ENSC 477/895

Biomedical Image Acquisition

Lab 2 – Simulated Computed Tomography (CT)

Lestley Gabo 301170055 lgabo@sfu.ca
Yousra Wakil 301298273 ywakil@sfu.ca
Karen Ly-Ma 301148442 klyma@sfu.ca

1 Introduction

To generate a Computed Tomography (CT) image, an object is exposed to x-ray beams from a source. A detector is aligned opposite to the x-ray source and an object is then placed between the source and the detector. Every image that the source receives is an x-ray projection. These projections are collected to reconstruct a 2-D cross section of the object. The source and the detector is then rotated around the object in order to generate a CT image.

The objective of this lab is to acquire the data of a simulated CT measurement. For our lab we are only simulating the generation of a CT image. Figure 1 below is the setup of this simulation and it is done by using an optical system and total absorbers.

The purpose of using a simulated CT measurement is to expose us to the concepts and principles of CT without being exposed to X-rays. Compared to a real CT scan in which the object is stationary while the source and detector rotates around, only the object rotates in this simulation. We will use an optical ray system that hits a rotating object with a laser and then this data is recorded by a detector. We then use a software to choose by how many degrees the sample should rotate, we will use 180º, and by how much degrees of rotation (we should get better resolutions with lower degrees) the object should have.

In this lab we will gather the projection data and then graph our results as a sinogram. We will then reconstruct the sinogram into an unfiltered and filtered image through backprojection. Next, we are given an x-ray sinogram which we will reconstruct an image from and add noise into it to observe the effects of dose and noise on its signal-to-noise ratio (SNR). Lastly, from a dataset provided we will obtain a volumetric CT data set, an axial slice, coronal slice, and a sagittal slice using a medical image visualization software.

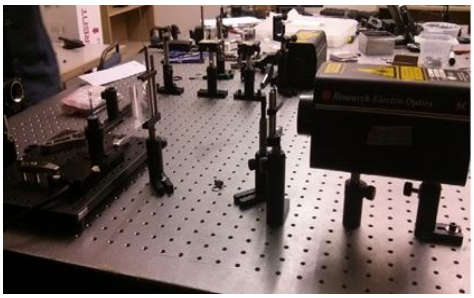

Figure 1: The diagram for the system

2 Methods

2.1 Sinogram

To begin the lab, we process the projection data obtained from using optical rays on a sample rotated at angles from 0° to 180°. We imported the sinogram data we acquired and utilized the command '*imagesc*' to recreate the sinogram in MATLAB. Sinograms for step size of 1 and step size of 5 can be seen in Figure 7 and Figure 8 respectively.

2.2 Backprojection

In order to reconstruct the image we utilize the method of backprojection. If we let $g(l,\theta)$ be the 2-D Radon transform of a function $f(x,y)$ we cannot determine $f(x,y)$ uniquely from a single projection. In order to create an image we simply assign every point on $L(l_o,\theta_o)$ the value $g(l_o,\theta_o)$ and repeating this for all l we obtain the function of the backprojection image as shown in equation (1) [5].

$$b_\theta(x,y) = g(x\cos\theta + y\,\sin\theta,\theta) \tag{1}$$

As stated above, this one projection is not enough to see our image therefore we add up all of the backprojection images from 0° to 180°. This summation image is called the *laminogram*, given by equation (2) [5].

$$f_b(x,y) = \int_0^{\pi} b_\theta(x,y)d\theta \quad (2)$$

To reconstruct the image, we utilize this theory of backprojection and work in spatial domain. The sinogram data we acquired in the lab as plotted above is the image of $g(l,\theta)$. This means we already have the projection of the object at various angles.

To do backprojection, we take all 1362 projection data points at an angle, and copy the data 1362 times to create a 1362×1362 at that specific angle with reference to angle 90°. This mimics the effect of smearing the projection back across the image at the acquired angle. As mentioned in the theory, we need to repeat the process and obtain projections at angles from 0° to 180° in order to have enough information and increase the quality of the reconstructed image.

Finally, as seen in equation (2) we sum up all the 180 images at the various rotations and the image is reconstructed. Projection data of where the optical ray hits the image will be overlapped in the 180 images and summed together to create a higher pixel value than when the ray misses the image. Using '*imagesc*' command again, the image is reconstructed as seen in Section 3.3.

2.3 Filtered Backprojection

In this section we add a filter to reduce the blurriness of the reconstructed image. Since most of the blurriness are at low frequencies, we apply a high pass filter, specifically a Ram-Lak filter to our reconstruction.

The backprojection is done in spatial domain but to simplify the process of applying the filter, we first work in temporal domain then revert back to spatial domain. The projection data at an angle is converted to the temporal domain by using Fourier transformation. The constructed ramp filter is shown below in Figure 2.

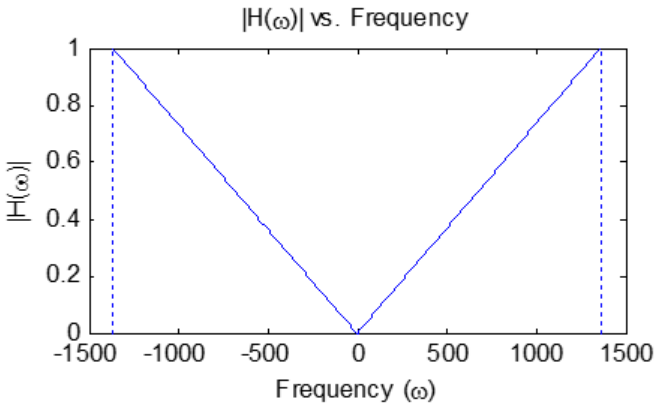

Figure 2: Ram-Lak Filter

Since we are in temporal domain, to apply the filter we can simply multiply each transformed projection data at an angle with the filter. We repeat this process for all the rotational angle positions we obtained during the lab between 0° and 180°. We do this for both the data of step size 1 and step size of 5. Once all the projections have been filtered, the inverse transform is applied on the projections of each angle to get back into spatial domain. Lastly, the backprojection process we discussed in Section 2.2 is applied to create a filtered reconstructed image. The results are seen in Section 3.4.

2.4 Reconstructing Image of Using Sinogram from X-rays

In the first case we used a sinogram produced from optical rays, we now look at reconstructing an image using a sinogram created by x-rays.

We use the '*imread*' function to acquire the sinogram data from the bmp file. We then apply unfiltered and filtered backprojection as discussed in Section 2.2 and Section 2.3 respectively to reconstruct the image.

Next, we add noise to the image to create a more realistic medical image. This process is done by using the function '*imnoise*' and applying a Gaussian type of noise. To control the noise level and keep it at a low amount, we set the mean at zero and varied the variance until our image is still recognizable. We decided that a variance of 0.0001 provided a sufficient amount of noise to our image.

Finally, the original data is divided by a constant to observe the effect of using a lower dose in the presence of noise. The unfiltered and filtered backprojection in Section 2.2 and Section 2.3 is once again applied. The various results for the reconstructed image of the sino1.bmp file is seen in Section 3.5.

2.5 Medical Image Visualization

We used the open source image visualization software FIJI. This software is based on ImageJ and is used for scientific image processing and analysis. Moreover, ImageJ is designed for scientific multidimensional images, and is very extensible [1][2].

We downloaded a dataset of a CT x-ray image of a person's chest showing their lungs [3]. The dataset was then imported to the FIJI visualization software. Using the Volume Viewer, we are able to view the volumetric CT data set, axial slice, coronal slice, and a sagittal slice. These results are shown in Section 3.6.

Table 1 below identifies each type of slice we observed in Volume Viewer. Figure 3, Figure 4 and Figure 5 shows the type of slices from an MRI scan of the brain [4].

Table 1: Representation of Various Slice Views

Axial Slice	The axial slice separates the bottom from the top (top view).
Coronal Slice	The coronal slice separates the front from the back (frontal view).
Sagittal Slice	The sagittal slice separates the left and right sides (lateral view).

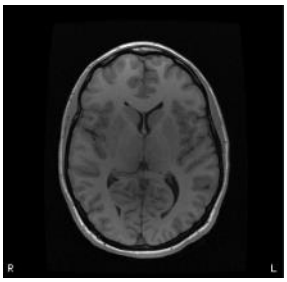

Figure 3: Axial Slice

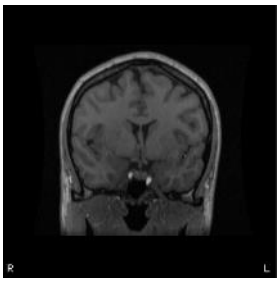

Figure 4: Coronal Slice

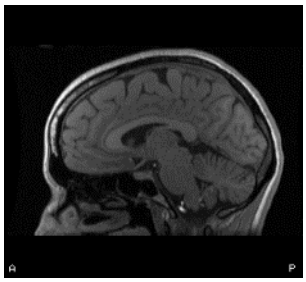

Figure 5: Sagittal Slice

3 Results

3.1 Sketch of Object

During the lab we observed the sample object being attenuated by optical rays. Figure 6 is a sketch of the top down view of the sample.

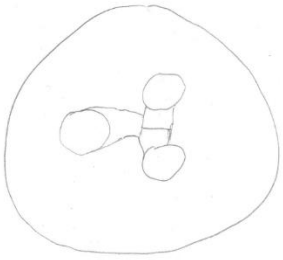

Figure 6: Top down view sketch of sample

3.2 Sinogram

In Section 2.1, we used the function '*imagesc'* to plot the sinogram in MATLAB for both projection data with step sizes of 1 and 5. The resulting sinograms are seen in Figure 7 and Figure 8 respectively. In the first figure, we obtain a projection data over 180 rotation positions, and the second we obtain data over 36 rotation positions from 0° to 180°. There are 1362 detectors used for acquiring the projection data over the image thus we have 1362 projection points as seen in the sinograms below. The results are discussed in Section 4.2.

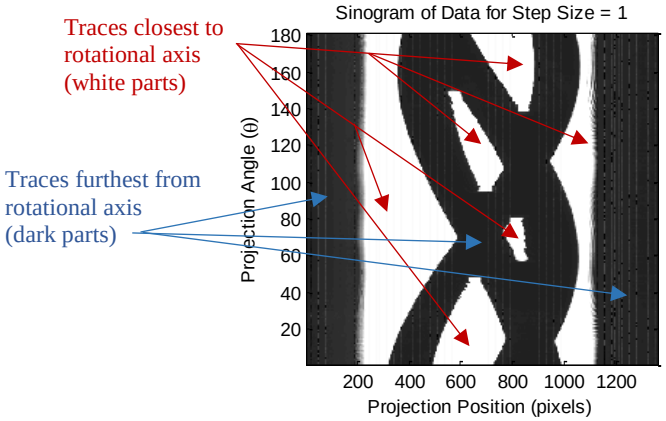

Figure 7: Sinogram image of step size = 1

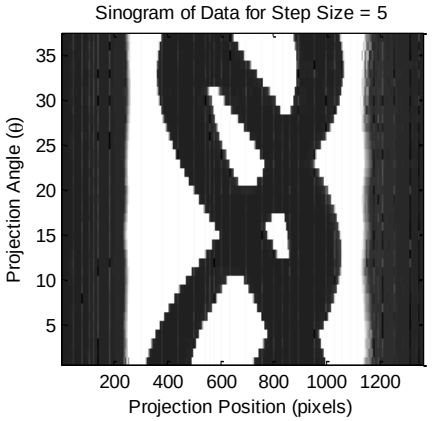

Figure 8: Sinogram image of step size = 5

3.3 Backprojection

From the backprojection method discussed in Section 2.2, the results of the reconstruction image in the lab for projection data of step sizes of 1 and 5 are seen in Figure 9 and Figure 10 below.

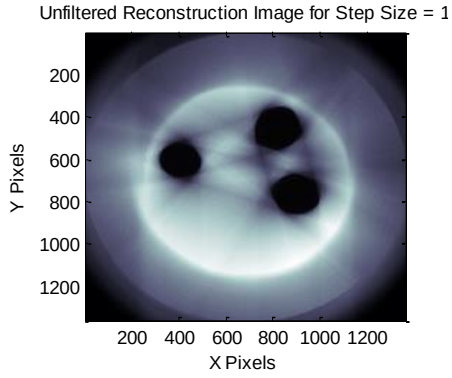

Figure 9: Unfiltered image with step size = 1

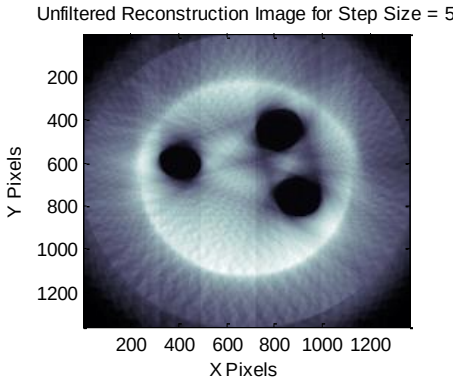

Figure 10: Unfiltered image with step size = 5

3.4 Filtered Backprojection

The results from adding the ramp filter to our projection data as discussed in Section 2.3 for step sizes 1 and 5 can be seen in Figure 11 and Figure 12 respectively.

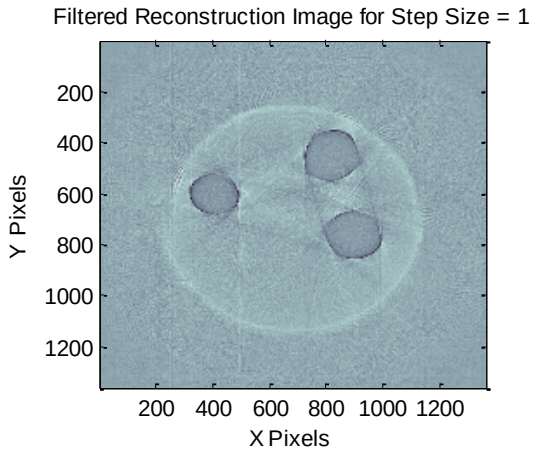

Figure 11: Filtered image with step size = 1

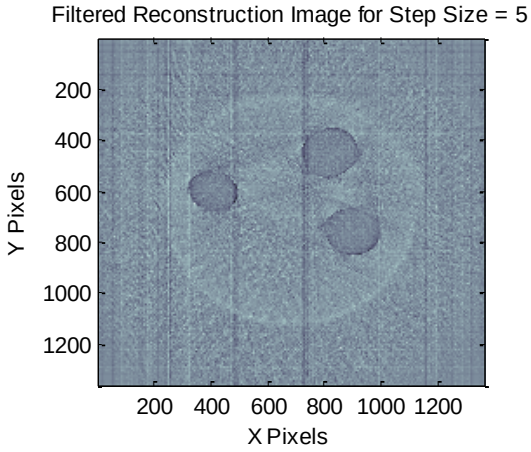

Figure 12: Filtered image with step size = 5

3.5 Reconstruction of Using Sinogram from X-rays

We reconstruct the image for a sinogram data obtained from using x-rays as described in Section 2.4. The first set of images, Figure 13 and Figure 14, display the reconstructed image for unfiltered and filtered backprojection respectively.

The results from adding some Gaussian noise on to the original unfiltered and filtered images are seen in Figure 15 and Figure 16.

The effects on the unfiltered and filtered image from lowering the dose by various constants in the presence can be seen from Figure 17 to Figure 20. Figure 17 and Figure 18 are results from lowering the dose by two times for the unfiltered and filtered image respectively. Figure 19 and Figure 20 shows results for lowering of the dose by 5 times. Finally, Figure 21 and Figure 22 displays the results for when the dose is lowered by 10 times.

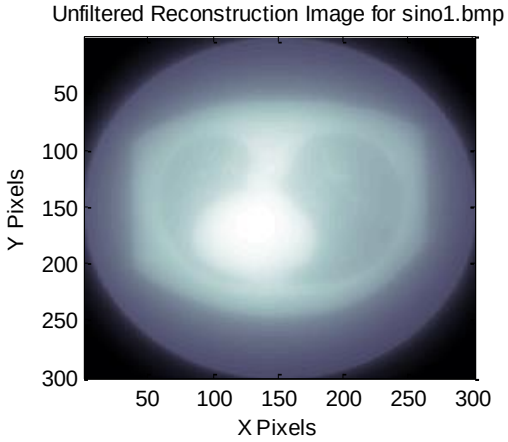

Figure 13: Unfiltered image with no noise

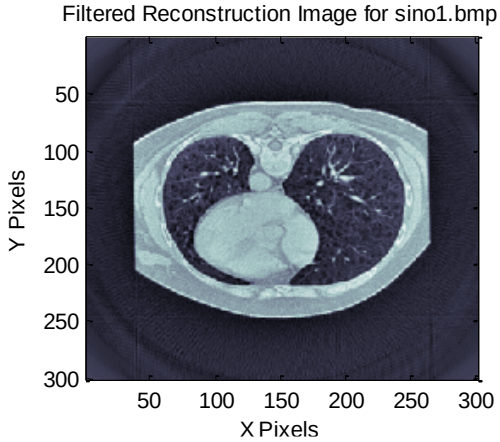

Figure 14: Filtered image with no noise

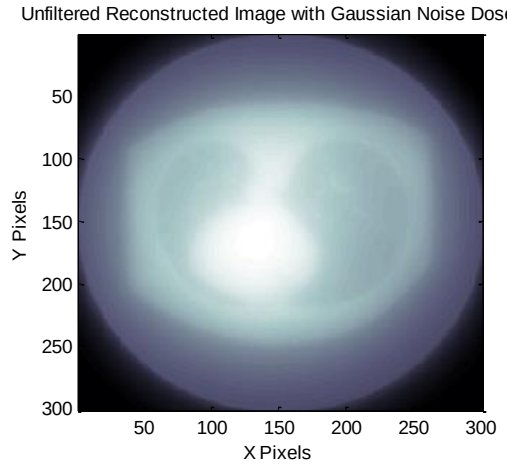

Figure 15: Unfiltered Image with Gaussian Noise Dose=1

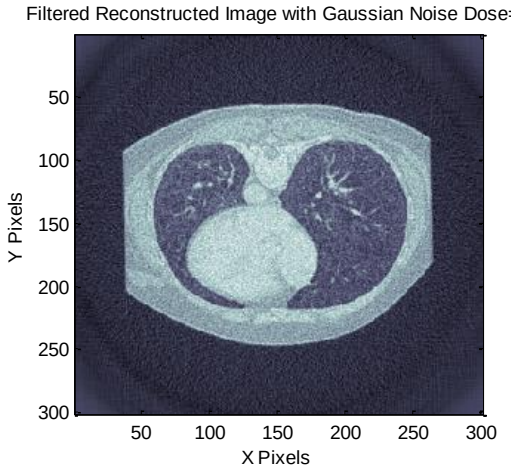

Figure 16: Filtered Image with Gaussian Noise Dose=1

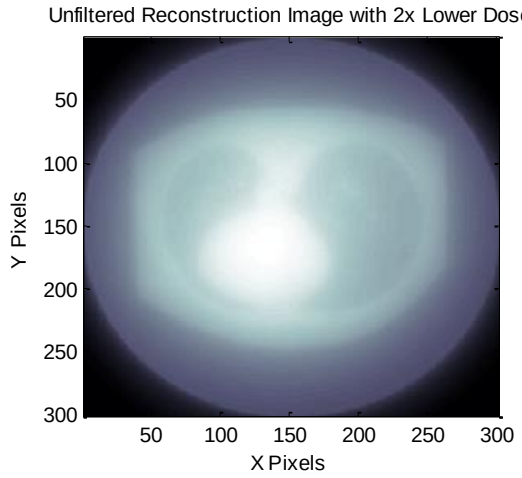

Figure 17: Unfiltered image with 2x dose

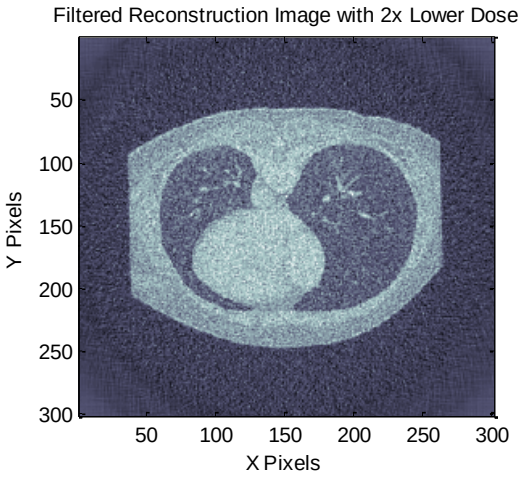

Figure 18: Filtered image with 2x dose

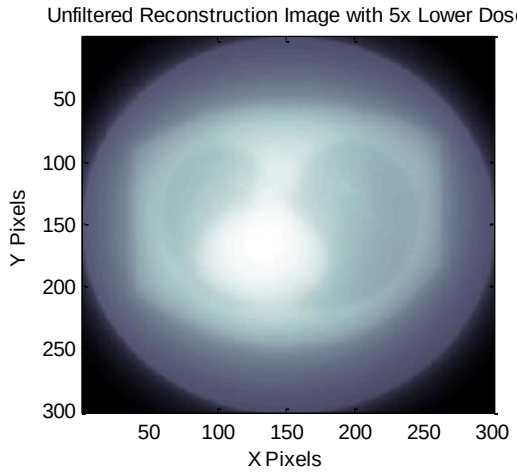

Figure 19: Unfiltered image with 5x dose

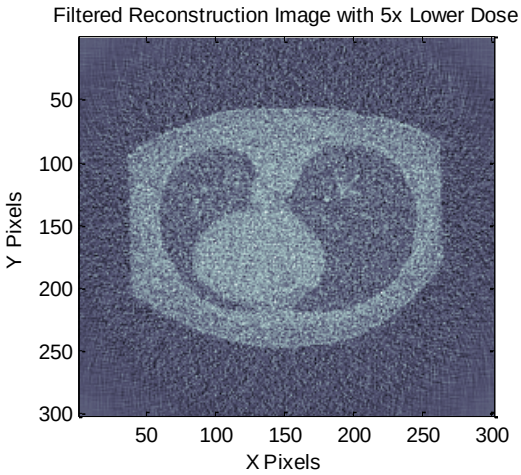

Figure 20: Filtered image with 5x dose

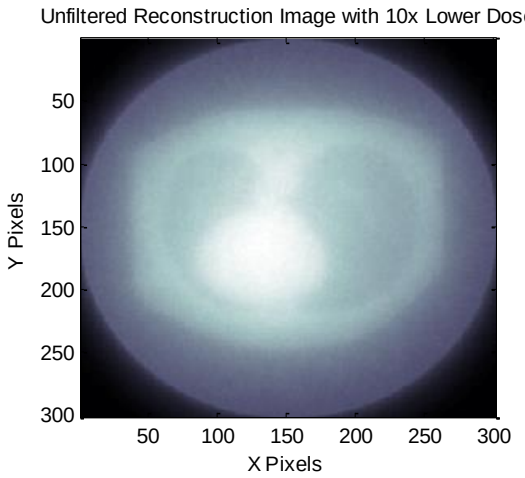

Figure 21: Unfiltered image with 10x dose

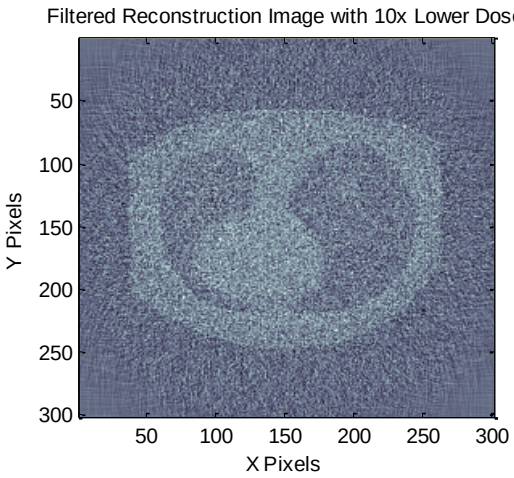

Figure 22:Filtered image with 10x dose

3.6 Medical Image Visualization

The results obtained by the medical image visualization of the CT chest x-ray by FIJI can be seen below. Figure 23 illustrates the image acquired by importing the sample to FIJI. The results for the volumetric CT dataset, an axial slice, coronal slice, and sagittal slice are shown by Figure 24, Figure 25, Figure 26 and Figure 27respectively.

Figure 23: Acquired Image of the sample

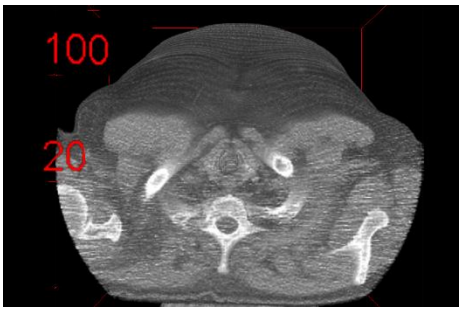

Figure 24: Volumetric CT dataset

Figure 25: Axial Slice

Figure 26: Coronal Slice

Figure 27: Sagittal Slice

4 Discussion

4.1 Comparison of Sketch with Reconstructed Image

For the sketch we see the top of the heads as three circles, which are then connected to the three legs and we see the legs merging together between the three heads. This is different from the reconstructed image that shows only three circles because the reconstructed image is an image of a straight cut in top down view of the object (axial slice). While the hand sketch is a view of the object from the top looking down.

The reconstructed image does not show the legs merging because that is not where the laser beams hit the object, the laser beams are rotated about the object where the three legs are separated and therefore we see three circles, with each circle representing a leg.

4.2 Sinogram

Comparing the two figures Figure 7 and Figure 8, we can see that with a smaller step size, there is more resolution. This make sense since we acquire more projection data at more angles between 0° to 180°. For a step size of 1, we obtain projection data over 180 rotation positions, but for a step size of 5, we only obtain data over 36 rotation positions. Since there is more projection data, we can expect the data of step size 1 to have a clearer reconstruction image than with the data for step size of 5. There were 1362 detectors and thus we have 1362 acquisitions of data per angle Figure 7.

As labelled in Figure 7, the traces that correspond to the nearest objects from the rotational axis is the white area in the sinogram. These traces are the objects that are hitting the target, and result in brightness in the sinogram. On the other hand, the furthest objects from the rotational axis is the darker part of the sinogram image which gets attenuated.

4.3 Backprojection

As seen in Figure 9 and Figure 10 the resolution of the reconstructed image with step size of 1 is higher than with step size 5. Figure 10 is more pixelated similar to what we saw for the sinogram for step size 5. As discussed in Section 4.2, since we only have projection data for 36 rotation positions from 0° to 180° we expect to a more pixelated image in comparison to when we have 180 rotation positions. Having more step size is more ideal in creating a clearer image.

We can also note that both images appear very blurry. Blurriness is expected since the low frequencies, where pixel changes less over space, is also attained in the image. To remove this we add a filter as discussed in the next section.

4.4 Filtered Backprojection

Since the reconstructed images Figure 9 and Figure 10 are blurry, we want to add a filter. High frequencies reside in the areas where there are drastic changes in pixel intensities. These regions create sharper edges in comparison to low frequencies where pixel intensity changes less over space creating a more blurred effect. This is the reason why we want to use a Ram-Lak filter (ramp filter) because it allows high frequencies to pass and blocks off the blurriness caused by low frequencies. Also note that we had to use the *'fftshift'* command to move the zero frequency component of our ramp filter, this would be equivalent to using an inverted ramp, since are essentially just swapping the positive and negative slopes of our ramp.

By adding the Ram-Lak filter, the reconstructed images Figure 11 and Figure 12 appear sharper and clearer in comparison to Figure 9 and Figure 10 respectively. The filtered reconstructed image of step size 5 is more faded since it has less projection data in comparison to the reconstruction image of step size of 1.

4.5 Reconstructing Image of Using Sinogram from X-rays

From Section 3.5, we can see that the reconstructed image is a pair of lungs. We begin with comparing the filtered and unfiltered reconstructed images for all cases, with and without noise. The unfiltered version of the image as seen in Figure 13, Figure 15, Figure 17, Figure 19 and Figure 21 all contain blurriness. After adding the ramp filter, we expect the blurriness from the low frequency to be removed and create a clearer and higher resolution image. The results agree with what we expected as seen in Figure 14, Figure 16, Figure 18, Figure 20 and Figure 22 and can be compared to their corresponding unfiltered version.

4.5.1 Reconstruction Using Sinogram from X-Rays versus Sinogram from Optical Rays

As seen in Figure 14 the reconstructed image of the lungs for sinogram data obtain from using x-rays appears less washed out than in the reconstructed image from the sinogram of the optical rays acquired in the lab of Figure 11.

This make sense since optical rays either hits the sample with total absorption or misses and has total transmission. The reconstruction is essentially similar to black or white data since it is either a hit or miss. The pixels in relation to this then is either very high for total absorption or very low for total transmission. When we apply the filter, we remove all the dark intensity areas where pixels remain constant occupied in the low frequency area, and we are left with only very high frequency where pixels are changing drastically. This is why we see a much lighter and washed out reconstructed image in comparison to the result from the sinogram from x-rays.

Sinogram obtained with real x-rays have a more dynamic range of pixel intensities depending on the object the x-ray is going through. A wider range of pixel intensities means we have more ranges in frequencies. When we apply the filter, the blurriness from the low frequencies are filtered out but we still have a range of different changes in pixel intensities. This is why our reconstructed image in the x-ray case has more ranges of intensities and does not look washout compared to the optics case.

4.5.2 The Addition of Noise

We expected that adding a small amount of Gaussian noise to the image by setting the mean to zero and the variance at 0.0001 would make our reconstructed image slightly grainier. This holds true as seen in Figure 16 in comparison with Figure 14 when there is no noise.

From Figure 18, Figure 20 and Figure 22, we can see that as keep our Gaussian noise at the set mean and variance, and lower the dose by factors 2, 5 and 10, the resolution of the reconstructed image decreases. This make sense because by dividing our image down by some factor and keeping the noise power constant at the same mean and variance, we are reducing the amount of signal to noise ratio. This is why the SNR reduces as we increase the dose number. This holds true with the results since we can see that the reconstructed image becomes more and more grainy as we divide the image by a higher constant as observed in Figure 18, Figure 20 and Figure 21.

4.6 Medical Image Visualization

The results indicate the successful medical image visualization using an open source image visualization software FIJI. The results obtained in Figure 24 show a volumetric CT x-ray chest dataset which is a 3-D volumetric representation from 2-D slices. We also obtained various slices for the CT chest x-ray which we determined from utilizing Table 1.

When we comparing our results with the brain example shown in Figure 3, Figure 4 and Figure 5 we can see that Figure 25, Figure 26, and Figure 27 are respectively the axial slice which shows the top view, the coronal slice which shows the frontal view slice, and the sagittal slice which shows the lateral view.

5 Conclusion

In this simulation the source and detectors are stationary while the object is being rotated while in a real CT imaging the source and detectors rotate around a stationary object instead. They both result in a tomographic image, but the use of optical rays to attenuate a rotating object exposes us to the concepts and principles of CT without being exposed to x-ray beams of a real CT. We can conclude that there were differences between the optical ray sinogram data and the actual x-ray sinogram data. This difference was clearly seen when comparing their filtered reconstructed images.

From the sinogram, backprojection, and filtered backprojection sections we can conclude that having lower degrees of rotation does give a better resolution. For our data, we want a higher number of rotation positions, which is done by lowering the number of step size. When using a higher step size it resulted in a lower number of rotation positions. We can see that this causes our reconstructed images to have lower resolutions as seen clearly when comparing the images with step size of 1 with the images with step size of 5.

Adding a filter to the backprojection creates clearer image. Comparing the filtered and unfiltered backprojections on either step sizes, the unfiltered images are blurrier. The Ram-Lak filter we added allowed more high frequencies to pass through while at the same time blocking the blurry low frequencies.

For the sinogram from x-rays we were given, after reconstruction without noise we can conclude that it is a CT image of a pair of lungs. The difference is because the pixels in relation to the optical rays are either very high for total absorption or very low for total transmission so the addition of a Ram-Lak filter removes the dark intensity areas from the low frequencies and leaves only very high frequencies where pixels are changing drastically. On the other hand, the x-ray sinogram has a more dynamic range of pixel intensity so it has more ranges of frequencies. So even after the Ram-Lak filter removes the blurriness from the low frequencies, there is still a wide range of intensities which does not cause the reconstructed image to look washed out. In essence, the reconstructed image from the optical ray sinogram is much lighter and more washed out compared to the reconstructed image from the x-rays sinogram.

When adding noise to the x-ray sinogram and then lowering the dose by factors of 2, 5, and 10 we could clearly see the resolution of the reconstructed images decreasing with each increasing factor. Each factor of dose relates to how much our data was being divided while the noise power remained constant. So every increasing dose reduced the amount of SNR which is seen from the increasingly grainier image with each increasing dose.

We also obtained a CT data set which we then showed its axial slice, coronal slice, and sagittal slice using a visualization software called FIJI. This medical image visualization software was surprisingly powerful and we were able to easily acquire the different slices using the software.

6 Reference

[1] K. W. Eliceiri. (2005, September). *Tools for Visualizing Multidimensional Images from Living Specimens* [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1562/2004-11-22-IR-377/abstract;jsessionid=D26E4CAB007C76B8F9F1B98258DDE0D3.f01t04

[2] W. Rasband. (2012). *ImageJ* [Online]. Available: http://imagej.net/ImageJ

[3] VIA/I-ELCAP Public Access Research Database. "Chest Analysis Results," [Online]. Available: https://veet.via.cornell.edu/cgi-bin/datac/liview.cgi.

[4] J.E. Zapawa. "Radiologic Anatomy," [Online]. Available: https://www2.med.wayne.edu/diagRadiology/Anatomy_Modules/brain/brain.html.

[5] J.L. Prince, "Computed Tomography," in *Medical Imaging Signals and Systems*, 2nd ed. NJ, Prentice Hall, 2015, ch.6, pp. 186-234.

7 Appendix

```
%Lab 2 Part 1 (optical ray sinogram; projection & filtered projection)
clc;
clear all;
data=importdata('Sinogram180deg1step.txt');
%data1=importdata('Sinogram180deg5step.txt');
%Size of the width/length of image
size=1362;
%Checks results to iradon function
% I1 = iradon(data,1:181, 'linear', 1, size);
      % figure(1)
      % imagesc(I1)
      % colormap('bone')
      filter_on= '1';
      if filter_on == '1'
         for i= 1:181
            % Use fftshift to move zero frequency component
            % Essential creates an inverted ramp
            fft_data=fftshift(fft(data(:,i)));
            % Considers case for when pixel size even and odd
            if(mod(size,2)==0) % even
                %Ramp filter
                x=linspace(0,1,size/2);
                ramp= [linspace(1, 0 , size/2) x(2:end) 1];
            else
                %odd
                x=linspace(0,1,round(size/2));
                ramp= [linspace(1, 0 , round(size/2)) x(2:end)];
            end
            % Add ramp to projection
            filtered = fft_data.*ramp';
            filtered= ifftshift(filtered);
            data(:,i)=real(ifft(filtered));
         end
      end
      backproj_img=0;
      for i= 1:181
         % Copies a projection vector at angle i 301 times to create a
square image
         img = repmat(data(:,i), 1, size);
         % Rotates  at acquired angle i
         rotateimg = imrotate(img, 90+i, 'bilinear', 'crop');
         % sums up all the square image at each angle i together
         backproj_img = backproj_img + rotateimg;
      end
      figure(2);
      imagesc(backproj_img)
      colormap('bone')
      title('Filtered Reconstruction Image')
      ylabel('Y Pixels') % x-axis label
      xlabel('X Pixels') % y-axis label
```

```
%Lab 2 Part 2 (x-ray sinogram, reconstructing image with and without filtering and with and without noise and dosage)
clc;
clear all;
% Dose constant
dose=1;
orig_img=(imread('sino1.bmp')')/dose;
%Add Gaussian noise to system mean=0 var=0.0001
data=double(imnoise(orig_img, 'gaussian', 0, 0.0001));
%Image size = 301 pixels
      size=301;

      filter_on= '1';
      if filter_on == '1'
          for i= 1:180
             % Use fftshift to move zero frequency component
             % Essential creates an inverted ramp
             fft_data=fftshift(fft(data(:,i)));
             % Considers case for when pixel size even and odd
             if(mod(size,2)==0) % even
                 %Ramp filter
                 x=linspace(0,1,size/2);
                 ramp= [linspace(1, 0 , size/2) x(2:end) 1];
             else
                 %odd
                 x=linspace(0,1,round(size/2));
                 ramp= [linspace(1, 0 , round(size/2)) x(2:end)];
             end
             % Add ramp to projection
             filtered = fft_data.*ramp';
             filtered= ifftshift(filtered);
             data(:,i)=real(ifft(filtered));
          end
      end
      backproj_img=0;
      for i= 1:180
         % Copies a projection vector at angle i 301 times to create a square image
         img = repmat(data(:,i), 1, size);
         % Rotates  at acquired angle i
         rotateimg = imrotate(img, 90+i, 'bilinear', 'crop');
         % sums up all the square image at each angle i together
         backproj_img = backproj_img + rotateimg;
      end
      imagesc(backproj_img)
      colormap('bone')
      title('Filtered Reconstructed Image with Gaussian Noise Dose=1')
      ylabel('Y Pixels') % x-axis label
      xlabel('X Pixels') % y-axis label
```