Ghanavati's project the mouse brain was stained by hematoxylin. Another project publish by M. Q. Zhang's team was looking forward to achieve a three-dimensional(3D) visualization of rat brain microvasculature, with a rat brain in a condition of permanent focal ischaemia by synchrotron radiation. Instead of the full microvasculature system, we haven't found the 3D imaging result of the small components, like vessels, in the white matter of rat brain. Thus, we decided to do a project to visualize the 3D structure of the white matter.

And in order to get a clear images of the brain sample, the study of the X-ray scan and projection processing is also a necessary point in our project. So for reaching the goal, which is visualizing the 3D structure of the white matter, finding good techniques to optimize the projections and reconstruction results would also be an important part of our project.

The sample we used is a white matter of a perfusion fixated rat brain, extract with a 1mm biopsy punch. The sample is stained by OsO₄. The X-ray CT scan is finished with the DTU PCXCT system and the Zeiss micro-CT system in DTU Imaging Center.

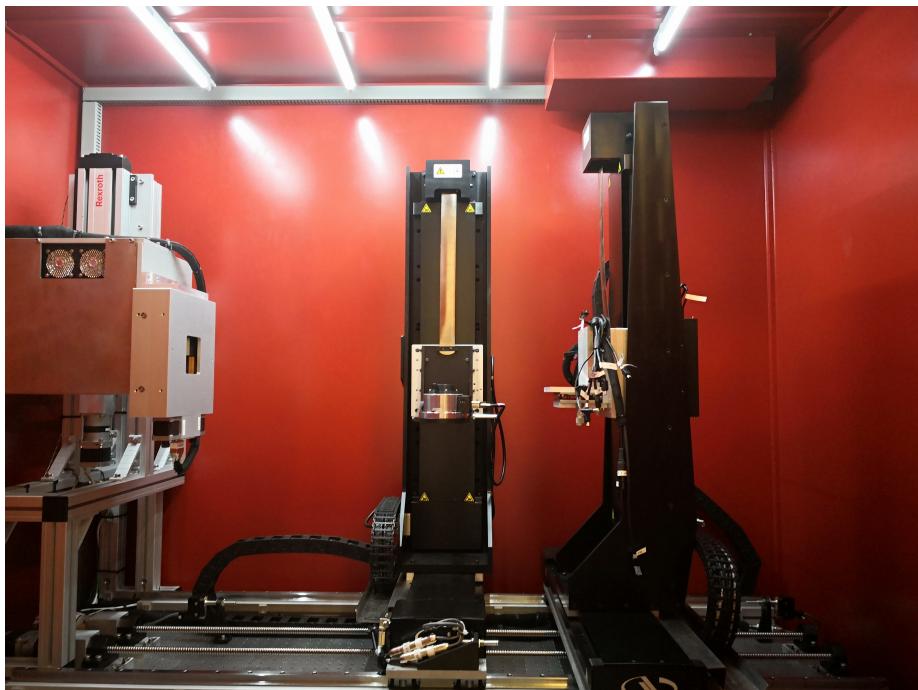


Figure 1.1: The DTU PCXCT system.



Figure 1.2: The Zeiss X-ray micro-CT system. This system gives much better result than DTU PCXCT system.

1.2 Scope of this Project

During our project, we firstly examined the DTU PCXCT system with a match head. In order to get the 3D volume, we studies some reconstruction tools, and finally we chose the Astra-Toolbox. After the first data set of projections are obtained, we found that there are bad pixels which would impact the SNR of the reconstructed images. In order to reduce such impacts we tried to find out method for detecting and correcting the bad pixels. The contrast of the reconstructed image was also not acceptable, so we studied and applied some image enhancement techniques in order to make the images clearer. Afterwards, we scanned the rat brain, and tried our optimization methods for the brain projections. However, with a long-period test, we found that it is hard to get images with an SNR making the image analysis possible. Thus, we scanned the brain sample with Zeiss scanner, then received projections and reconstructed images with hardly any noise. The new data set made the analysis of the white matter volume possible, so we segmented the volume, elected the biggest vessels, and attempted to analyze their orientation with the structure tensor method. After all we discussed the process and results of our projects, and summarized

the conclusions of this project.

1.3 Outline of this Thesis

The first chapter of this thesis gives a brief introduction of the motivation, the scope of this projection, and the outline of the full thesis. The second chapter shows our study of the reconstruction algorithms and toolboxes, the reconstruction results with the Astra-toolbox, and image enhancement efforts. The third chapter explains our attempts of optimizing the projections, including the bad pixel detection, bad pixel correction and the off-center correction. The fourth chapter introduced our study about the image analysis techniques for the white matter in rat brain, and the analysis results of the sample. Chapter 5 states the discussion of our project and results, discusses the achievements and limitations of our work, and gives an outlook of the future works.

CHAPTER 2

Reconstruction and CT Image Processing

2.1 Reconstruction with Astra-Toolbox

2.1.1 Study of the Reconstruction and Toolboxes

Reconstruction is an mathematical process, which means generating a set of tomographic images based on X-ray projections. A lot of reconstruction methods are found and developed, and they are used for different types of X-ray sources and scans. Generally, for 2D CT scan we are using parallel beam and fan beam. For 3D CT scan, parallel beam and cone beam are utilized. Considering DTU PCXCT system supports only cone beam scan, our study is mainly about the methods for 3D cone beam reconstruction.

[FDK84] and [NWX⁺10] introduced the process of cone beam CT scan, and the related reconstruction formulae. Figure 2.1 is a demonstration of a 3D cone beam CT system. The main components involved in the system are the source, the detector and the sample. The sample is placed between the source and the detector. During the acquisition, the source is emitting the X-ray, and the detector units are counting how many photons hit on them after being absorbed

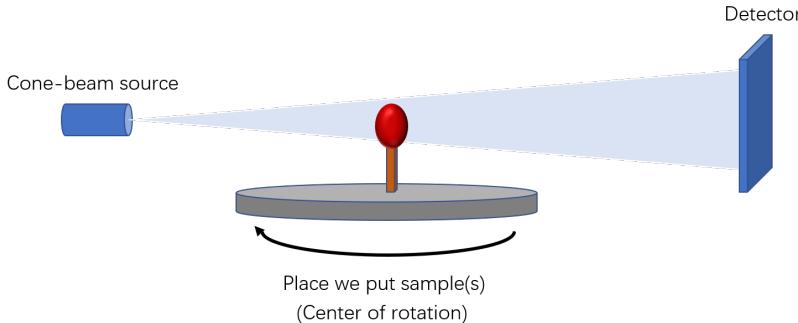


Figure 2.1: A demonstration of a cone beam CT system.

by the sample. And the sample rotates after each scan, in order that a set of projections from different directions are taken.

A lot of algorithms for 3D cone beam reconstruction are discovered, like Filtered Back-Projection (FBP), Feldkamp-Davis-Kress (FDK), Simultaneous Algebraic Reconstruction Technique (SART)[KSW02], Simultaneous Iterative Reconstruction Technique (SIRT)[Gil72], and Conjugate Gradients Least Squares (CGLS)[HS52]. Also, there are toolboxes which have already implemented such algorithms, like Tomopy[GDCXJ14], Astra-Toolbox[PBS13], TIGRE[BDHS16], etc. The comparison of them is shown in Table 2.1.

Toolbox	3D Reconstruction Algorithms	GPU Supporting	Programming Language
Astra-Toolbox	FDK, SIRT, CGLS	YES	matlab
Tomopy	ART, BART, FBP, GRIDREC, MLEM, OSEM, OSPML, PML, SIRT	YES (with Astra-Toolbox)	python
TIGRE	SART, OS-SART, SIRT, CGLS, MLEM and Total Variation Algorithms	YES	matlab, python

Table 2.1: A comparison of the 3 toolboxes.

After a discussion with teammates and supervisors, the Astra-toolbox was chosen as our reconstruction tool. In our opinion, Astra-toolbox has a better document which could give us a clearer tutorials and code examples. Another point was that considering we had to do the processing of projections and reconstructed images with Matlab Image Processing Toolbox, a reconstruction in the same environment was more convenient for us to handle the working process

and the data. Based on these preferences we had a study on the Astra-toolbox.

After reading the example code, our first step was testing different algorithms for reconstruction: BP3D, FDK, SIRT and CGLS. We tested it with a cube and a cube which has a cavity in the center of it. We firstly created such volumes, then used the simulation functions provided by Astra-toolbox to generate the sinograms of the objects. After we got the data set of sinograms, we used them as input of the different algorithms to get the reconstructed 3D volumes, and had a look at slices of the volumes. Figure 2.2 and Figure 2.3 are the results of the test.

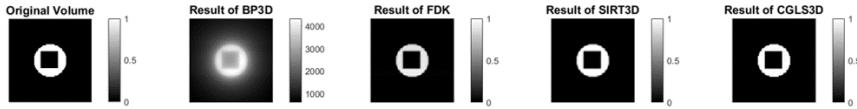


Figure 2.2: Test Result of the tube.

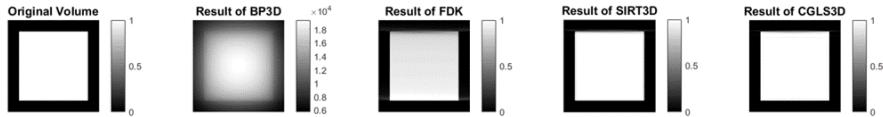


Figure 2.3: Test result of the cube.

It is easy to find that for the tube, the FDK, SIRT and CGLS are all working well. And the FDK algorithm doesn't work well for the cube scan. In order to find more about the difference between SIRT and CGLS, we tested the 2D version for them with the phantom image generated by the matlab build-in function. Figure 2.4 shows the test result:

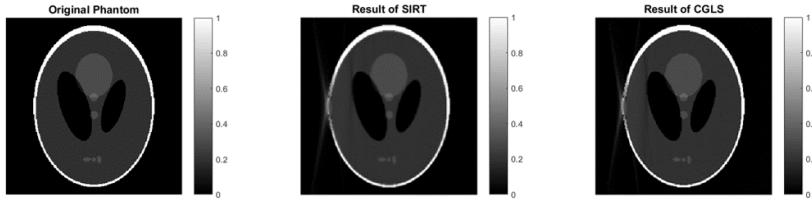


Figure 2.4: Test result of the phantom image.

From the result of SIRT and CGLS, we can find that the both SIRT and CGLS images have the artifact. The difference between them is that the SIRT image is blurred, which making the artifact looks better; and the CGLS image is clearer in details, which is also making the artifacts sharp. Under our preference and the suggestion from the project team, we decided to use SIRT as the main algorithm, and CGLS as the spare method being used when required.

2.1.2 Reconstruction Results

Figure 2.5 is the reconstructed slices of a match head. Projections were output by the DTU PCXCT system. 500 projections were photoed during the scan. The acquisition time was 1s, and the detector counted the photon 3 times. The source-sample-distance was 330mm and the sample-detector-distance was 834mm.

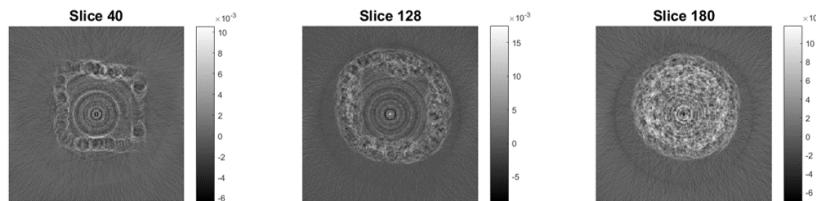


Figure 2.5: Reconstruction results of the match head.

From the reconstruction result we could find that the main part of the match head was projected and reconstructed, but not with a high SNR. Also, the ring

artifacts caused by the bad pixels were clear in the slices. And the background, which was influencing the clarity of the match head, looks having very small value less than 0. In order to comprehend the result more deeply, we had a look at the histogram, which is shown in 2.6:

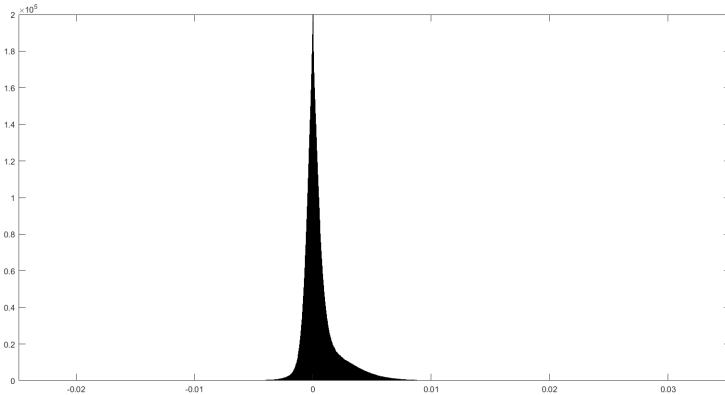


Figure 2.6: The histogram of the reconstructed volume. X-axis is the grayscale value and Y-axis represents the number of pixels.

Ideally, there should be a Gaussian distribution in the histogram. However, in our result, the "Gaussian distribution" looks bending, and the peak is 0. This figure has a y-axis limit of 200000; in fact, the pixels having the values near 0 is much more than we can see in this figure (the number of zero pixels is approximately 3.5×10^6).

With choosing some window in the background, we have found that almost all backgrounds were having the value smaller than 0. So we tried to set the pixels smaller than 0 to 0, and had a look at the slices shown in Figure 2.7 again:

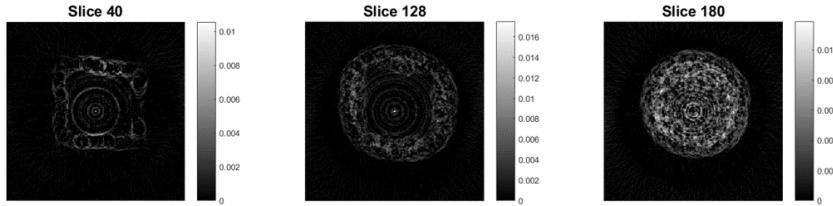


Figure 2.7: Test reconstruction results of the match head, in which the negative values are removed.

Compared with 2.5, we can see that both the background and some pixels in the match head were erased. With this volume we would be able to see the structure of the match head shown in Figure 2.8:

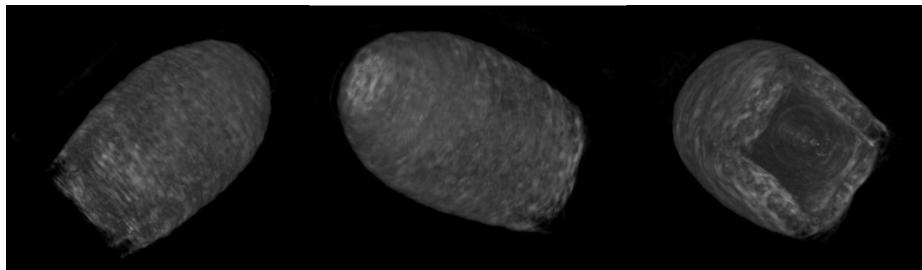


Figure 2.8: The volume of the match head. The 3D volume is displayed by Fiji-ImageJ Toolbox.

Which looks like a match head should be in reality. Also, another scan made by the DTU PCXCT system with the brain sample is also processed, giving result shown in Figure 2.9:

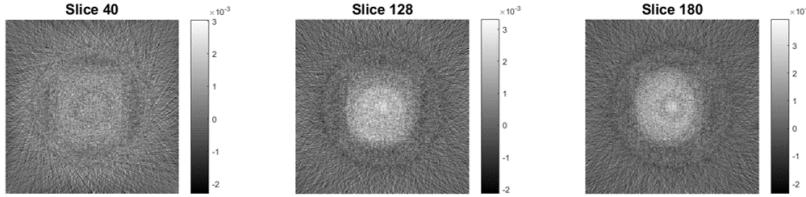


Figure 2.9: Test reconstruction results of the brain sample.

From 2.9 we could find that the brain sample could be seen in the slices, but the SNR is even worse than that of the match head reconstruction. Furthermore, we tried to delete the negatives as well, and obtained results shown in Figure 2.10:

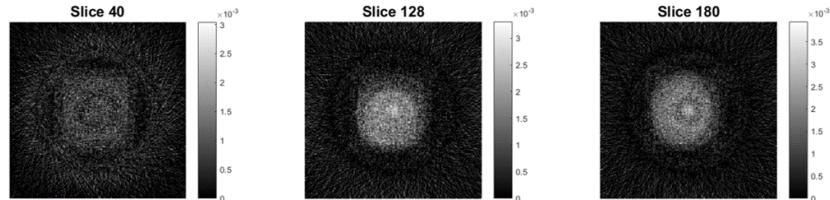


Figure 2.10: Test reconstruction results of the brain sample, in which the negative values are removed.

And we found that the background noises were partially removed, but it was still noisy. Also, we can see the sample part was clearer than before, although the inner structure was still invisible to us. Moreover, in Figure 2.11, we can find that the the noisy background influenced us greatly, which made us can hardly see the brain sample.

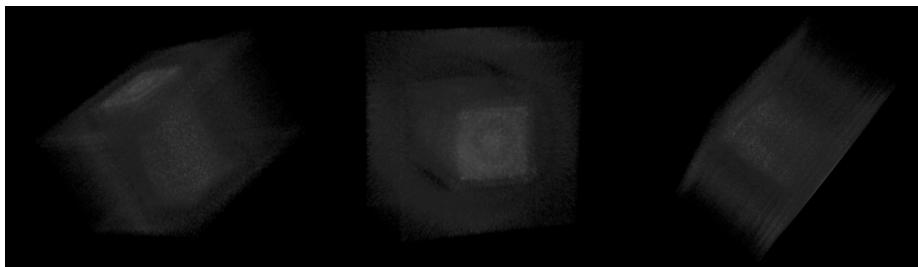


Figure 2.11: The volume of the rat brain. The 3D volume is displayed by Fiji-ImageJ Toolbox. The noise surrounding the sample formed a cube.

We deleted the values smaller than 0 in both results, in order to get better images. However, there is a risk that we cannot make sure that whether the negative-removal process detected other important information which is not in the background. Thus we tried to find out whether there are image enhancement method for processing the reconstructed images, then we needn't to delete the background violently.

2.2 Image Enhancement

Generally, the values in the output 3D volume are not in the range of $[0, 1]$ or other standard color ranges. Since sometimes the computer programs only take the values in range $[0, 1]$ in consideration while rendering or outputting, some detail may be invisible if we do nothing to the voxel value intensity. Thus, the image normalization and enhancement methods should be implemented before analyzing the image.

2.2.1 Study of the Image Enhancement Methods

When we think about image enhancement, the first method we come up with is very likely to be histogram equalization. Histogram equalization is a general used method, the main idea of it is adjusting the original histograms which are concentrated in one area to a balanced distributed ones. With mathematical expression, the histogram of a grayscale image can be represented by:

$$P_s(s_k) = n_k/n \quad 0 \leq s_k \leq 1, \quad k = 1, \dots, L - 1$$

Where $P_s(s_k)$ is the probability of the occurrence of k -th gray level, and L is the number of levels. Then as [Gon16] discussed the expression of histogram equalization would be:

$$t_s = HE(s_k) = \sum_{i=0}^k (P_s(s_i)) \quad 0 \leq s_k \leq 1, \quad k = 1, \dots, L - 1$$

Thus, we can calculate the adjusted grayscale value of each pixel with the original histograms.

Histogram equalization is good for enhance the images whose grayscale distribution is very concentrated, however, it sometimes gives an output with some over-enhanced details if there are some high peaks in the histogram. Thus we studied another method, gamma correction, which is also widely used in image enhancement. The main idea of this method is using the power-law formula to compress the ranges of grayscales, in order that the visibility of the detail information in the image is optimized. The formula of this method is:

$$s = cr^\gamma$$

Another solution is gray stretch, which is also used for emphasize one or some components in the images. Doesn't like gamma correction with this method, we are able to stretch the dynamic range of the images. The most simple version of gray stretch is linear contrast stretch. For emphasize the interesting information of each image, we divide the pixels into three sections with the gray value: S_a , S_b and S . The main idea of this method is compressing the S_a and S_b parts of the image, and stress the S , the components we want to enhance. A mathematical expression of this method is:

$$I_{out}(i, j) = \begin{cases} \frac{g_a}{f_a} I_{in}(i, j) & I_{in}(i, j) \leq f_a \\ \frac{g_a - g_b}{f_a - f_b} (I_{in}(i, j) - f_a) - g_a & f_a < I_{in}(i, j) \leq f_b \\ \frac{255 - g_b}{255 - f_b} (I_{in}(i, j) - f_b) + g_b & I_{in}(i, j) > f_b \end{cases} \quad (2.1)$$

Where $I_{in}(i, j)$ and $I_{out}(i, j)$ represents the pixel at column i , row j in the input and output images. $[0, f_a]$ and $[f_b, 255]$ (in double matrix image it is $[0, f_a]$ and

$[f_b, 1]$) are the two ranges defining the pixels in S_a and S_b , as well as pixels with gray value in $[f_a, f_b]$ are in S . Also, g_a and g_b defines the slope of the linear transformation.

Thus, after our study we have 3 methods for enhancing our reconstructed images. Then we will know that which is the best methods for enhancing our images.

2.2.2 Application in Match Head Images

Before using the enhancing the images, a image normalization is suggested considering some technique requires the range of values. Thus we could normalize the image with the following formula:

$$I_{new} = \frac{(I - \min(I))(\max(I_{new}) - \min(I_{new}))}{\max(I) - \min(I)} + \min(I_{new}); \quad (2.2)$$

Where I is the original image, and I_{new} is the normalized image. $\min(I)$ and $\max(I)$ represents the minimum and maximum value in I , and $\min(I_{new})$ and $\max(I_{new})$ is defined by user, determining the value range of I_{new} . After normalization, we could try the techniques we mentioned above and see their performance of processing our slices.

We could see that after the the HE enhancement, both the noise and the main part of the match head turned brighter, which is shown in 2.12. The match head was clearer after the enhancement; however, the noises, like ring artifacts, were also greatly enhanced. So we can say that after histogram equalization the contrast it grater than before but it is still hard to make the image clearer.

From the result in Figure 2.13, we could see that this enhancement made the noises in background unclear than the original one (see 2.12). Especially the result with $\gamma = 2$ looks similar to the method setting the negative pixels to 0 we mentioned above. The similarity can be seen clear with a comparison between their histograms, which is shown in 2.14.

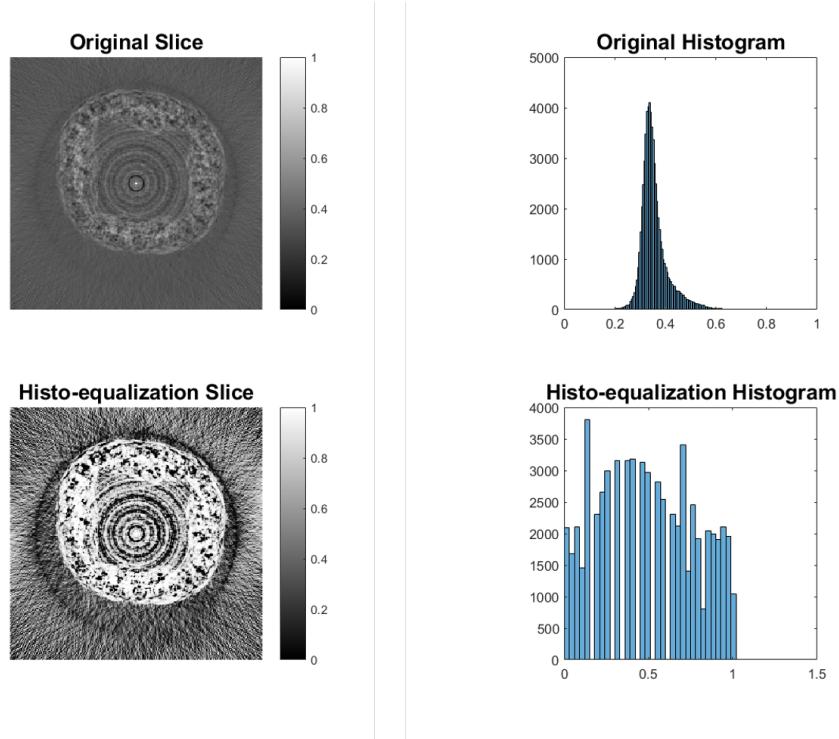


Figure 2.12: The result and histogram of the original and Histogram-Equalization enhanced slice.

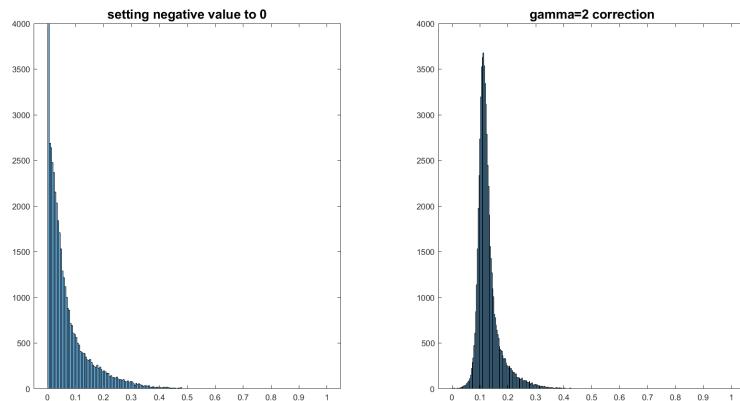


Figure 2.14: The histograms of negative-deleted normalized image and the gamma corrected with $\gamma = 2$

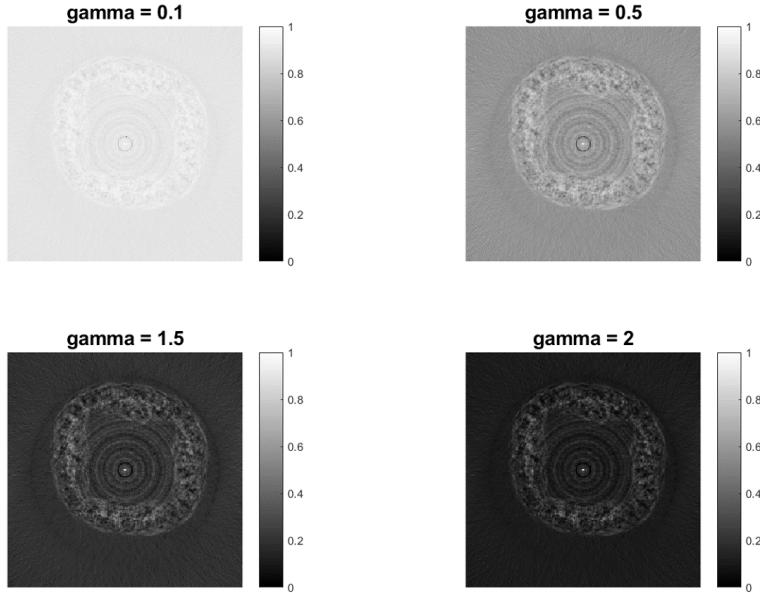


Figure 2.13: Results of gamma correction.

Then with a comparison, the histogram of negative-deleted normalized image, we call it left histogram, looks like a crooked version of the histogram of the gamma correction result, we call it right histogram. The right histogram looks more like a normal distribution. And the value range of pixels in the two images are also similar, which could explain the similarity of the two pictures. For ensuring the two images are really similar, we calculated and compared the hash value of two images. The length of their hash is both 16384, and there are 14416 common characters in the same location in two hash values. So the two results could be considered to be similar.

For the gray stretch method, we have already known this technique can be used to reduce the background of the image, and enhance the part we interested, if we could give it proper values. After choosing some windows in background and find the value range of them, we found that the background pixels are mostly smaller than 0.4, and the biggest value in the match head was smaller than 0.7. Thus we set $f_a = 0.4, g_a = 0.2$ to reduce the background, and $f_b = 0.7, g_b = 0.9$ to stretch the match head.

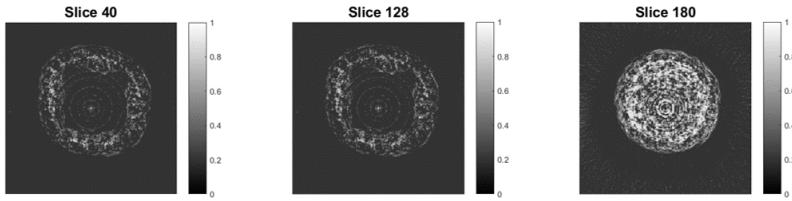


Figure 2.15: The results of gray stretch.

The results shown in 2.15 is also giving us surprise: the technique is giving the similar result of negative-deletion and the gamma correction with $\gamma = 2$, which means this algorithm also works well for us to enhance the contrast so that the match head is visible. Then we decided to apply these methods in brain images.

2.2.3 Application in Brain Images

In order to attempt to obtain a set of better images of the brain, we tried the enhancement with the brain slices as well. The images in Figure 2.16 are the results of the attempts.

From the results we can see that the HE-enhancement didn't work good as well; and the gray stretch and gamma correction, which were giving us good results of match head, worked not so good in this time. According to the result and our study about the values in matrices, we can find that the values in the background are very closed to the values in the brain, which means the contrast of the original image was poor. It is easy for us to make backgrounds invisible; however, if we do that, the information in the sample part will also be reduced. Thus in such condition, an optimization on the system takes more effect than struggling on the bad slices.

In summary, gamma correction and gray stretch can help us with optimizing the reconstructed results. Also, the techniques doesn't work well in all images, just like in our experiment, the enhancement gives us better match head slices, but image slices in which information is lost.

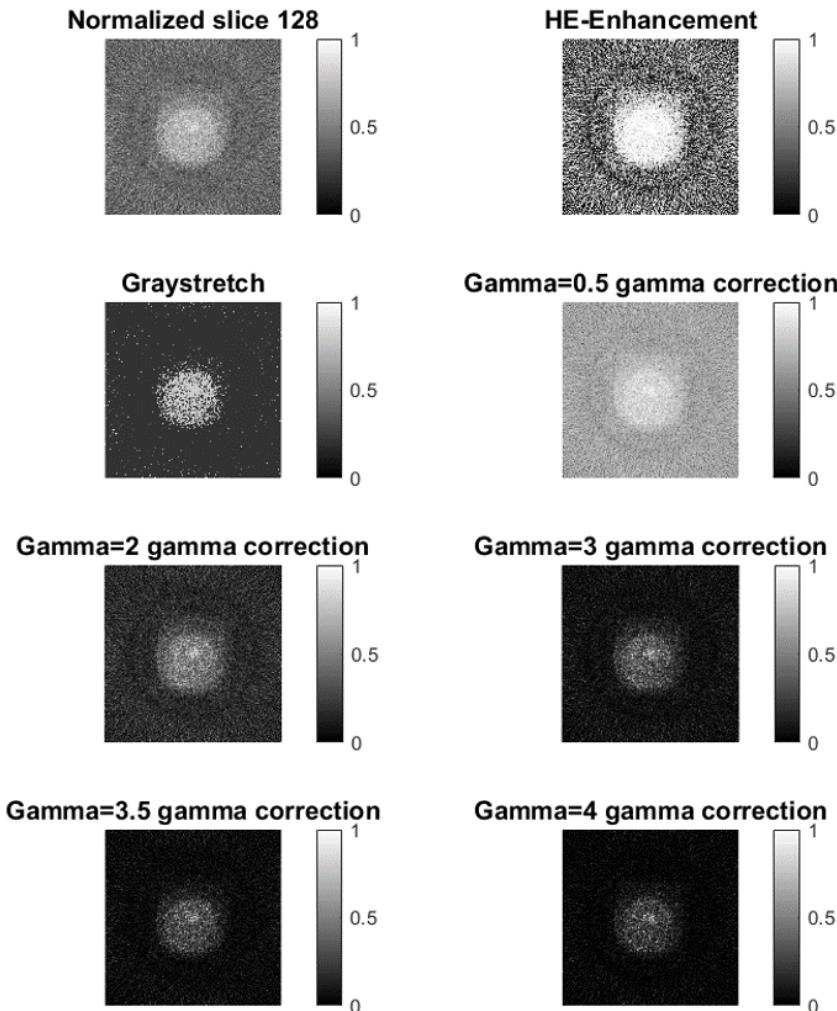


Figure 2.16: The attempts of enhancing brain sample images.

CHAPTER 3

Projection Preprocessing

Projects are the 2D matrices storing the result of the x-ray scan. With the matlab image viewer, the projection could be visualized so that a brief check of the projections could be implemented. Generally there would be problems in the projections, like bad pixels, off-center, etc. Thus, a preprocess for the projections could be introduced before reconstruction in order that images with a higher quality can be obtained.

3.1 Method to Detect Bad Pixels

It is hard to find an authoritative definition of "bad pixels", as [RMS14] says, the definition of "bad pixels" or "cosmetic defects" are seldom discussed. In [RMS14], they defined the bad pixels as the super set of those deemed inoperative by the manufacturer, and the pixels which the user prefer to exclude in the belief that such pixels would downgrade the science although there are calibration efforts. Another thesis [BKZV04] discussing the detection and correction from the sensor aspect, cast that bad pixels are resulted by a number of bad photodetectors on the sensing device array. An article [JMLA02] classified bad pixels into 3 species. They are (1) dead pixels, which are always producing the same signal; (2) noisy pixels, which are having the noise greater than a fixed

threshold; and (3) blinking or drafting pixels which are having temporal behavior different than the pixels deemed as the good ones. Hot pixel is also mentioned by [HLC⁺], with an definition of the pixels with a high dark current, resulted by the detector units warmer than their operating temperature; while leaving them alone they would continually increase. In summary, we could say that the bad pixels are the ones giving abnormal values, which is different than the actual number of quantum that one detector unit should response. In our projections, we also found the different types of bad pixels. One kind satisfied to the definition of dead pixels in [JMLA02], which are always giving similar value to 1, 0 or NaN. The second kind of them are giving different value in scans, but having a more sensitive response than the good pixels with the increasing acquisition time.

The clearest bad pixels are those dot which visibly has the different value with its neighbours when we are plotting the projection matrices with "imshow" or "imagesc" tools of matlab. Moreover, there are some pixels have different (higher or lower) reflections to the beam, resulting a brighter or darker points on the projections. Thus, for a better imaging result, detecting and correcting the bad pixels is suggested to be attempted.

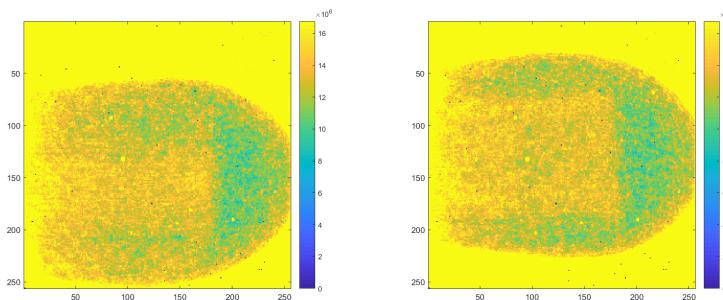


Figure 3.1: Bad pixels in match head projections.

Another method is the gradient method. As we know, the dead pixels on the flat fields always have a value much bigger or smaller than their neighbours (unless their neighbours are also dead pixels). A sudden decrease or increase from the neighbour might results a big gradient. So a gradient method can also be considered for detecting the dead pixels.

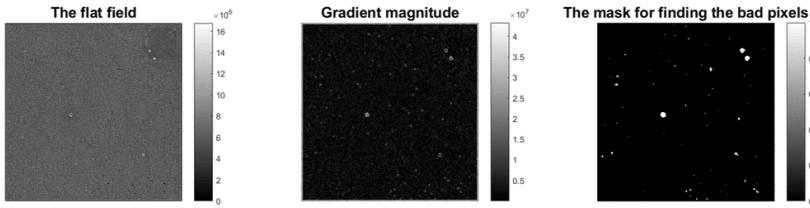


Figure 3.2: The flat field, its gradient magnitude and the mask made with the magnitude.

Figure 3.2 shows a flat field and the gradient magnitude of it. In the magnitude, we can find that there are bright dots, which are the detected bad pixels. With this pixels, we can then generate a mask used for the correction, which would be discussed us in the next section.

Dead pixels are the most easily detected bad pixels. Two slices of match head containing such pixels are shown on Figure 3.1. Visibly such pixels are the very bright or dark pixels compared with their neighbours, so human could directly find the pixels appear in the matrices. However, another kind of the bad pixels, which are named 'abnormal pixels' in [Lyu18], are not really easily to be found considering sometimes it looks like the relative detect units are working well considering the value on such locations changes with the rotation of the sample. So we have to find the methods for full-fill the set of the bad pixels.

Generally, the flat field is the best choice for detecting the bad pixels, as there would be no scanned sample impacting the output of detector. If the pixel is acting abnormal in the flat field then we could know that such a bad pixel is resulted by the detector instead of the movement or other specific properties of the sample.

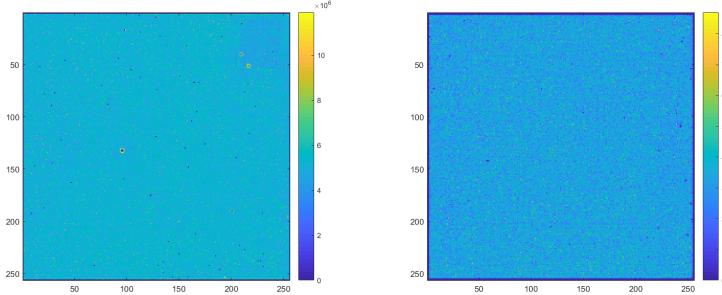


Figure 3.3: Flat fields of the match and the brain sample scan.

Figure 3.3 are two flat fields from the match and brain scan. We can easily see the dead pixels in the figure, but not the abnormal pixels. For visualizing the performance of the pixels, [Lyu18] cast a method, which is observing the count of each pixel in different acquisition times.

Each pixels on the detector are able to measure the number of photon while they are exposed in the beam. According to the working principle of the detector, the output value of one pixel should be in proportion to the acquisition time, if the acquisition time is not too long:

$$R = kt$$

Where R is the output of the pixel on the detector, and t is the acquisition time. Ideally, the coefficient k of each pixel should be similar, considering they are exposed in the same beam. However, because of the defect of detector, some pixels might have more or less sensitive response to the increase of the acquisition time.

For visualizing the response of each pixel clearly, we could plot the flat field data on a Cartesian Coordinate, where y-axis is the response (count) R and x-axis is the acquisition time t . So for the points of each pixel, a linear regression result of such points could represent how the acquisition time is influencing the output of that pixel; and the slope, which is represented by k , can show the sensitivity of a specific pixel. By plotting the fitting results of all pixels, it would be easy

for us to see that some pixel are having a higher or lower k value:

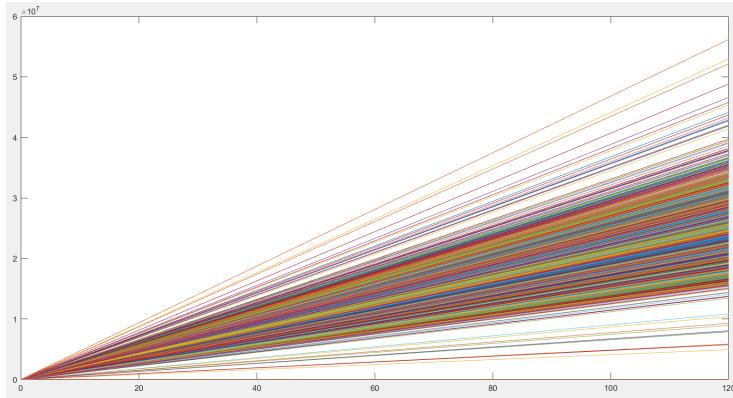


Figure 3.4: The linear regression result of the pixels.

Figure 3.4 shows the slopes, in other words, k values of the pixels obtained by the linear regression. By plotting them in a coordinate, we can see that most lines appear in one area, which represents the well-working detector units; and the lines which are not plotted in this area are resulted by the abnormal units. With the histogram shown in 3.5, we can see the distribution of the values clearer:

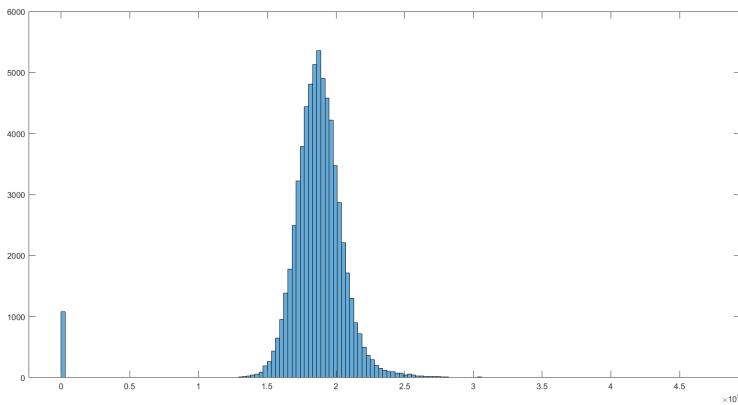


Figure 3.5: The histogram of the slope values.

We can see that in the histogram there is a Gaussian distribution in the center.

Also, there are a lot of pixels having the slope 0, which means the dead pixels. Considering their value seldom changes, they made the slope 0. In [Lyu18], J. Lyu suggested a method, which is using the Gaussian fit to obtain the expression of the Gaussian distribution, then choose the pixels lied in the confidence interval. With this approach we could find out almost all (sometimes more than) dead pixel and abnormal pixels.

3.2 Method to Correct Bad Pixels

We mentioned that the bad pixels are not representing the actual number of photon caught on the detector unit. With a set of abnormal measure results, the bad pixels would produce artifacts in the reconstructed image, making it hard to identify and analyze the content of the image by algorithm or even by human eyes. Also, if the normalized projection has NaN values, some algorithms will be not able to work. A lot of articles are introducing how to fix the detector in order to reduce or remove the bad pixels; however, our duty is trying to correct the projections without doing works on the detector. Thus, it is better for us to find and study one or some good correction methods with which we can estimate or directly get the real values which should have been returned on the bad pixels.

3.2.1 Median Filter

The most simple way to correct the NaN pixels in normalized projections is setting them to zero. However, although it makes the algorithm work, such pixels are still producing artifacts (ring artifacts) considering they are transformed to the dead pixels which are giving the same values. So we have to think about solutions setting them to non-equal values, and the values should be a good estimate of the actual value otherwise the correction is pointless. One general method is median filter, which means replacing the bad pixel with its neighbours' median value. Normally we are using a 3×3 or 5×5 -size-window, and the center of the window is the bad pixel we are correcting. Figure 3.6 is a demonstration of median filter:

0.4093	0.2478	0.2502	0.1296	0.3069	
0.3437	0.1811	0.1363	0.0988	0.0586	
0.2156	0.1185	NaN	0.0566	0.1384	
0.1786	0.2000	0.2968	0.1787	0.1087	
0.2254	0.1834	0.2229	0.2146	0.2720	

0.4093	0.2478	0.2502	0.1296	0.3069
0.3437	0.1811	0.1363	0.0988	0.0586
0.2156	0.1185	0.1917	0.0566	0.1384
0.1786	0.2000	0.2968	0.1787	0.1087
0.2254	0.1834	0.2229	0.2146	0.2720

Figure 3.6: A demonstration of the median filter. The median of the value is calculated and chosen for correcting the NaN pixel.

One problem here is that if the bad pixels are closed enough (or the window is big enough), it is possible that there are more than 1 bad pixels in a window. Our solution is then ignore the pixels which are also bad, and only choosing the median value of the right-measured ones.

After using the bad pixel detection method we introduced in previous section, we could get a mask which covering the detected bad pixels. Then we would not only be able to correct the dead pixels but also the abnormal pixels in the un-normalized projections, just like the result of Figure 3.7.

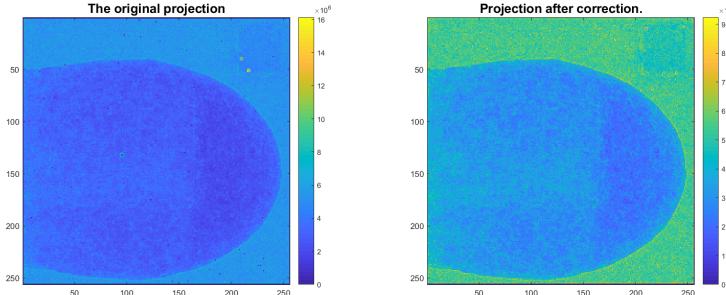


Figure 3.7: The result of correcting a projection with median filter.

3.2.2 Calibration with Referenced Projection

From the previous section, we have known that the bad pixels are resulted by the problem with the detector units. And our target is to get the true values of every unit despite which approach we used. Thus, if one unit is not working well for counting the photon from the beam which is influenced by a part of the

sample, we can move the sample a little (by some pixels) and use another normal unit to count the missing part. A mathematical expression for this process is:

$$P(i, j) = P_c(i + \sigma, j + \gamma)$$

Where $P(i, j)$ is the value the pixel at column i , row j on the projection supposed to have, which means the number of photons hit on the detector unit at column i , row j , we call it $U(i, j)$. P_c is the projection for calibration, which is obtained by moving the sample σ pixel to the right and γ pixel to the bottom. It is easy to know that then the detector unit $U(i + \sigma, j + \gamma)$ is counting the photon number passed though the same part as $U(i, j)$ in the previous scan. Thus, we are able to calibrate the bad pixels by the second scan. Figure 3.8 shows the original projection and another one used for calibration.

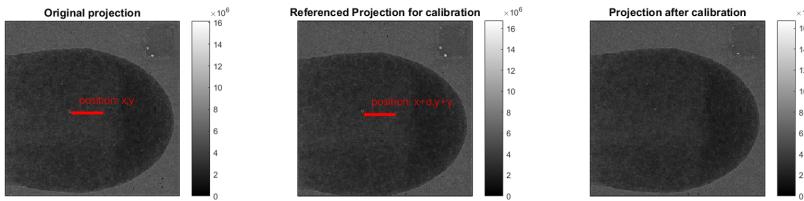


Figure 3.8: Original projection, reference projection and the calibration result.

Normally, the pixel pointed by the red arrow in the original projection has the same value as the pointed pixel in the referenced projection. However, from the projection we could see that the pixel on the original projection is a bad pixel. So we are able to assign that pixel value with the corresponded one in the referenced projection. Here we used $\sigma = 0, \gamma = 6$ for the second scan, which means in the second scan we moved the sample 6 pixels down. In the projection after calibration, we could easily find that the number of bad pixels is reduced.

The advantage of this method is that we do not need to guess the right values of the bad pixels, but find the actual value of them. However, one condition should be considered, which is $U(i, j)$ and $U(i + \sigma, j + \gamma)$ are both bad pixels. Then we cannot get the actual value of $P(i, j)$. So we suggest making sure that each $U(i + \sigma, j + \gamma)$ is a good pixel before the second scan. If we cannot find such a pair (σ, γ) , the third scan could be taken in account; if the third scan

is still impossible to cover every bad pixel, then there might be too many bad units in the detector.

3.3 Stripe Removal in Sinograms

Sinogram is a set of projections in different angles, and one sinogram consists of the same row in each projection. If we show the sinogram vertically, we can easily find that the rows are the same to them in original projections, and the y-axis represents the number of one projection in the serial, which also means the angle of that projection.

So we could know that, if there is a bad pixel in a row of the detector, then in each projection there would be a bad pixel in that row. Thus, a stripe would be visible in a sinogram containing this row in all projections. So our target here could be removing stripes from a image instead of correcting the pixels from a number of projections. Thus, we have to find methods for detecting and correcting the stripes in sinograms.

[MTMS09] cast a method for removing the stripes in images, including sinograms. In this article, a combined wavelet-fourier filtering is introduced. For detecting the stripes in the images, Beat Münch and his team suggested a single-level wavelet decomposition, with which we could get a approximation coefficients matrix, and the detail coefficients matrices in horizontal, vertical and diagonal. Hence, we are able to get the information of stripes of the image in each decomposition level with the detail coefficients matrix in vertical C_v .

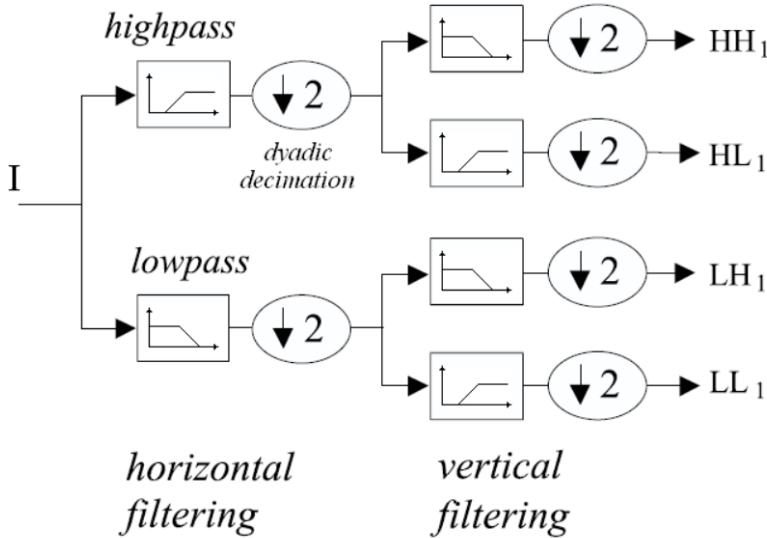


Figure 3.9: The steps of a wavelet decomposition, referenced from [Mas09]

After the C_v is obtained, the article proposes to run a Fourier transform with C_v under the theorem that there would be no frequency components resulted by the vertical stripes in \hat{y} of the frequency domain $F(\hat{x}, \hat{y})$ of $f(x, y)$. So the solution is damping the vertical information by removing the Fourier coefficients $F(\hat{x}, \hat{y})$ at all \hat{x} for $\hat{y} = 0$. And the damping is achieved by multiplying the FFT coefficients with a Gaussian Function:

$$g(\hat{x}, \hat{y}) = 1 - e^{-\frac{\hat{y}^2}{2\sigma^2}} \quad (3.1)$$

Then by the inverse FFT with the damping result, a new vertical band C'_v in which the information of stripes is removed could be obtained. Finally, the authors did a wavelet reconstruction (NOT tomographic reconstruction) for the new image, with the new vertical band and the other bands without changes.

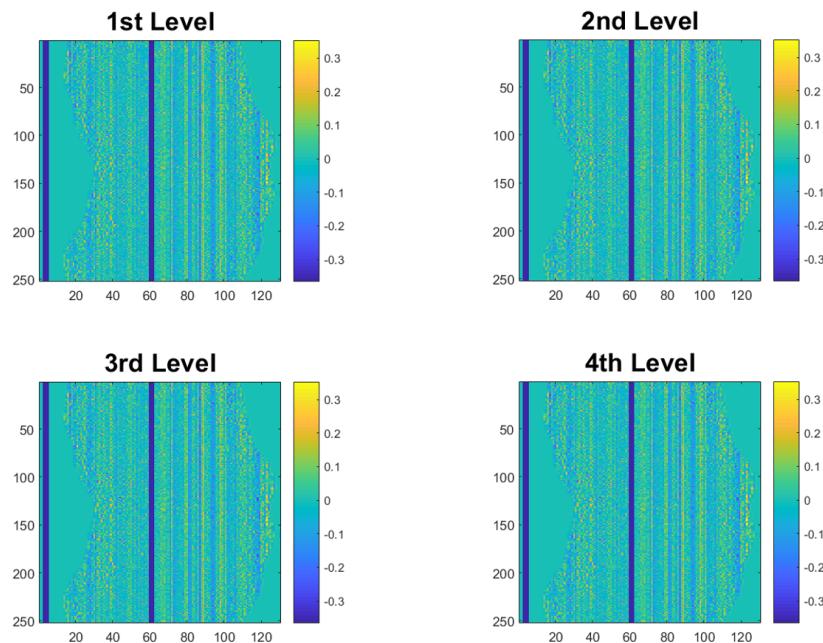


Figure 3.10: Decomposition of one sinogram.

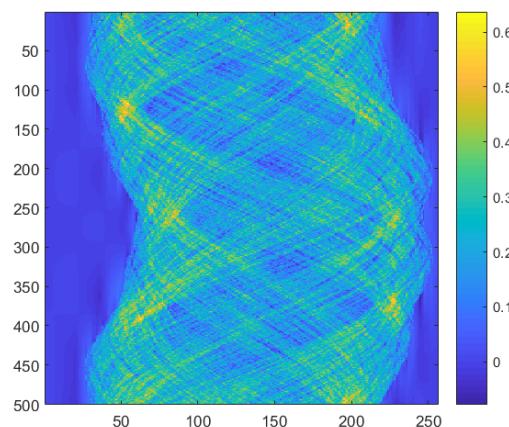


Figure 3.11: The new sinogram after a wavelet-Fourier filter. The decomposition level $L = 4$, sigma is 4 and the wavelet is DB3.

3.3.1 Processing Results

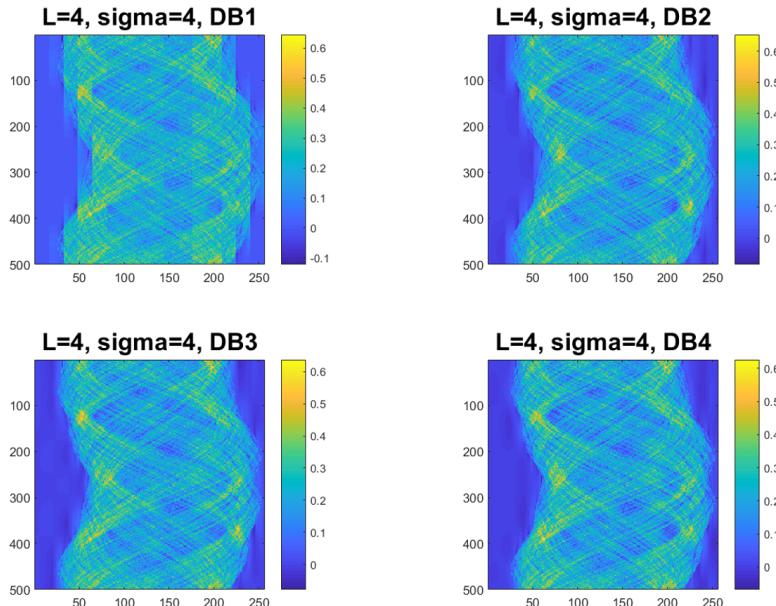


Figure 3.12: Testing the influence of the number of vanishing moments.

Results in Figure 3.12 shows that the bigger the vanishing moments is, the cleaner the stripes are erased. In the first sinogram processed with DB1, the stripes could still be seen, and became fatter; however, in the DB4 image, the stripes are almost vanished.

Results in Figure 3.13 tells us that more decomposition level could give us a sinogram with less stripes. In the result of 1 level, 2 levels and 3 levels, the stripe are easy to see but in the last result the stripes are erased well.

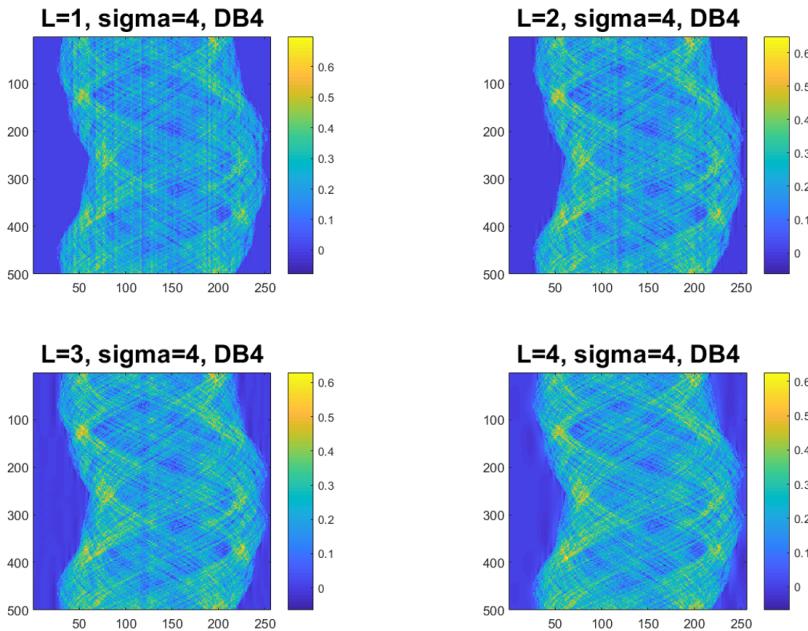


Figure 3.13: Testing the influence of decomposition level (L in image).

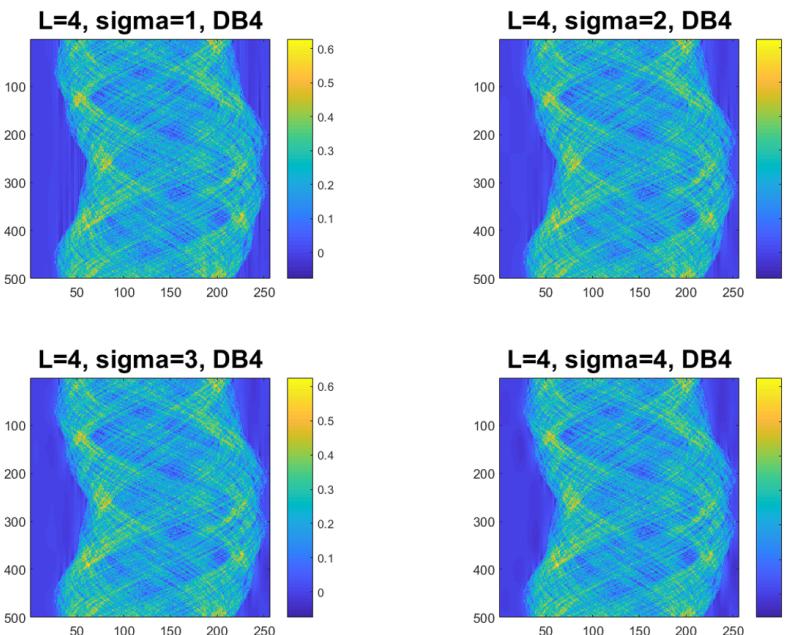


Figure 3.14: Testing the influence of sigma value.

Results in Figure 3.14 gives us the influence of the sigma value. The bigger the sigma, the more information of the stripe is lost. The change is not clear in the main body of sinogram, but easy to see in the sinogram background.

Like Beat Münch said in his article, there is a contradict with this algorithm: the clearer we erase the stripes, the more information is lost. We could of course use decomposition level of 64, or a DB42 wavelet; however, after that we cannot make sure how much information is damped. That's way we suggest a balance between the performance of the algorithm, and the information loss we could tolerate.

3.4 Off-Center Correction

3.4.1 Problem Analysis

Off-center is also a general problem in the CT imaging. Normally, the center of the sample should be aligned with the center of rotation. However, sometimes it is not easy to achieve such thing considering the sample and the rotating plate could be very small for us to put the sample at the appropriate position. Then during the scan, while the sample is rotating, it is possible that in the projections photoed by the detector, the graph of sample is moving left and right. Figure 3.15 demonstrates the off-center problem and how it is impacting the sinogram graphically:

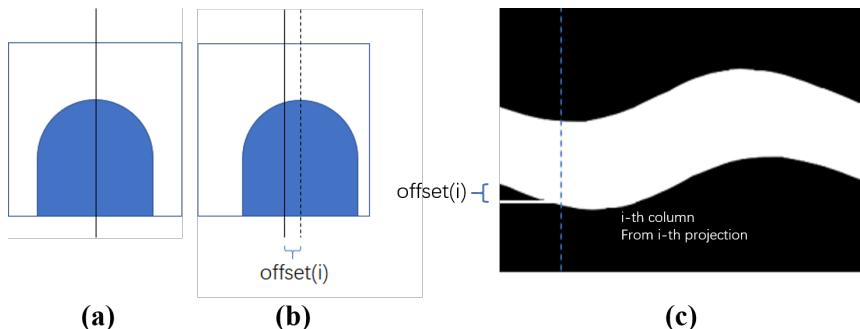


Figure 3.15: Description of the off-center problem.

Normally, the well centered project looks like 3.15 (a). If the sample is not well aligned, just like that in 3.15 (b), an $offset(i)$ would appear at i -th column,

which comes from i -th projection. Then the sinogram is bending, and the offset in the projection $offset(i)$ is equal to the column offset in sinogram which is shown in 3.15 (c).

We could see considering the columns in the sinograms equals to the same row of different projections, while the sample is moving, the offset of the sample, refers to the offset of a row in the projection, would equal to the offset of the correspond column. If we consider the relationship between position of the sample and the center of rotation, which is demonstrated in Figure 3.16:

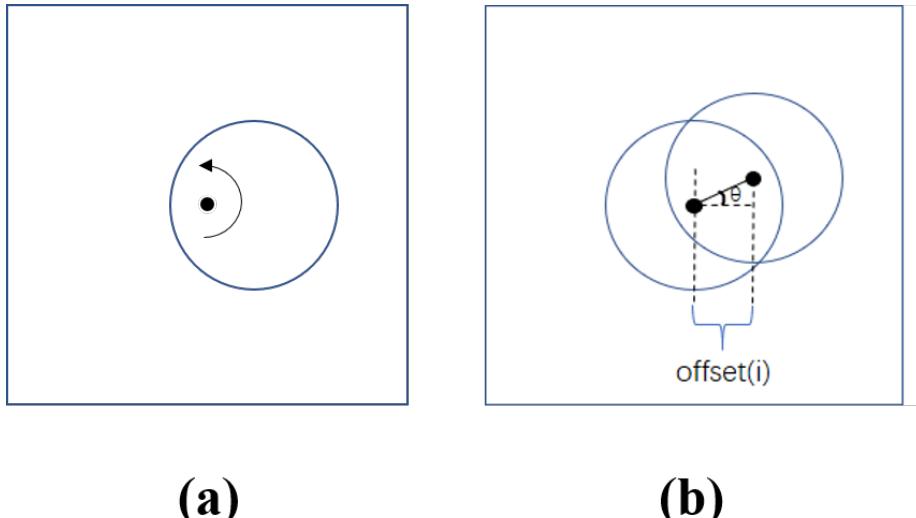


Figure 3.16: Relationship between the position of the sample and center of rotation. (a) shows the initial position of the sample, where the center of rotation is not the center of sample. (b) demonstrates the i -th scan, and the rotate angle is θ .

Thus, we could see that the offset in the i -th projection, refers to that in i -th scan, can be calculated by the rotation angle θ and the distance d between the sample center and the rotation center:

$$offset(i) = d * \cos\theta, \text{ where } \theta = \frac{\theta_{all}}{n} * (i - 1) \quad i = 0, 1, \dots, n - 1 \quad (3.2)$$

Where n is the total number of projections, and θ_{all} is the rotation angle of the full scan. Generally, θ_{all} is π or 2π in the CT scans.

3.4.2 Simulation of Off-center Conditions

In the previous section, we have explained the off-center problem. In order to study the relationship between the change of sinogram bewtween the off-center distance in the horzonital and vertical directions, we designed and implemented an experiment. We want to simulate the sinograms of the 3D tube which is located in the center of the volume (see Figure 3.17 with Astra-toolbox, which is going to be introduced in next chapter. Then we move the tube, simulate the new volumes again and look at how the sinogram changes.

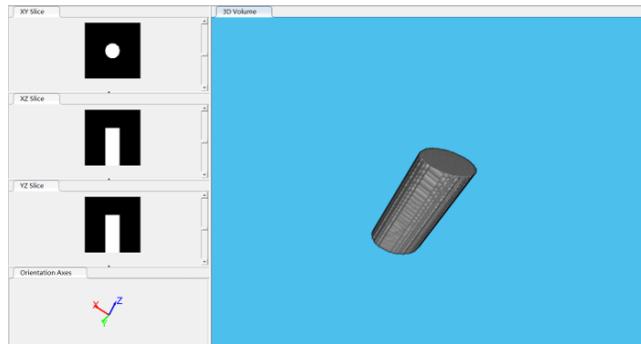


Figure 3.17: The tube volume used for the simulation.

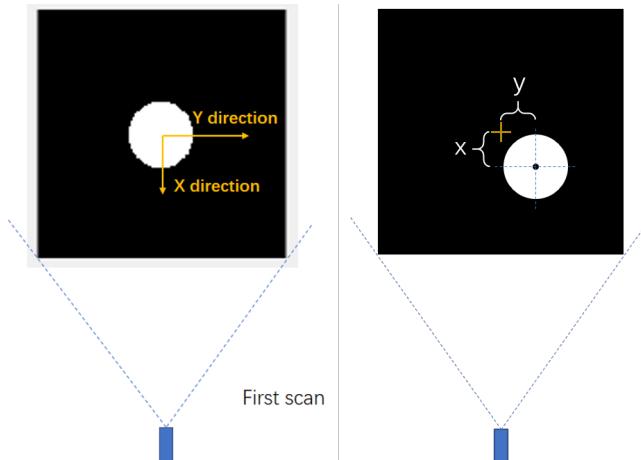


Figure 3.18: A demonstration of the movement of sample in the simulation.

For getting a sinogram from the off-centered samples, we decided to move the

tube to X-direction (vertical) and Y-directions (horizontal) by x and y pixels. The movement of the sample is shown in Figure 3.18. Then a set of sinograms would be obtained. Figure 3.19 shows the simulation result of the movements:

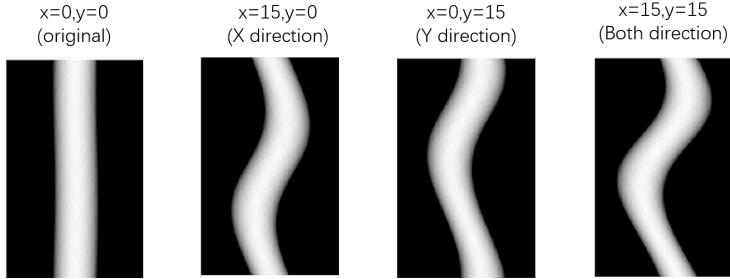


Figure 3.19: The original sinogram and the changed sinogram by moving the sample.

With the simulation, we can easily find that the shape of the sinogram is changing to the graph of a sine function, cosine function or the Fourier series. By plotting the each middle point in the sinogram columns, the changes looks clearer:

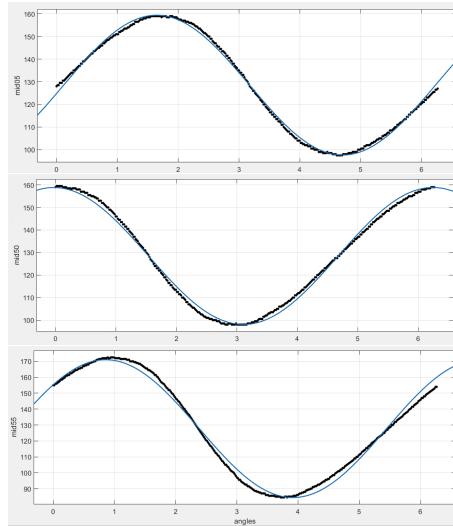


Figure 3.20: The middle points compared with the corresponding Fourier fitting result. The top, middle and bottom graphs are the result of movement $x = 0, y = 0$, $x = 5, y = 0$ and $x = 5, y = 5$, respectively. The X-axis represents the angle of the projection and Y-axis is the position of middle points.

The fitting of the points gives a good result, which means that the movement of columns of the sinogram follows a Fourier series graph. This also confirms the mathematical relationship, that is, if the sample is misaligned x pixels in vertical and y pixels in horizontal, from the perspective of i -th scan with a current rotation angle θ , the $offset(i)$ reflects the combination of the impacts of the off-center in two directions.

3.4.3 Application in our sinogram

Thus, for testing the result we got from the simulation, we chose a sinogram from the match head scan, and removes the background of it:

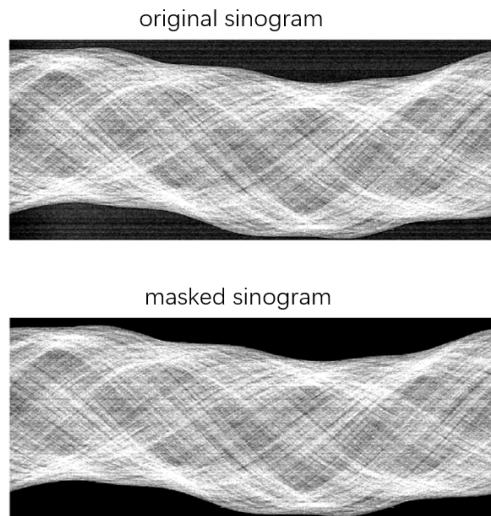


Figure 3.21: Removing the background of a sinogram.

And then we plotted the middle point of each sinogram columns, and run a Fourier fitting for the curve of such plot (the input of the curve fit is the column numbers, and the output is the position of the middle points). Figure 3.22 shows the curve and the fitting result:

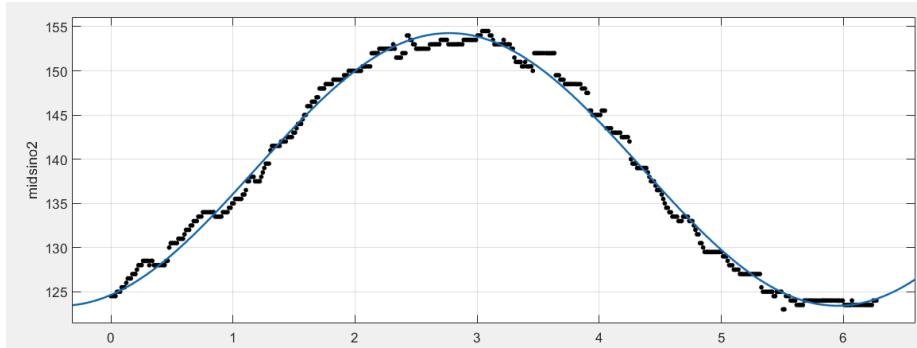


Figure 3.22: The curve of middle points and the fitting result. The X-axis represents the angle of the projection and Y-axis is the position of middle points.

By the fitting, we can know that the expression of the curve could be:

$$y = f(x) = a_0 + a_1 \cos(\omega x) + b_b \cos(\omega x) \quad (3.3)$$

Where we can easily understand that the a_0 is approximately equals to the middle point of the i_0 -th column from the projection of i_0 -th scan, in which the center of sample coincides the center of rotation best. Thus, for obtaining the offset in each column, the offset of i -th column can be computed with:

$$\text{offset}(i) = f(i) - a_0 = a_1 \cos(\omega \theta) + b_b \cos(\omega \theta) \quad (3.4)$$

where we have

$$\theta = \frac{\theta_{\text{all}}}{n} * (i - 1) \quad i = 0, 1, \dots, n - 1 \quad (3.5)$$

for all projections. So after the alignment, the middle points of the sinogram columns would have the same vertical position, just like Figure 3.23 shows:

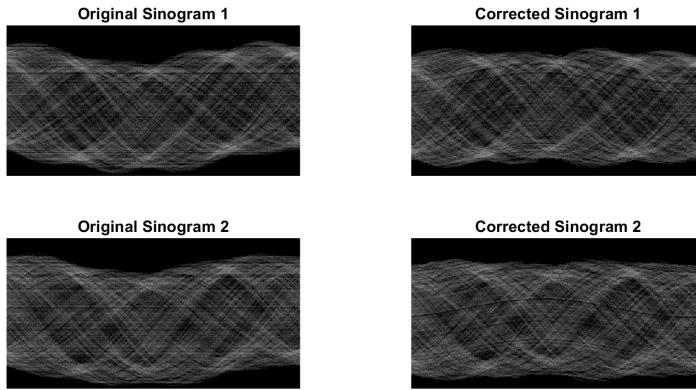


Figure 3.23: The aligned sinogram compared with the original one.

We could see that after the alignment, the sinograms are in the middle of the image, which means the off-center problem is solved. One thing we should notice is that after the correction, the stripes we mentioned in previous section got bending than before, which may influence the stripe removal result. Thus we suggest that the wavelet-FFT filter should be implemented before the off-center correction algorithm.

CHAPTER 4

Analyzing of 3D Volume

Image analysis is a process of extracting the meaningful, interesting or special information from a 2D image or 3D volume. After the CT scans and reconstructions by Zeiss system, we have obtained the 3D volume of a small part of rat brain. While looking at the slices and volumes of them, some points triggered our interest and reminded us some questions looking forward to study with. After all we decided to do the following analysis to the volumes: image segmentation ,and vessel analysis for the rat brain.

4.1 Image Segmentation Study

Image segmentation, just like how we call it, is segmenting the image into different sets based on the grayscale value, the shape, the content, etc. An article, [HS85], which is published in 1985, cast a definition of "good segmentation", which is the regions should be uniform and homogeneous, with respect to some characteristic such as the texture or the gray tone. So in our segmentation, we expect that the brain vessels could be independently displayed and saved, without containing other parts which has the same grayscale value as them. So with a good segmentation, the accuracy of our analysis could be ensured.

Nowadays a lot of articles, like [PP93], [Eff00] and [DD12], are trying to review the image segmentation techniques. And considering the high demand of methods with higher efficiency and performance in fields like computer vision and medical imaging, a lot of algorithms are discovered. Under a study of such methods, we find out some techniques which could be applied in our analysis process. One thing should be noticed is that our images are all in grayscale colormap, so the algorithms for processing the color images would not be considered by us.

4.1.1 Thresholding

The most comprehensive method in image segmentation is thresholding, considering we could frequently see images (including daily photos) showing that different part are having the different grayscale values. The main idea for this method is setting a value as the threshold T , and the pixels with value lower than T are assigned to one set, otherwise it is put in another set. A mathematical expression of this method is:

$$I(i, j) \in \begin{cases} S_1 & I(i, j) < T \\ S_2 & I(i, j) \geq T \end{cases} \quad (4.1)$$

We could easily understand that while using this method, the choice of T would be the main problem. Also, sometimes we cannot make sure that whether there exists a threshold making the two sets well distinguished. And in practical, it is highly possible that we have to find out more than one threshold, and segmenting the image into a number of sets. For this problem, our entry point is the histogram of the image, with which we could see the distribution of the pixels in the grayscale value bins.

During our theoretical study, we found that the Gaussian distribution in images are mentioned and discussed, like [GS98] and [BBM07]. Also, with the study of our brain images from Zeiss system, the Gaussian distribution are clearly seen in the histograms. Thus in such images, we are able to find the optimal threshold(s), which could be a valley between two peaks; and for obtaining an accurate value of such valleys, a gaussian fitting could be used for find the function representing the relationship between the grayscale values and the number of pixels in each bin. Then the local minimums could be easily calculated, and the ones between two local maximums would be the thresholds we are looking for.

4.1.2 Contextual Segmentation

Thresholding methods are segmenting the images with only grayscale values, but not taking the relationships of the pixel locations. However, in images it is highly possible that the pixels belongs to one element is neighboured or closed to each other. So The techniques take such cases in consideration are implemented, which are called contextual segmentation.

Region growing is a typical and generally used contextual segmentation technical. The main idea of this method is we choose one or some seeds in the image, and for each seed we find out whether the neighbours are having the same properties (like grayscale value) as the seed, then we check the outer neighbours (i.e. neighbours of neighbours) iteratively until no other neighboured pixels can be found [AB94]. An expression for the region growing with grayscale difference is:

$$\delta_{i,j} = |I(i,j) - \frac{\sum_{y \in R} y}{|R|}| \quad (4.2)$$

$$R = \begin{cases} R + (i,j) & \delta_{i,j} \leq \Delta \\ R & \delta_{i,j} > \Delta \end{cases} \quad (4.3)$$

Where R is the set of pixels which have already been found and ensured to be in one region, and Δ is the maximum grayscale value difference used to check whether one pixel is similar to the elements in R . In practical, we could use an FIFO queue as the data structure: the neighbours of current pixel are pushed into the queue, and the next pixel we are checking is the top member in the queue, so a Breadth-First Search(BFS) is applied.

There are also a comprehensive segmentation method implemented with the techniques of edge detection. The main idea of such method is firstly detect the edges in the image; after the edges are obtained, the regions surround by pairs of edges could be considered as the different objects.

Edge detection with operators is a general used technique, according to the reviews in [AAKK10]. For example, the Sobel operators, which is shown below, is a good and widely used tool for computing the partial derivation:

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Sobel operators are masks in the form of matrices. With using the Sobel operators as a filter, we are able to get the gradient G_x and G_y in vertical and horizontal direction. Then we can calculate the gradient magnitude Δf with:

$$|\Delta f| = \sqrt{G_x^2 + G_y^2} \quad (4.4)$$

Also, there are other operators which are also widely used in edge detection, like Roberts cross edge detector, Prewitt detector, etc. In practical each edge detector would give different results, so it is better for us to test them on the image and choose the one fits our expectation best.

Connected-component labeling, which is also called region extraction, is a widely used segmentation method as well. The main idea of this method is checking the connectivity of the pixels, and the pixels connected with each other in one blob would be a independent region. There are several version of this algorithm, which is introduced in [HCS08] and [Eff00]. The most simple one is one-component-one-time version:

1. Starts at one unvisited pixel and set it to an unused label l .
2. Check its neighbours. If one neighbour does not belong to the background, and has not been labeled, then label it by l as well.
3. Repeat step 2 until the neighbours are all backgrounds.
4. Starts with another unvisited pixel and run from step 1 to step 3 again until all unlabeled pixels are the background.

There are also another versions, like two-pass algorithms (also known by Hoshen–Kopelman algorithm) or the run-based two-scan algorithm suggested in [HCS08] are implemented. In MATLAB, there is a build-in function `bwconncomp(BW, conn)` supporting the connected-component detection. The theory of this function is introduced by Steve Eddins in [Edd], with the following steps:

1. Use `find` function to find the non-zero elements in a image.
2. Make a list of which nodes are connected to each other. Two nodes are connected with each other if and only if they are neighboured by each other.
3. With the pairs of connected pixels, a sparse adjacency matrix could be built. Then run the Dulmage-Mendelsohn decomposition function `dmperm` for the matrix.

4. Find out which block each element belongs to with the permutation vectors and block boundaries returned by *dmperm*

Above are the image segmentation we studied and planned to use for our brain images. In practical, more than one technique would be used in the analysis for one sample. In the following sections we will introduce how we are analyzing our sample.

4.2 Brain Vessel Analysis

4.2.1 Deleting the background

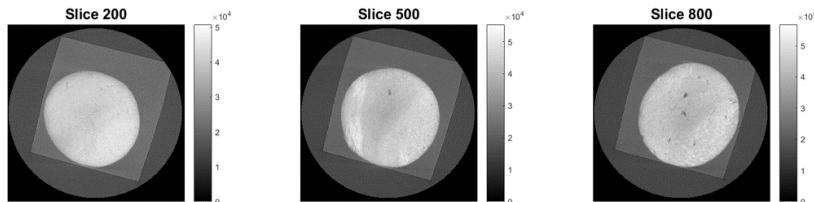


Figure 4.1: 3 slices of the brain volume. We could easily find that there are 3 parts in each image.

By looking at slices of the rat brain, which is shown in 4.1, it is easy for us to find that each slices has three parts: the outer big circle, the EPON and the brain. Considering it is clearly that these three parts are in different grayscale ranges, we could firstly have a look at the histogram of the volume:

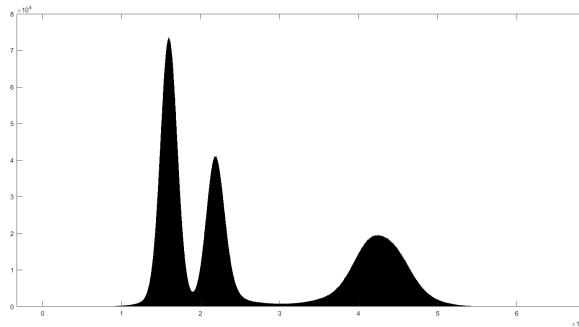


Figure 4.2: The histogram of the brain volume. X-axis is the grayscale values and Y-axis is the number of pixels in each bin.

In 4.2, a perfect condition for us is exposed: three gaussian distributions are contained by the full histogram, and each distributions is not mixed with another one too much. Considering the brain part in the slices are having the biggest grayscale values, the rightmost gaussian distribution in the histogram should represents the brain part. Fortunately, this distribution is far from the one on the left, so the threshold for segmentating this part could be directly obtained by finding the minimum between it and the one on the left.

Following the logic we introduced in previous section, we implemented a gaussian fitting to the data sets X and Y , where X is each grayscale value and Y represents how many pixels has value x . The plot of the fitting and the original curve is shown in Figure 4.3:

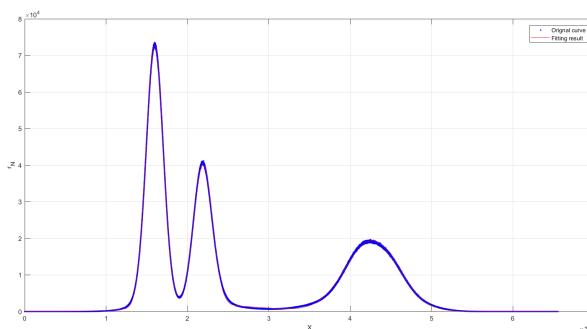


Figure 4.3: The original curve and the fitting graph. X-axis is the grayscale values and Y-axis is the number of pixels in each bin.

We can see that the fitting graph is covering the original curve well, and the coefficient of determination (R^2) which is 0.9986 also proved the credibility of the fitting. For calculating the minimum on the left of the rightmost peak, a theorem is used:

$$f'(x) = -f(x) \implies \text{peak}(f'(x)) = \text{valley}(f(x)) \quad (4.5)$$

Which means that we can run the build-in function `findpeaks(data)` in MATLAB with $-f(x)$ if the fitting result is $f(x)$. The output was 32316, and there were 899 pixels has this grayscale value. After we segmented the image with threshold 32316, we obtained a new volume, and Figure 4.4 shows some slices of it:

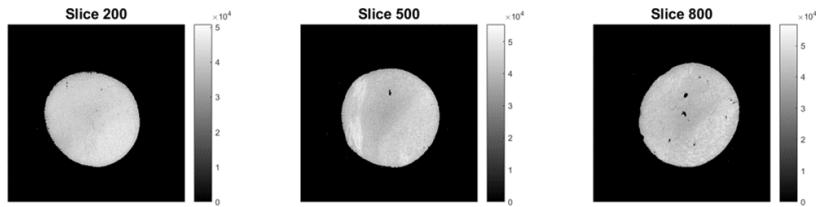


Figure 4.4: 3 slices of the segmented brain volume. The brain part were extract out and the background were deleted.

4.2.2 Extracting the vessel

With a discussion of the brain structure, we found that the black part in the brain should be the vessel:

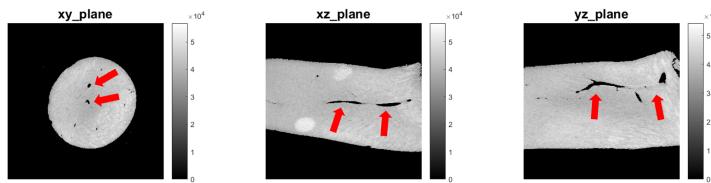


Figure 4.5: The vessel parts in the brain visible in slices from different directions.

Considering the vessel has the same gray scale value to the EPON background which is containing the brain sample (because the OsO₄ can not stain the vessel), and the vessel part in the brain is connected to the outside, we had to cut the part and keeps only the vessels in the data set. For cutting the outer part, we firstly chose the top slice, morphologically closed it with a disk structure element, and a mask was obtained. Then we extracted the part covered by the mask in each slice; however, the sample has an irregular shape, so there were still some outer part left in the volume, and connected with the biggest vessel part; so we attempted to cut some bottom slices until the outer part is invisible in the volume. Finally, the biggest three vessels in the sample, which is shown in Figure 4.6 is obtained by us.



Vessel 1



Vessel 2



Vessel 3

Figure 4.6: 3D shortcut of the vessels.

Besides the vessels, we also used the region growing algorithm, and extracted one of the tissue out from the brain volume. In the xz_plain slice in 4.5, there

are two big white dots, which are cross-sections of the two tissues.

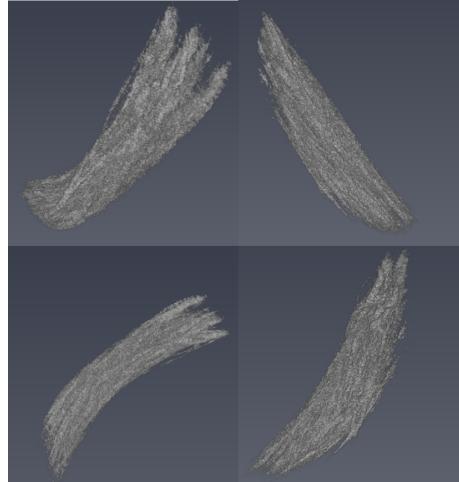


Figure 4.7: 3D shortcut of one of the tissues.

4.2.3 Visualizing the Vessel Orientation

In [Cha16], Lucie Chapelle introduced a technique for determining the fiber orientation, which may also be useful for the study of vessel. The tool suggested in the article is gradient structure tensor, which is a tensor matrix representing the partial derivatives in an image. In [KHB⁺10], a definition of strucutre tensor matrix is cast by M. Karuse, which is :

$$J_\rho(x) = K_\rho * \nabla I_\sigma(x) \nabla I_\sigma^T(x) \quad (4.6)$$

where

$$\nabla I_\sigma(x) \nabla I_\sigma^T(x) = \begin{bmatrix} I_x^2 & I_x I_y & I_x I_z \\ I_x I_y & I_y^2 & I_y I_z \\ I_x I_z & I_y I_z & I_z^2 \end{bmatrix}$$

I_x , I_y and I_z are the partial derivatives over the x , y and z axes respectively. And the formula for calculating the orientation vector is:

$$w(x) = \operatorname{argmin}_v [I(x + v) - I(x)]^2 = \operatorname{argmin}_v v^T J_\rho(x) v \quad (4.7)$$

And according to [KHB⁺10] it is easy to know that the eigenvector corresponding to the smallest eigenvalue determines the fiber orientation vector $w(x)$. Con-

sidering the orientation of each vectors could be described by the elevation angle θ and the azimuth angle ϕ , it is easy for us to find that the two angles, θ and ϕ , can be calculated with the following method if the eigenvectors are already known by us:

$$\phi = \frac{\pi}{2} + \arctan\left(\frac{w_y}{w_x}\right) \quad \text{and} \quad \theta = \frac{\pi}{2} + \arcsin(w_z) \quad (4.8)$$

Thus we can use the technique to visualize the orientation of the vessels. We chose the biggest vessel, which is called vessel 1 in Figure 4.6 as our research object considering it has most branches. From the results in 4.6 we could see that the vessel has very different radius in each part of it, so it is hard to find a perfect ρ value without giving incorrect results. Thus we down-scaled the volume, in order that there is a ρ closed to the radius in every part. The result of the vessel orientation is shown in Figure 4.8.

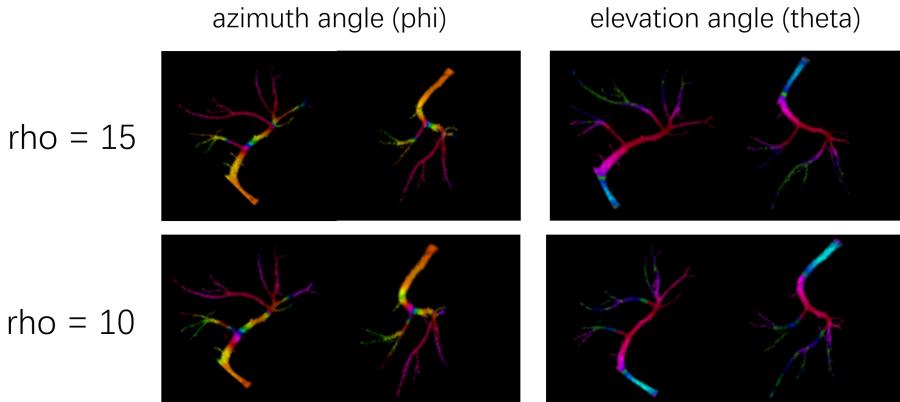


Figure 4.8: 3D shortcut of the colored vessel. The vessel volume is down-scaled, the size of it is $\frac{1}{64}$ of the original volume.

We can see that in the two results, most parts are colored well, but some branching points are experiencing the sudden change of the color. And considering we have used the same method for analyzing the fiber orientation, the approach should not be invalid. We tried to find out the method to fix this problem, but they doesn't work. Regarding this problem, we cast a hypothesis: the ρ value is hard to choose for orientation analysis of vessels. In [Cha16], the author stated that the error is resulted by the difference between ρ and radius of fibers r ; and the minimal error is obtained when $\rho = r$. Considering there are two differences between vessels and fibres: first, the vessels are branching in some position, but

the fiber sample we analyzed was not overlapped in 3-dimension space (in other words, crossing and inserting into each other); and the fibers has approximately the same radius, but vessel doesn't, considering big vessels are branching into small vessels. (In the 2d orientation of the fiber slices, there are also color sudden changes in the branching points.) That might be why analyzing the vessel orientation is harder than fiber.

CHAPTER 5

Discussion

5.1 Achievements and Limitations

This master project finished the works we outlined in the introduction. As a preparation, we have applied the projection preprocessing techniques. The bad pixel detection methods are well detected the bad pixels in the projections, and the correction approaches removed the bad pixels. The detection of the pixels could also provide a good reference to the technicians who are maintaining the equipment, as a report about the condition of the CT system. By the result of the bad pixel detection, the technicians can easily find out which units are experiencing errors, then fix the detector or the other components of the CT system (like the cooling system) and uses the detection method again in order to see how many bad pixels are reduced. In conclusion, this method might be helpful for the people who want to know how to improve the CT system and the quality of the images from it.

Among the bad pixel detection methods, the most recommended one is the slope method. In chapter 3, we have discussed that the slopes of the acquisition-time-count lines are the evidence for detecting the abnormal pixels. Moreover, the dead pixels, whose lines can be represented by $y = c$ where c is a constant number. So the slope of such lines is 0, which is also an abnormal value. Consequently, we find that the slope method could find both dead pixels and abnormal

pixels; that is the biggest advantage of this method.

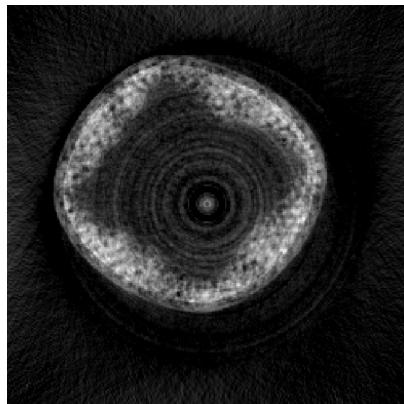
However, fixing the bad pixels could be a long and difficult work, and it is also hard to ensure all units working well in the detector. Then our bad pixel correction approaches could solve this problem, by using mathematical methods instead of waiting for the repair of the detector. The median filter and the stripe removal method can be used for removing the bad pixels in the projections without another scan; and the calibration could be implemented while we have the time for a new scan for fulfilling the pixels with wrong values. The calibration method is most suggested by us, considering it is giving the pixels real values instead of guessing the values with the mathematical tricks.

Nonetheless, there is still some of limitations to the methods. Firstly, the method might not work while there are too many bad pixels. For the mathematical approaches, the calculation of the output relies on the values of the pixel's neighbour. If there are too many other bad pixels lied on the neighbour of the correcting one, the value we got is hardly approximated to the actual value there should be. For example, a extreme condition is that while we are using the the median filter, and in the window all pixels are bad pixels, then there would be no reference to the pixel for the bad pixels to be corrected.

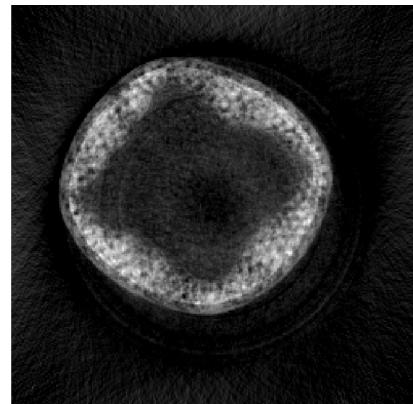
In such conditions, attempts of the correction is not better than waiting for the maintenance of the equipment, considering no matter we correct it or not, the projection is losing too many information, sometime it might be the critical information. Just imagine that in some slices the small vessel part is covered by the bad pixels, then it is impossible to get the vessel images reflecting the real shape of it.

The stripe removal method also has a limitation, which is also discussed by Beat Munch in [MTMS09]. The wavelet-Fourier filter is reducing the stripes by damping some information in the sinogram. Idealy, we expect it only erases the stripes, and do nothing to the good part; however, in practical the algorithm is both removing the stripes and the useful information. From the result of our test, we could see that the increasing of the decomposition level, vanishing moment and sigma, makes more good information lost. Furthermore, after the correction, we could see blurred fat rings in the reconstructed image. In 5.1, the reconstruction result of the stripe-free sinograms of match head, original ring artifacts are not covering too many useful information, so the result looks not bad; however, in the projections of the brain sample shown in 5.2, the ring artifacts are covering the brain part so after the damping of the sinogram, the brain parts are hardly to see considering the fat rings are bloking them.

The limitation of the calibration is easy to comprehend. Considering we are using the values in the second scan to correct the projections from the first

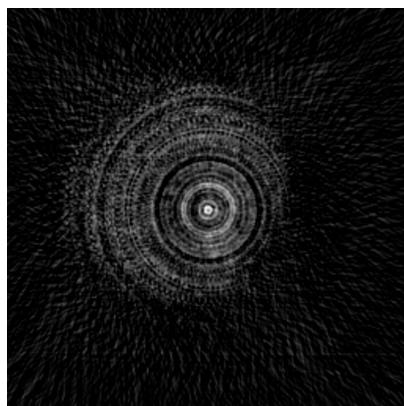


(a)

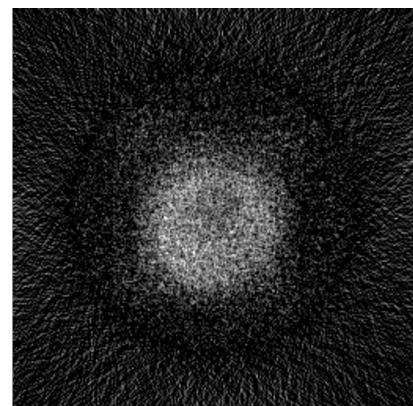


(b)

Figure 5.1: The original match head slice (a) and the one reconstructed with a stripe-free sinogram (b).



(a)



(b)

Figure 5.2: The original (a) and stripe-free-reconstructed brain slices (b).

scan, there is a precondition that the a pixel in the second scan is not bad if it is used for correcting a bad pixel in the first scan; otherwise, we are using a bad pixels to correct the other bad pixel. In order to achieve this precondition, there should not be so many bad pixels in the projections. Furthermore, in some institutions a high utilization of the X-ray CT equipment is frequent, so this approach which is using a longer time for one experiment might not be welcomed.

The performance of our off-center algorithm works good regarding to our simulation and the result. After the correction, the match head is located well in the center of the image. Furthermore, the stripes in the sinogram are bent after the off-center correction, so we highly suggest that the stripe-removal algorithm should be implemented before the off-center correction considering there would be hardly stripes in the aligned sinogram.

With the experiment testing the performance of image enhancement techniques, we found that gamma correction and gray stretch sometimes can give good results, and sometimes cannot help so much with the contrast. The effect of the techniques has already shown in the match head enhancement results; however, the limitation is also visible in the attempts with brain sample. Thus we could see a limitation of this method: the performance relies on the original images to some extend. It is easy to understand: if we want to enhance the contrast of two parts in one image, at least the two parts should have difference on grayscale values. Otherwise reducing one part would also make another one invisible.

Unfortunately, no matter how we process the projections, the clarity (SNR or CNR) of the brain images could not support our requirement for the analysis, so we have to use the Zeiss system instead. This tells us that before planning the image analysis, we have to evaluate the performance of the CT equipment; it is not always impossible to see the structure with low SNR images, considering there are too many abnormal values. Thus, we cannot ensure that the analysis result is right or useful.

By looking at the 3D image of the vessel in brain sample, we could say that our segmentation method works well for extracting the vessel out, even though the connection between the vessel and the background influenced us a lot. In such images which has isolated parts in the histogram, using Gaussian fitting for calculate the threshold is a good idea. Also, by using the region growing and connected-component methods, we could both pick the vessel out; however the region growing is an iterative algorithm, it takes longer time than the connected-component labeling. For segmenting out the vessels from the volume contains a lot of connected components, the labeling algorithm is a better choice; however while we want to segment something from a connected image, like the white tissue, region growing is more recommended.

Based on the orientation analysis result, we suggest a hypothesis that the choosing of ρ for the Gaussian filter is a big problem in vessel orientation. The difference of the radius in parts of vessel and the branching may affect the performance of vessel orientation analysis. So in conclusion, we suggest some preprocessing of the vessel before analyzing.

5.2 Future Work

In this thesis we used a sample stained with OsO₄, giving the components in the white matter higher grayscale values except the vessels. However from the theory study, we have known that there are some reagents which are able to stain the vessels, like hematoxylin. So the future work could focus on segmenting the white matter with hematoxylin, segment the blood vessel out and analysis the performance of it.

In previous section, we mentioned the hypothesis in order to explain the reason why the result in the orientation analysis is not good enough. Thus, future work to prove or falsification the hypothesis is expected. And if it is proved, the method of finding out how we should analyze the orientation could be watched for as well. We also look forward the studies on the structure of the white matter, like the nerves or other tissues. Other characterization of the vessels in white matter, like the diameter or the density, is also going to be studied.

Another future research is to study more about the bad pixel detection and correction method, including defining the limitation of the methods quantitatively. For example, we have know that if there are too many bad pixels, the calibration method would not work, so we could find a number or expression in order to find the maximum number or density of the bad pixels in a projection (detector).

Finally, we are interested in whether the off-center correction could helps the bad pixel avoidance. For example if we can find a region in the detector without bad pixels, we would be able to keep the sample scanned in that region. However if the sample is off-centered, during the rotation the sample part in the projection might contains some bad pixels. If we could calculate the offset, and move the sample in every scan in order to keep the sample part in the region without bad pixels, we are able to avoid the impact of the bad units on the detector.

5.3 Conclusion

To conclude, the works of this master project are valuable for the future use of the DTU PCXCT system and the forthcoming studies about the white matter in the rat brain. The method we attempted to process the projections, that is, optimizing the result of the reconstructions, is also a good reference for the other users of that system. We sincerely hope that our result could help the users of the DTU PCXCT system obtain better image sets, and help the maintainer find out how to improve it.

APPENDIX A

Other Figures

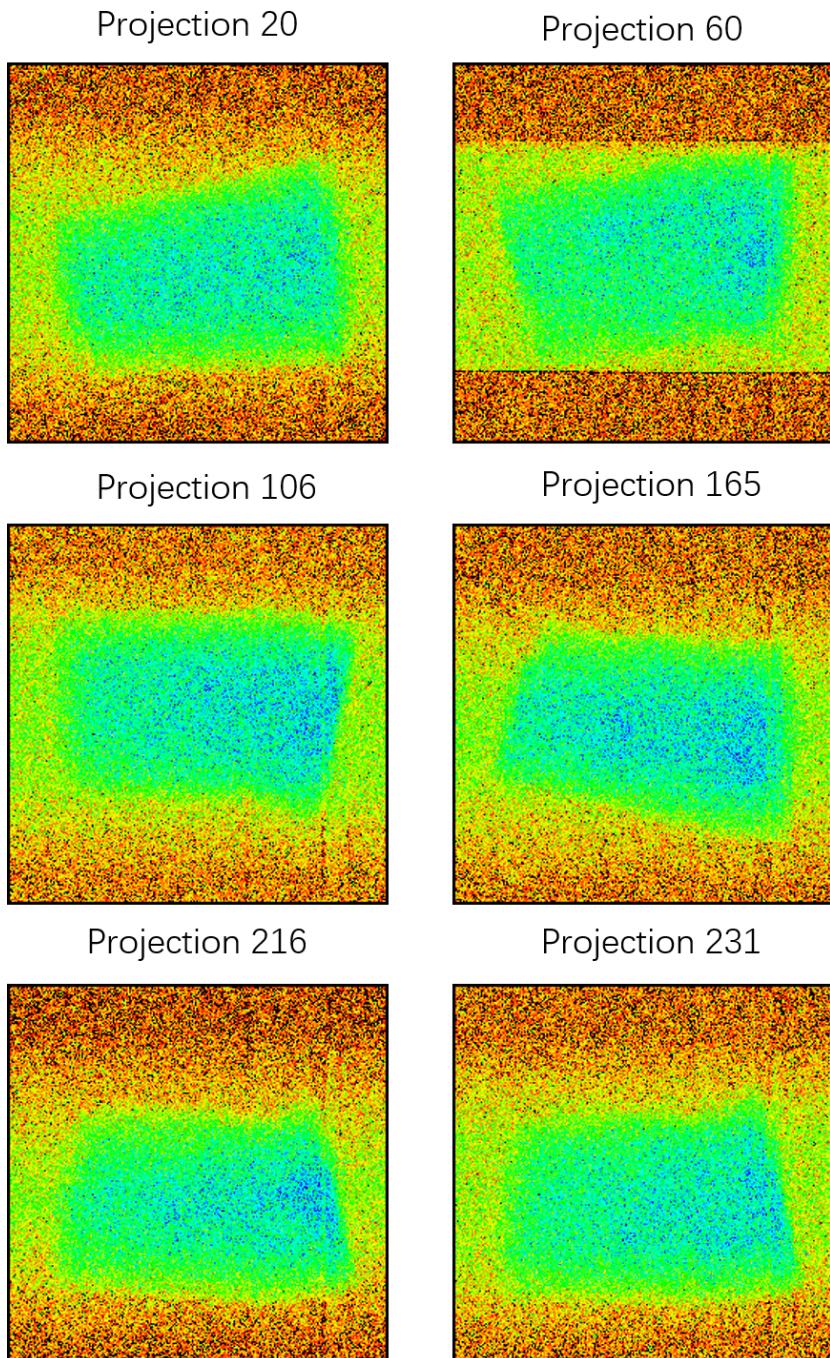


Figure A.1: Some projections of brain samples taken by DTU PCXCT system, with 1-80 KeV energy bin. In order to see the projections clearer, we colored the images of them.

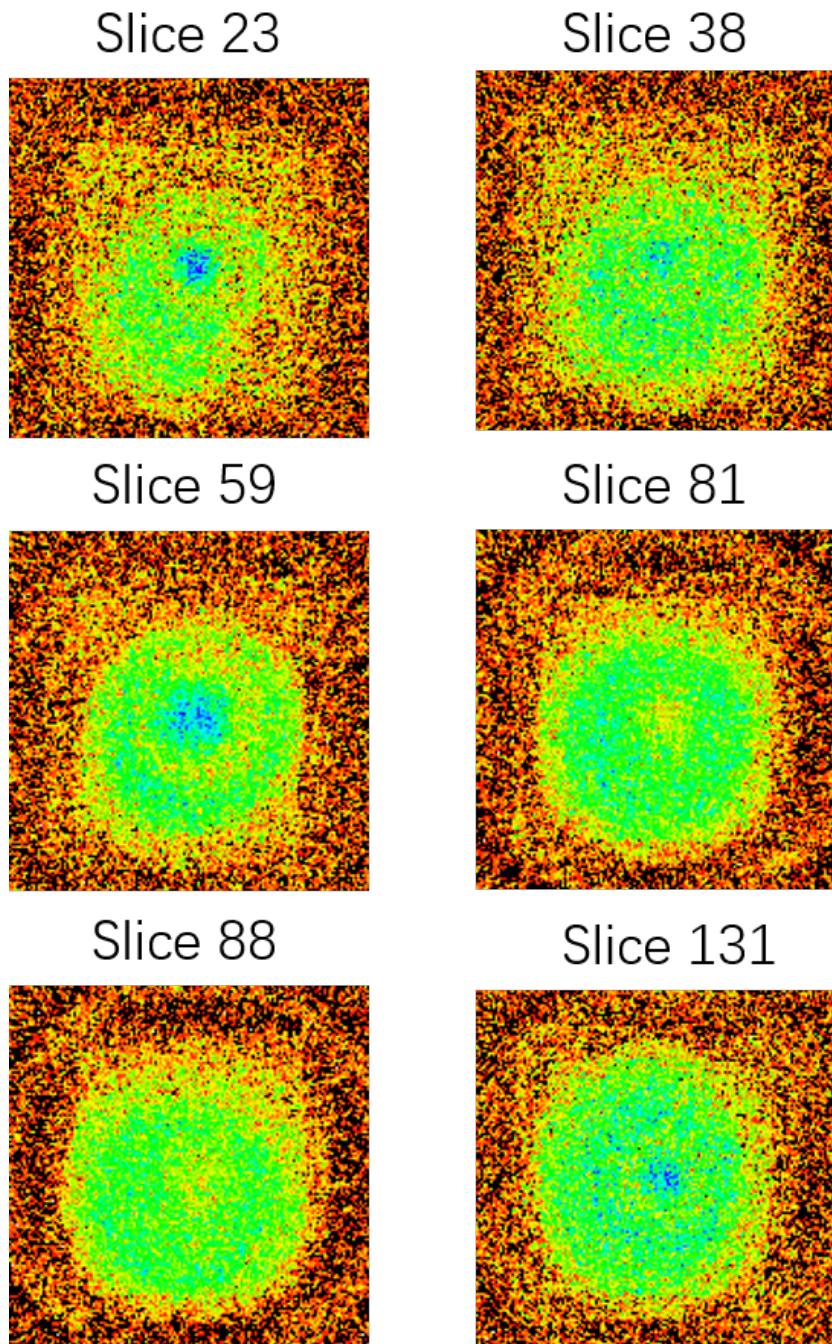


Figure A.2: Reconstruction result with the slices. In order to see the slices clearer, we colored the image and deleted some background noises.

Bibliography

- [AAKK10] Salem Saleh Al-Amri, NV Kalyankar, and SD Khamitkar. Image segmentation by using edge detection. *International Journal on computer science and engineering*, 2(3):804–807, 2010.
- [AB94] Rolf Adams and Leanne Bischof. Seeded region growing. *IEEE Transactions on pattern analysis and machine intelligence*, 16(6):641–647, 1994.
- [BBM07] Yakoub Bazi, Lorenzo Bruzzone, and Farid Melgani. Image thresholding based on the em algorithm and the generalized gaussian distribution. *Pattern Recognition*, 40(2):619–634, 2007.
- [BDHS16] Ander Biguri, Manjit Dosanjh, Steven Hancock, and Manuchehr Soleimani. Tigre: a matlab-gpu toolbox for cbct image reconstruction. *Biomedical Physics & Engineering Express*, 2(5):055010, 2016.
- [BKZV04] Izhak Baharav, Ramakrishna Kakarala, Xuemei Zhang, and Dietrich Werner Vook. Bad pixel detection and correction in an image sensing device, May 18 2004. US Patent 6,737,625.
- [Cha16] Lucie Chapelle. Characterization and modelling of the mechanical properties of mineral wool. 2016.
- [DD12] Rajeshwar Dass and Swapna Devi. Image segmentation techniques 1. 2012.
- [Edd] Steve Eddins. Connected component labeling. <https://blogs.mathworks.com/steve/2007/03/20/connected-component-labeling-part-3/>. Accessed: March 20, 2007.

- [Eff00] Nick Efford. *Digital image processing: a practical introduction using java (with CD-ROM)*. Addison-Wesley Longman Publishing Co., Inc., 2000.
- [FDK84] LA Feldkamp, LC Davis, and JW Kress. Practical cone-beam algorithm. *JOSA A*, 1(6):612–619, 1984.
- [GDCXJ14] Doga Gürsoy, Francesco De Carlo, Xianghui Xiao, and Chris Jacobsen. Tomopy: a framework for the analysis of synchrotron tomographic data. *Journal of synchrotron radiation*, 21(5):1188–1193, 2014.
- [Gil72] Peter Gilbert. Iterative methods for the three-dimensional reconstruction of an object from projections. *Journal of theoretical biology*, 36(1):105–117, 1972.
- [GLLS14] Sahar Ghanavati, X Yu Lisa, Jason P Lerch, and John G Sled. A perfusion procedure for imaging of the mouse cerebral vasculature by x-ray micro-ct. *Journal of neuroscience methods*, 221:70–77, 2014.
- [Gon16] Rafael C Gonzalez. Digital image processing, 2016.
- [GS98] Lalit Gupta and Thotsapon Sortrakul. A gaussian-mixture-based image segmentation algorithm. *Pattern Recognition*, 31(3):315–325, 1998.
- [HCS08] Lifeng He, Yuyan Chao, and Kenji Suzuki. A run-based two-scan labeling algorithm. *IEEE transactions on image processing*, 17(5):749–756, 2008.
- [HLC⁺] Robert S. Hill, Wayne B. L, Hughes Stx Corp, Don Lindler, and Richard Shaw. S. casertano, et al., eds. cosmic ray and hot pixel removal from stis ccd images.
- [HS52] Magnus Rudolph Hestenes and Eduard Stiefel. *Methods of conjugate gradients for solving linear systems*, volume 49. NBS Washington, DC, 1952.
- [HS85] Robert M Haralick and Linda G Shapiro. Image segmentation techniques. *Computer vision, graphics, and image processing*, 29(1):100–132, 1985.
- [JMLA02] Javier Alda Jose Manuel Lopez-Alonso. Bad pixel identification by means of principal components analysis. *Optical Engineering*, 41:41 – 41 – 6, 2002.

- [KHB⁺10] Michael Krause, Jan-Marcel Hausherr, Bernd Burgeth, Christian Herrmann, and Walter Krenkel. Determination of the fibre orientation in composites using the structure tensor and local x-ray transform. *Journal of Materials Science*, 45(4):888, 2010.
- [KSW02] Avinash C Kak, Malcolm Slaney, and Ge Wang. Principles of computerized tomographic imaging. *Medical Physics*, 29(1):107–107, 2002.
- [Lyu18] Jinning Lyu. Experimental study of feasibility of multi-energy k-edge imaging using photon-counting detectors. 2018.
- [Mas09] Mario Mastriani. Denoising and compression in wavelet domain via projection onto approximation coefficients. 35, 11 2009.
- [MTMS09] Beat Münch, Pavel Trtik, Federica Marone, and Marco Stampanoni. Stripe and ring artifact removal with combined wavelet — fourier filtering. *Opt. Express*, 17(10):8567–8591, May 2009.
- [NWX⁺10] Peter B Noël, Alan M Walczak, Jinhui Xu, Jason J Corso, Kenneth R Hoffmann, and Sebastian Schafer. Gpu-based cone beam computed tomography. *Computer methods and programs in biomedicine*, 98(3):271–277, 2010.
- [PBS13] Willem Jan Palenstijn, K Joost Batenburg, and Jan Sijbers. The astra tomography toolbox. In *13th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE*, volume 2013, pages 1139–1145, 2013.
- [PP93] Nikhil R Pal and Sankar K Pal. A review on image segmentation techniques. *Pattern recognition*, 26(9):1277–1294, 1993.
- [RMS14] Peter Wizinowich Roger M. Smith, David Hale. Bad pixel mapping, 2014.