

# MEDE3501 Medical Imaging

## Class Project

Group 3 Method III: Fourier Filtered Backprojection



## Project Report

~ March 2018 ~

Name

UID

Chui Mei Yee

3035380635

Dey Poonam Aditi

3035296945

Fung Chun Hin

3035277406

Lv Ruyi

3035142560

# 1. Introduction

Tomography is a technique to investigate the structure and composition of an object non-invasively, along spatial and temporal dimensions using photonic radiation, acoustic or electromagnetic waves, such as X-rays and gamma rays. Initially, the tomographic system acquires a set of partial measurements by scanning. These measurements are then used for tomographic projection and reconstruction of optimal tomographic images.

Initially, discrete image reconstruction algorithms were used for the development of computed tomographic (CT) images but they usually resulted from the iterative solution of a large system of equations. Analytical reconstruction algorithms, on the other hand, are where the measurement procedure is assumed to be continuous given by the Radon transform and the system equation is analytically solved. One of the commonly used analytical reconstruction algorithms, especially in CT scanning, due to its fast image reconstruction and ease of implementation is Fourier Filtered Back-projection. However, the lower computational processing time compromises with the signal-to-noise ratio and other artifacts which hampers the resolution of the images. Using low dose CT, which is gaining more popularity at present, further increases the noise pollution in the raw data leading to more quality deterioration.

The purpose of this report is to use Fourier Filtered Backprojection on a Shepp-Logan phantom image generated in MATLAB. After construction of the image, it is transformed into the radon and frequency domains. Next, nine filters are applied on the image, followed by a one-dimensional fast inverse fourier transform and backprojection. After reconstruction, the effects of sampling, missing projection, filtering, noise and other artifacts on the image are analyzed and discussed briefly.

## 2. Basic Reconstruction Principle

Our reconstruction method is Fourier Filtered Backprojection. The method can be divided into five steps. Firstly, we obtain the Radon transform  $g(s, \theta)$  from the original image.

Mathematically, this corresponds to the equation  $g(s, \theta) = \int_L \mu(x, y) dl$  for a certain angle  $\theta$ . In MATLAB, when the angles are defined, this is simply `[R, xp] = radon(P, theta)`, where R is the radon transform desired. We also wrote our own Radon Transform function, which can be found in the appendix. The shape of the obtained R would be [number of detectors, number of angles]. The second step of Fourier Filtered Backprojection is taking the Fourier Transform of the obtained radon transform data. The Fourier transform is given by

$$F(\omega_x, \omega_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp(-i(\omega_x x + \omega_y y)) dx dy$$

mathematically. In MATLAB, we make use of the fast fourier transform function: `proj_fft = fft(R, width)`. The width here is the desired dimension of Fourier Transform and can be tuned by a transform coefficient in our project. Thirdly, a 1D high-pass filter is applied to each column of the data obtained from the last step. All the filtering is in fourier domain. In our project, nine filters (including a ‘none’ for comparison) are available and all of them have a cutoff frequency, which can be tuned by a cut off coefficient. After the filtering, inverse 1D fourier transform is applied to the data. This is

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) \exp(i\omega t) d\omega$$

mathematically and is simply `inverse_f = real(ifft(filtered))` in MATLAB. Again, we make use of the fast inverse fourier transform function and only save the real parts. The final step of Fourier Filtered Backprojection is back-projection. For a certain point, we

integrate the data for all defined angles in step one: 
$$\hat{f}(x, y) \equiv Bg = \int_0^\pi g(s = x \cos \theta + y \sin \theta, \theta) d\theta$$
. In MATLAB, we make use three ‘for loops’ to calculate backprojection for all points. The linear interpolation method is used as default here.

The final Matlab function we developed has the form of ‘`function final_img = FFB(phantom_img, filter_type, dtheta, coe_transform, cut_off)`’. The five arguments are original phantom image, filter type in string (‘lowpasscosine’, ‘shepplogan’, ‘hamming’, ‘hann’, ‘blackman’, ‘bartlett’, ‘barthannwin’), the projection angle interval, self-defined fourier transform coefficient and filter cut off coefficient. The size of the fourier transform is defined to be `width = 2^nextpow2(num_detectors)*(2^coe_transform)` and the actual filter used has a size of `width*cut_off`. With this function defined, parameters can be easily tuned. Extra features in radon domain, such as noise, only require slight changes of the function. For more information about the code please refer to the appendix. A simple proof of Fourier Filtered Backprojection is also provided there.

### 3. Numerical Results

#### 3.1 Quantitative Quality Evaluation

In order to compare and evaluate the quality of the images with the original phantom image when implementing different effects on them, two parameters are tested, namely, the mean squared error (MSE) and the structural similarity index (SSIM).

##### 3.1.1 Mean Squared Error

The MSE is the cumulative squared error between the original image and the image to be evaluated and is given by,

$$MSE = \frac{1}{MN} \sum_{y=1}^M \sum_{x=1}^N [I(x,y) - I'(x,y)]^2$$

##### 3.1.2 Structural Similarity Index

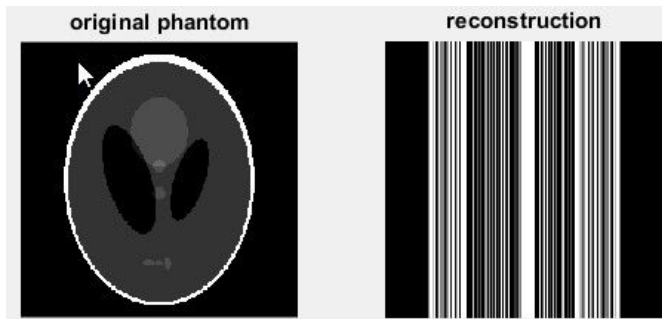
SSIM is a metric that quantifies the perpetual difference between two similar objects, one of which being the original image and the other undergoing possible degradation due to processing. The formula of SSIM is given by,

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

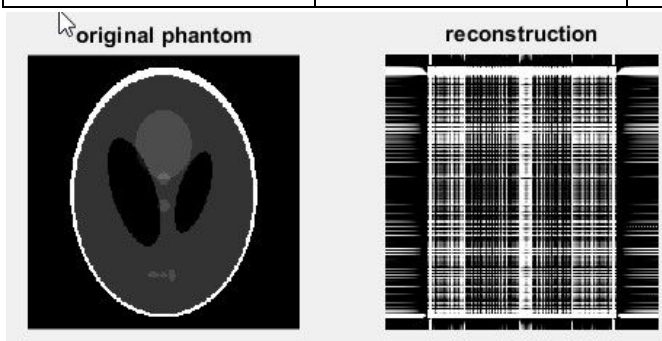
#### 3.2 Effect of sampling

Several sampling interval was tested to see the difference of reconstruction image quality. SSIM and MSE are used to measure the image quality.

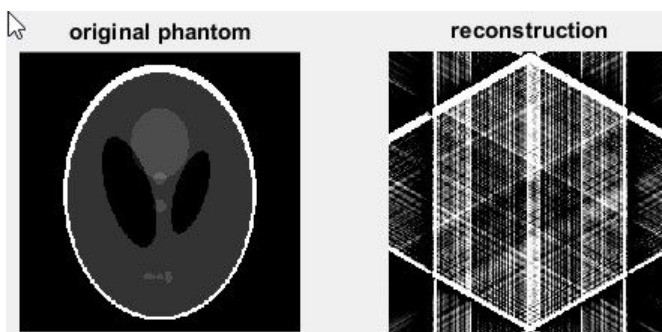
All images are using the same filter (Ram Lak) in order to do the comparison under same condition.



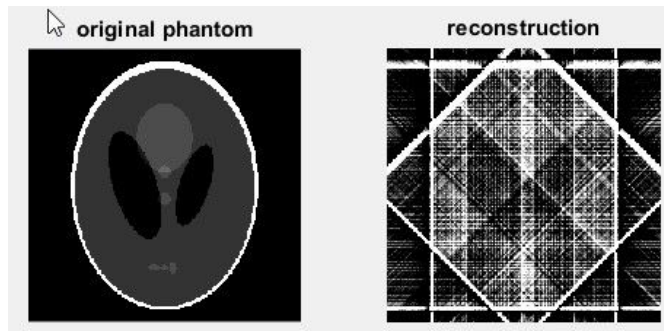
Sampling interval (degree)	No. of projection	SSIM	MSE
180	1	0.0015675	10.8377



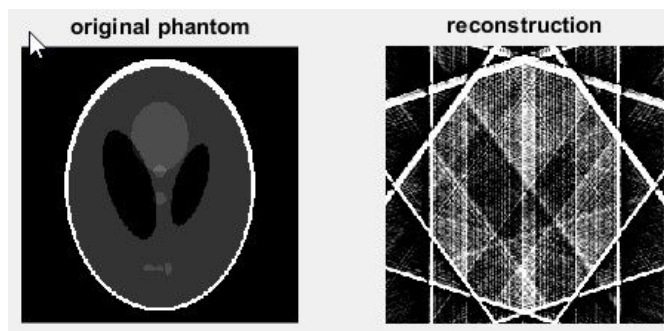
Sampling interval (degree)	No. of projection	SSIM	MSE
90	2	0.0052992	4.0166



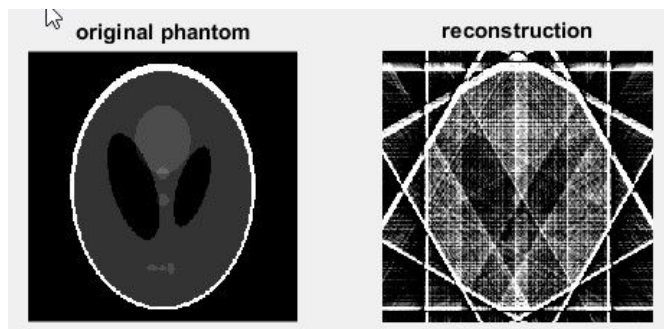
Sampling interval (degree)	No. of projection	SSIM	MSE
60	3	0.010761	1.7944



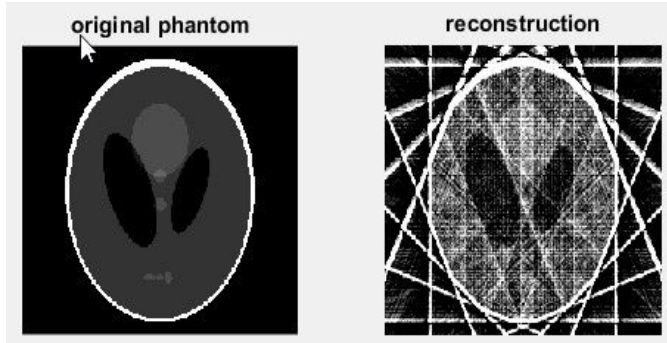
Sampling interval (degree)	No. of projection	SSIM	MSE
45	4	0.0137	1.3971



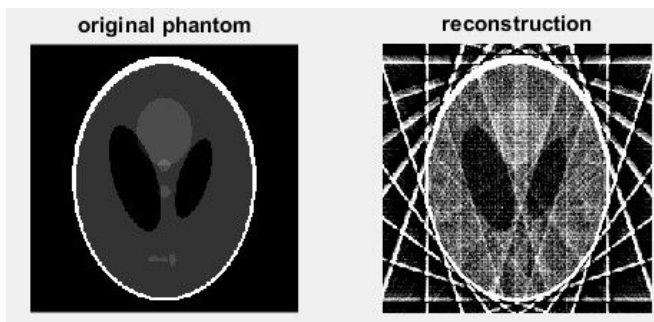
Sampling interval(degree)	No. of projection	SSIM	MSE
36	5	0.022741	0.9373



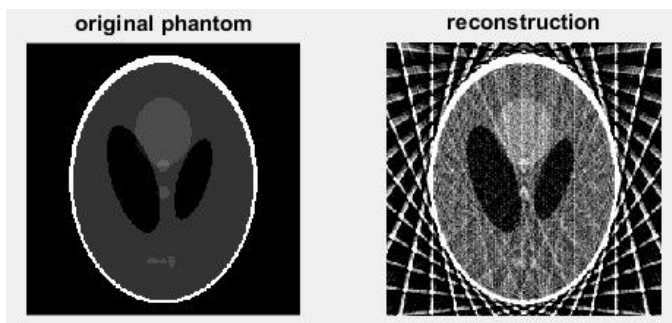
Sampling interval (degree)	No. of projection	SSIM	MSE
30	6	0.030531	0.8233



Sampling interval (degree)	No. of projection	SSIM	MSE
22.5	8	0.044517	0.5855

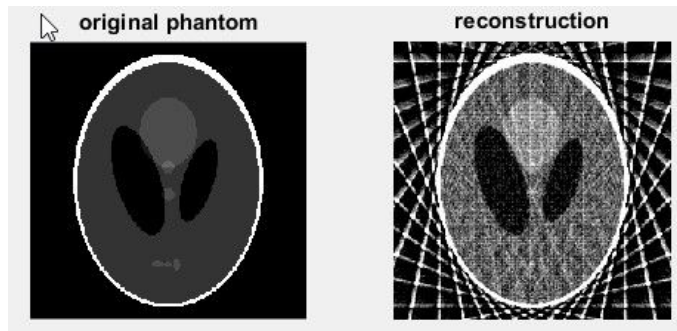


Sampling interval (degree)	No. of projection	SSIM	MSE
18	10	0.057074	0.4476

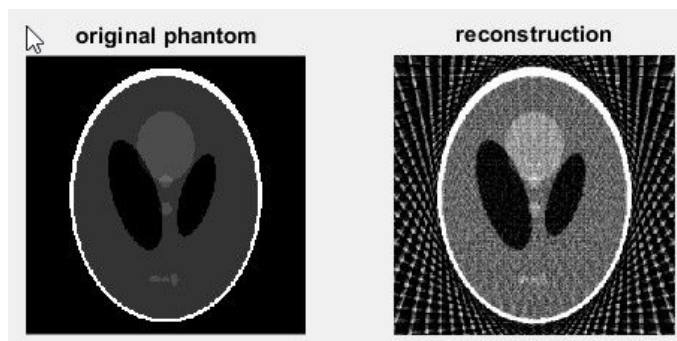


Sampling interval (degree)	No. of projection	SSIM	MSE
12	15	0.082255	0.2702

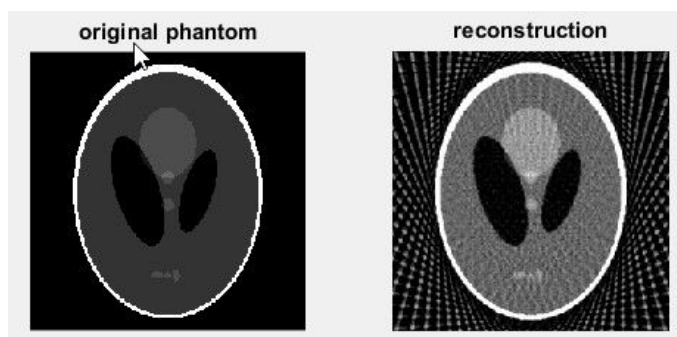




Sampling interval (degree)	No. of projection	SSIM	MSE
11.25	16	0.082603	0.2543

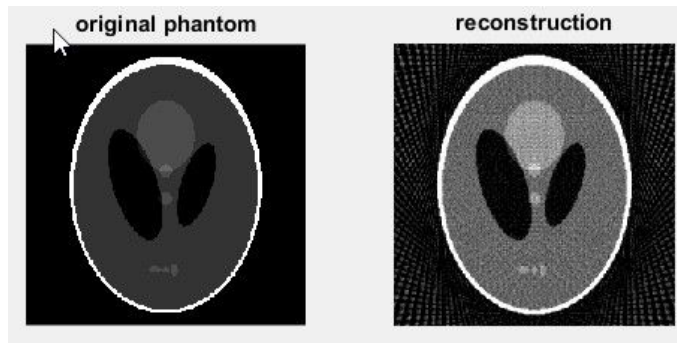


Sampling interval (degree)	No. of projection	SSIM	MSE
5.625	32	0.12704	0.1130

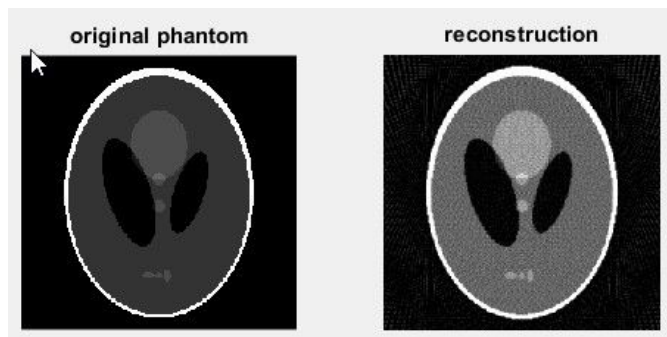


Sampling interval (degree)	No. of projection	SSIM	MSE
5	36	0.1896	0.0839

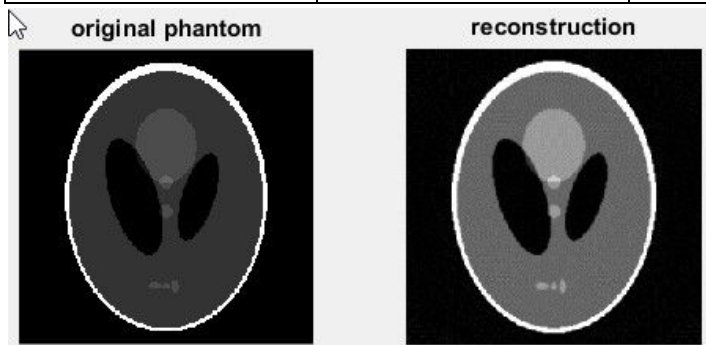




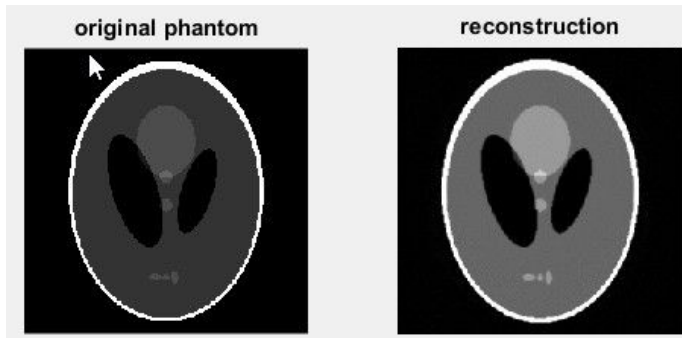
Sampling interval (degree)	No. of projection	SSIM	MSE
3	60	0.19137	0.0669



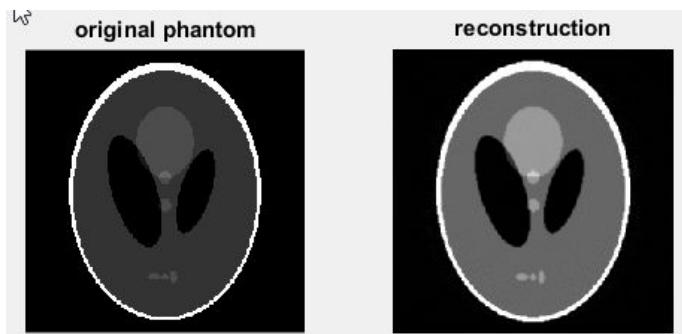
Sampling interval (degree)	No. of projection	SSIM	MSE
2	90	0.29511	0.0567



Sampling interval (degree)	No. of projection	SSIM	MSE
1	180	0.58067	0.0536



Sampling interval (degree)	No. of projection	SSIM	MSE
0.5	360	0.78464	0.0532






Sampling interval (degree)	No. of projection	SSIM	MSE
0.25	720	0.81354	0.0531



Sampling interval (degree)	No. of projection	SSIM	MSE
0.1	1800	0.81526	0.0531

### 3.3 Effect of filtering




For filtering, nine different filters have been used, including no filter. Four of them are traditional backprojection filters, namely, lowpass cosine, Ram-Lak, Shepp-logan and Hamming filters are used. Also, four other FIR high-pass filters are used, including, Hanning, Blackman, Bartlett and Bartlett-Hanning filters. For each of the filters implemented, sampling interval was at 0.5, fourier transform coefficient was at 8. No external noise was added to the images and the number of projections was not altered.

A. Original Phantom	B. No filter	C. Ram-Lak Filter
<p>original phantom</p> 	<p>No Filter</p> 	<p>Ram-Lak</p> 




*Table 1.* A) This figure shows the original image of the Shepp-Logan phantom. B) This figure shows the reconstructed phantom without any filter. C) This figure shows a filtered and reconstructed image. This projection was passed through a high-pass Ram-Lak filter.

#### 3.3.1 Varying filters with a constant cut-off coefficient




For the following images, the cut-off coefficient was kept constant and different filters were used which were analysed and compared in two ways; quantitative analysis using mean error square (MSE) values and Structural Similarity Indexes (SSIM), and qualitative analysis by observation of the images.

	<b>original phantom</b> 	<b>Lowpass Cosine Filter</b> 	<b>Hamming Filter</b> 
MSE	N/A	0.0516	0.0504
SSIM	N/A	0.8442	0.8433

*Table 2.* The first figure is the original phantom, the second figure is with a Lowpass Cosine filter and the third figure is with a Hamming filter. Qualitatively, both of the filtered images look brighter than the original image. When comparing the MSE and SSIM values, there values are similar.

	<b>Shepp-Logan Filter</b> 	<b>Hanning Filter</b> 	<b>Blackman Filter</b> 
MSE	0.0525	0.0503	0.0489
SSIM	0.8141	0.8432	0.8371





*Table 3.* The images are with Shepp-Logan, Hanning and Blackman filters respectively. The images look very similar to each other and the Hamming and cosine filtered images.

	<b>Bartlett Filter</b> 	<b>Bartlett-Hanning Filter</b> 	<b>Ram-Lak</b> 
MSE	0.0454	0.0122	0.0532
SSIM	0.3900	0.85151	0.7846





*Table 4.* The filters used in these images are Bartlett, Bartlett-Hanning, and Ram-Lak filters. The Ram-Lak filter looks similar to the previous filtered images. However, the Bartlett and the Bartlett-Hanning filters look much dimmer than the original phantom image. Contrastingly, the Bartlett-Hanning filtered image has the lowest MSE and the highest SSIM.

### 3.3.2 Variable Cut-off Coefficients with constant filter

For this section, two different cut-off coefficients of the same filter were analysed and compared qualitatively and quantitatively using the same techniques used before

	Hamming Filter		Lowpass Cosine Filter	
	<b>Hamming</b> 	<b>Hamming Filter</b> 	<b>Lowpass Cosine</b> 	<b>Lowpass Cosine Filter</b> 
Cut-off Coefficient	0.8	1.0	0.8	1.0
MSE	0.0026	0.0504	0.0021	0.0516
SSIM	0.9148	0.8433	0.90195	0.8442

*Table 5.* For both the hamming filter and the lowpass cosine filter, the images for the 0.8 cut-off coefficient were dimmer compared to the 1.0 cut-off coefficient filters. The MSEs are also significantly lower and the SSIMs higher for the 0.8 cut-off compared to the 1.0


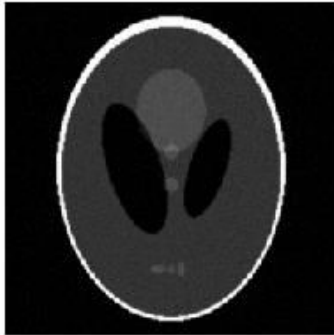
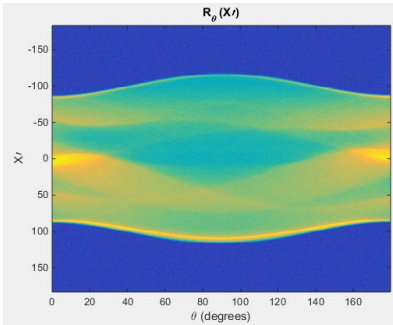


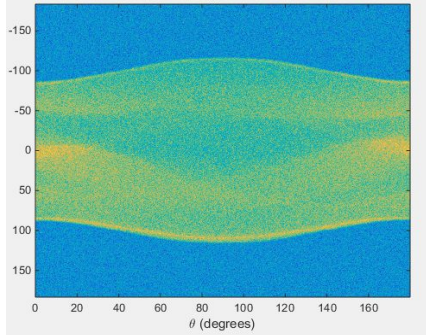

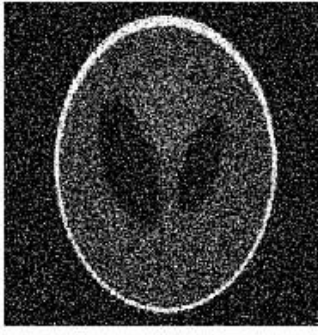
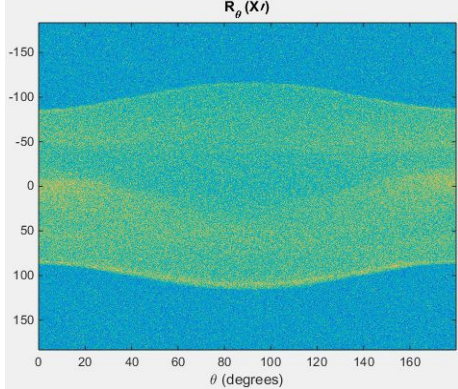
	Hanning Filter		Shepp-Logan Filter	
	reconstruction 	Hanning Filter 	Shepp-Logan 	Shepp-Logan Filter 
Cut-off Coefficient	0.8	1.0	0.8	1.0
MSE	0.0026	0.0503	0.0013	0.0525
SSIM	0.91669	0.8432	0.86308	0.8141

*Table 6.* For the Hanning and Shepp-Logan filters, the trend is similar to what is observed in *Table 5*.

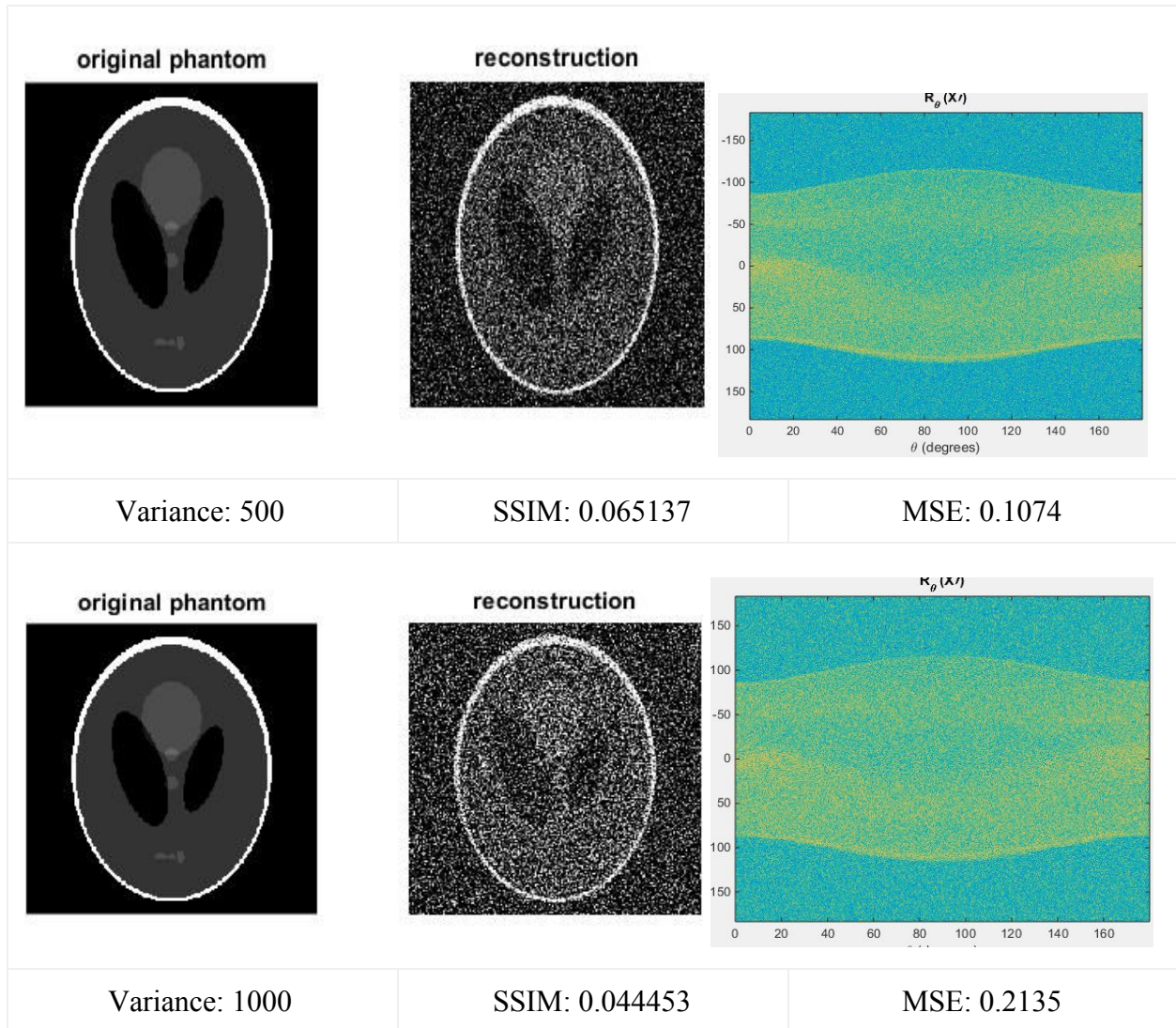
### 3.4.1 Effect of Gaussian White Noise in Radon Domain

Gaussian noise has a normal distribution in the time domain, and white noise is a random signal having equal intensity at different frequencies. Therefore, Gaussian white noise should be the most popular noise in the detection process if there are no big defects in the detector.

Different degree of Gaussian noise was added in the Radon space to analog the noise generated in the detection process. Sampling interval of 0.1 degree and the ramlak filter (coefficient of transform=5, cut off frequency=0.5) were used reconstruct all images under the same conditions. The results are shown below.

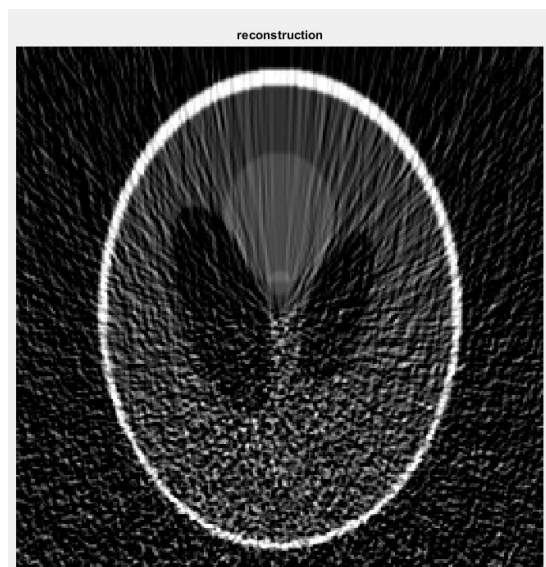
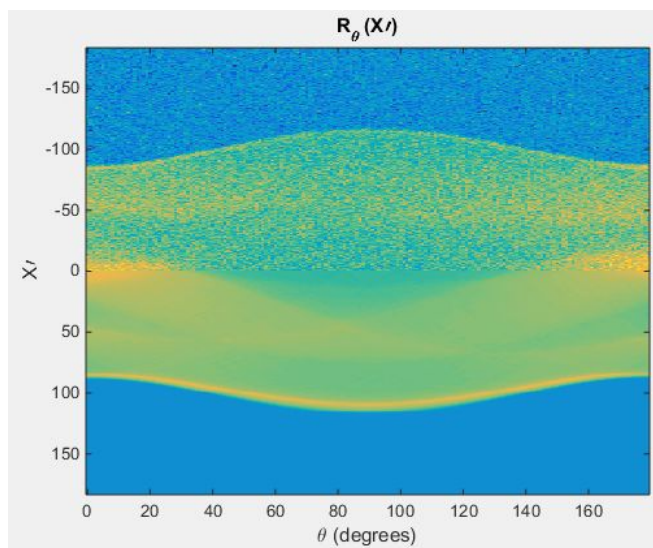
<p>original phantom</p> 	<p>reconstruction</p> 	
<p>Variance: 1</p>	<p>SSIM: 0.80048</p>	<p>MSE: 0.0019</p>
<p>original phantom</p> 	<p>reconstruction</p> 	
<p>Variance: 100</p>	<p>SSIM: 0.12233</p>	<p>MSE: 0.0226</p>
<p>original phantom</p> 	<p>reconstruction</p> 	
<p>Variance: 300</p>	<p>SSIM: 0.078663</p>	<p>MSE: 0.0654</p>



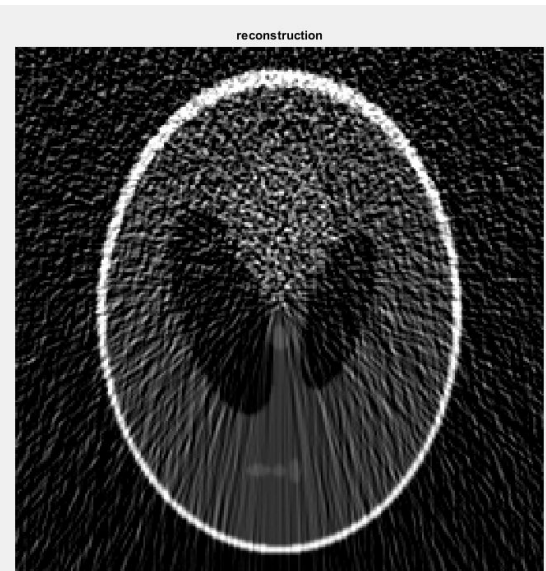
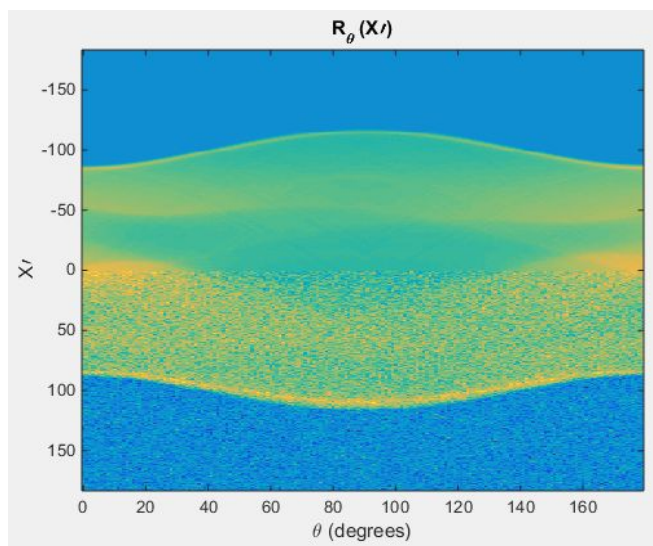


### 3.4.2 Noise Propagation

In this section we investigate the geographical effect of Gaussian white noise in Radon domain. In other words, we are interested in the noise propagation from Radon transform to the final reconstruction image in terms of positions. Four situations are simulated. The same white noise is added to the upper half, lower half, left half, and right half of the radon transform respectively. The noise-added radon transforms and their corresponding reconstructions are recorded as below.

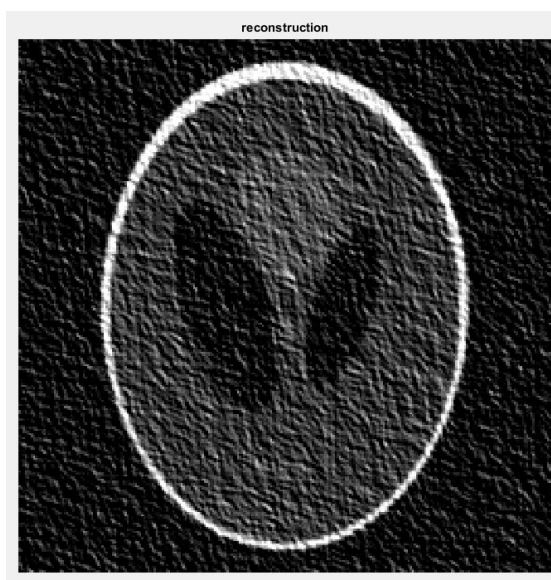
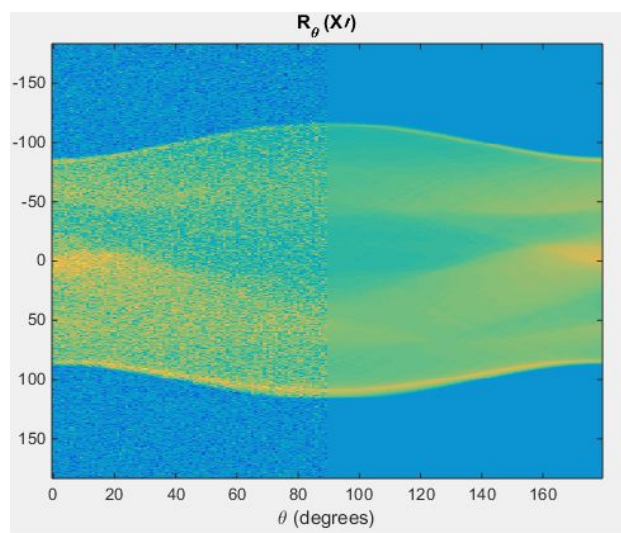


Upper-half noise

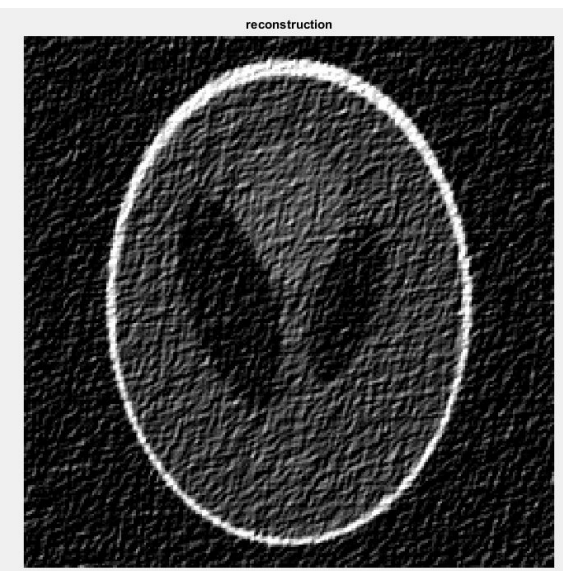
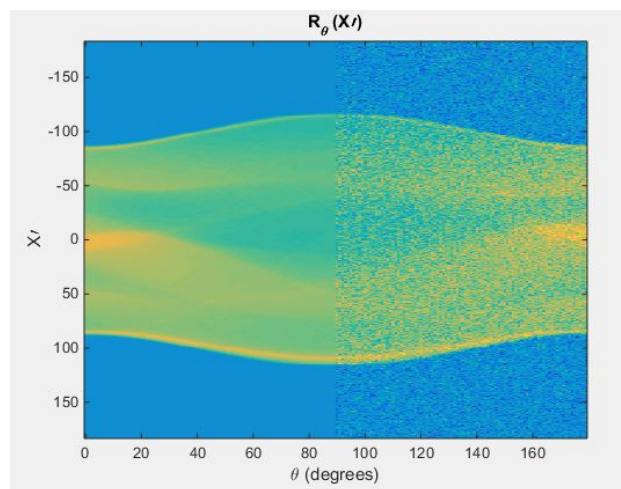


lower-half noise





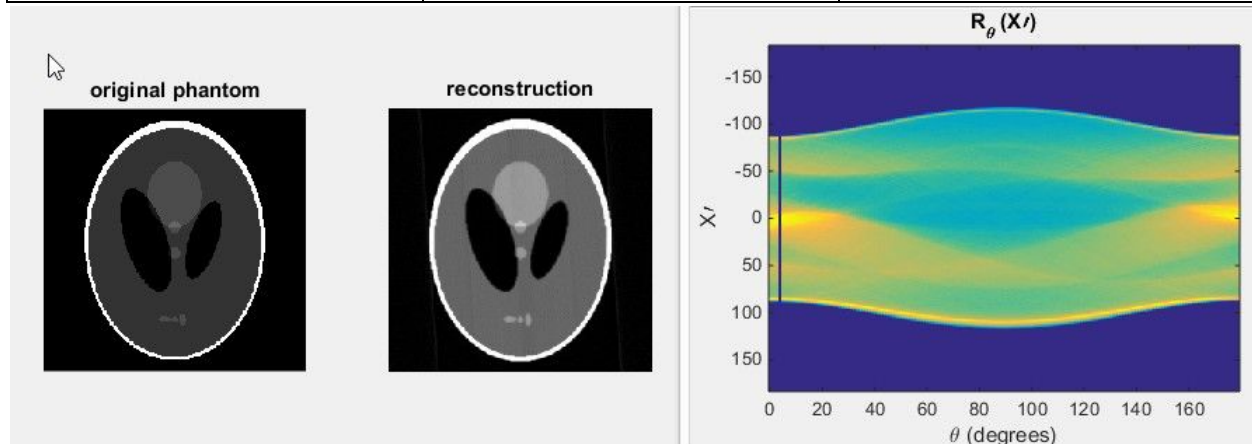
left-half noise



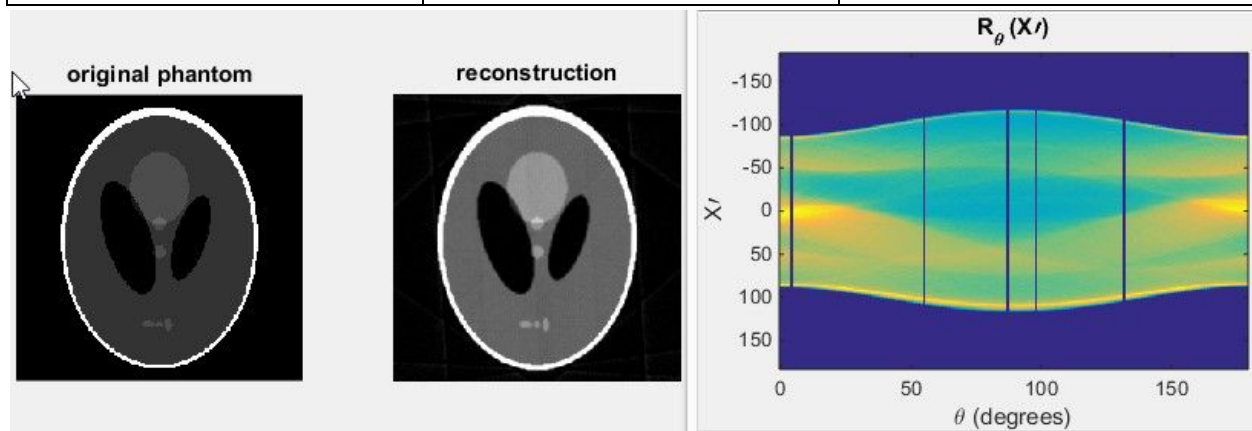
right-half noise

### 3.5.1 Effect of missing projection

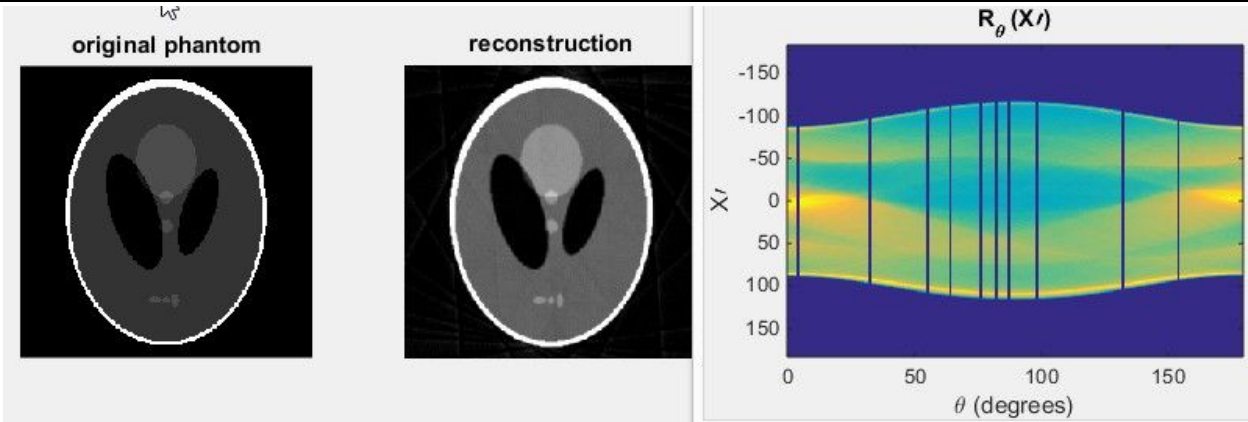
No. of missing projection	SSIM	MSE
1	0.67569	0.0505



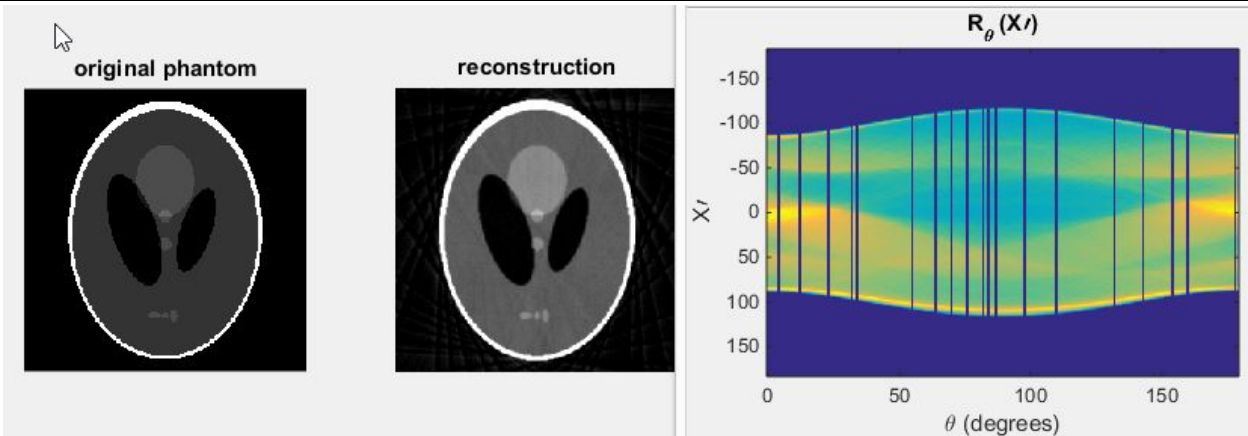
No. of missing projection	SSIM	MSE
5	0.57642	0.0470



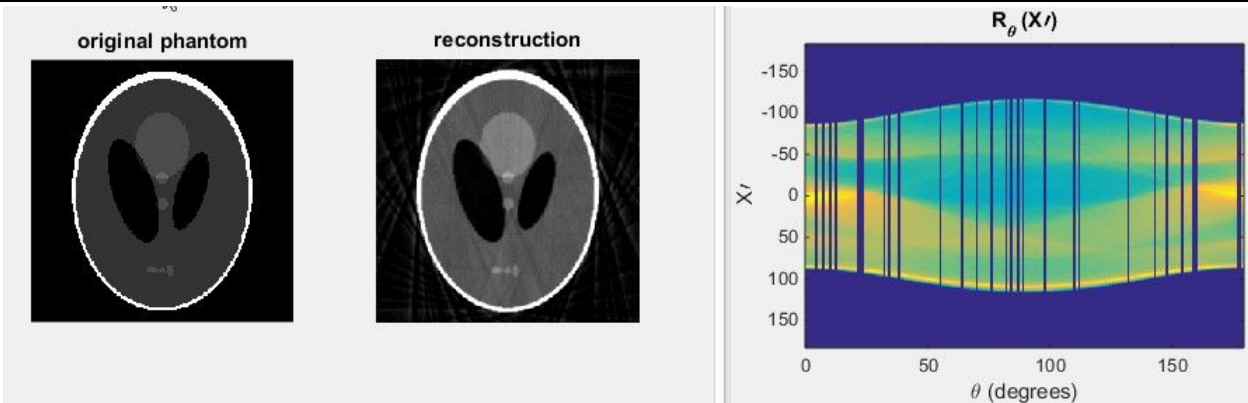
No. of missing projection	SSIM	MSE
10	0.50871	0.0425



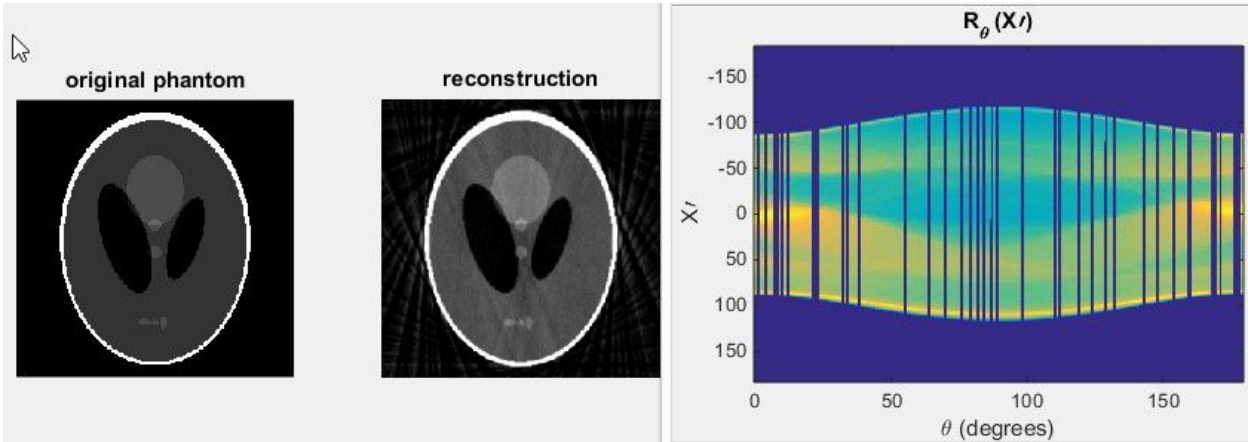
No. of missing projection	SSIM	MSE
20	0.4479	0.0343



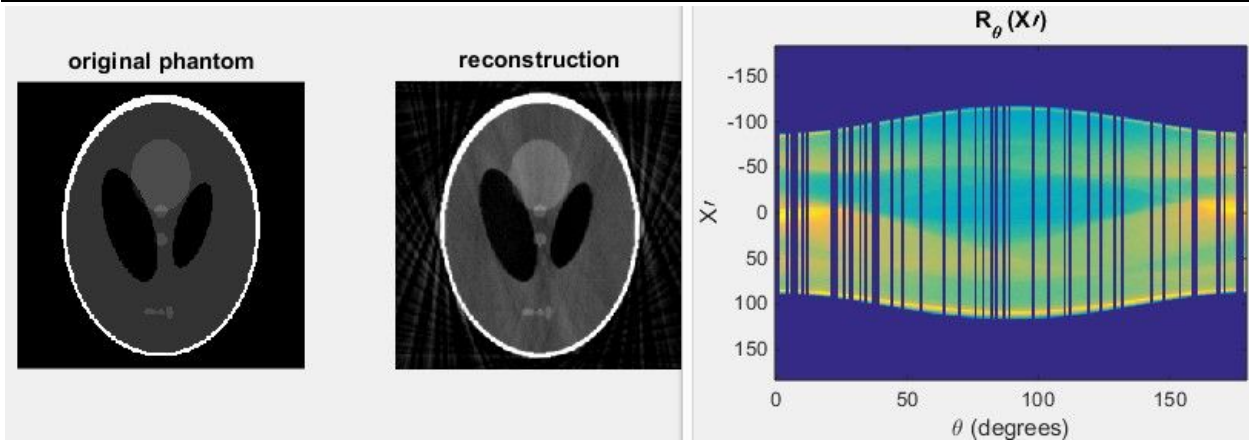
No. of missing projection	SSIM	MSE
30	0.42245	0.0270



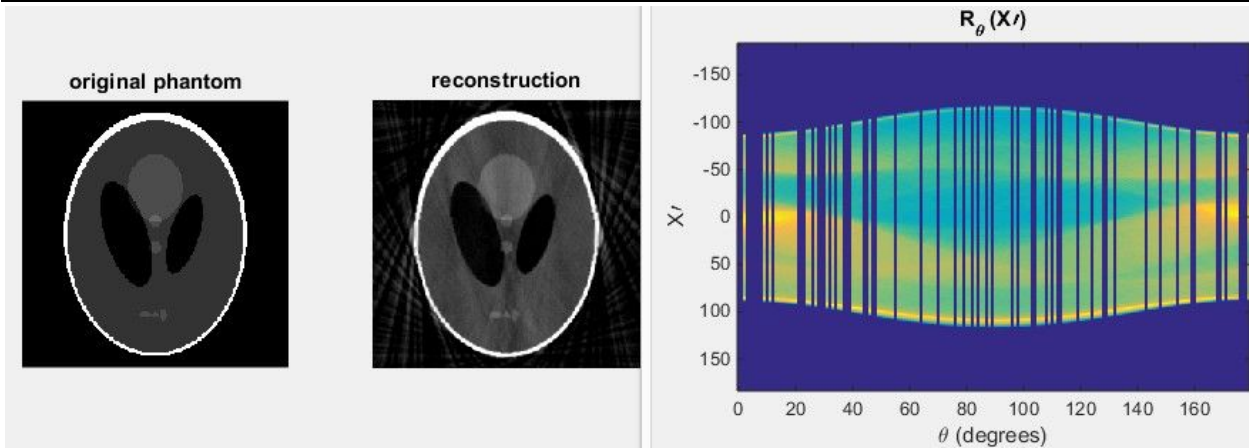
No. of missing projection	SSIM	MSE
40	0.42059	0.0208



No. of missing projection	SSIM	MSE
50	0.41138	0.0162

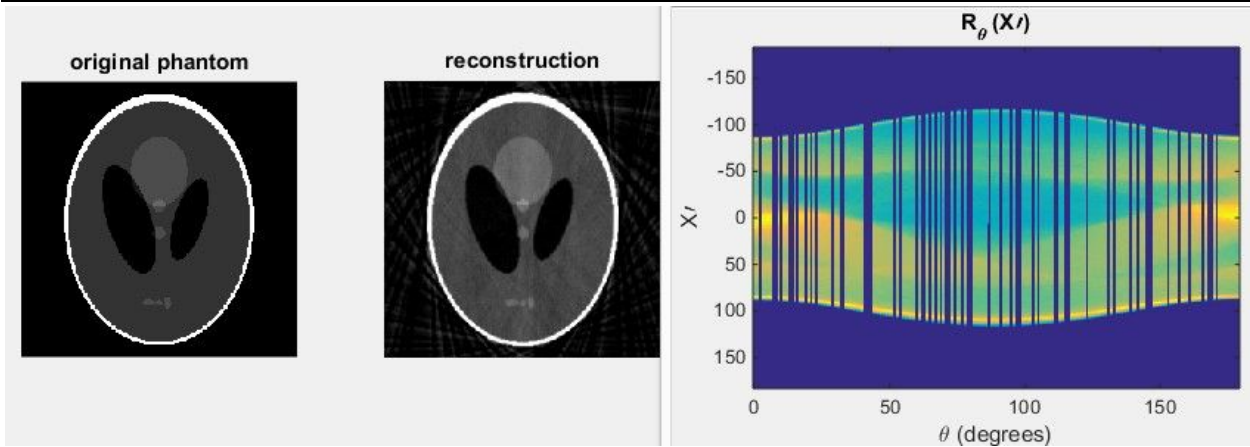


No. of missing projection	SSIM	MSE
60	0.40408	0.0139

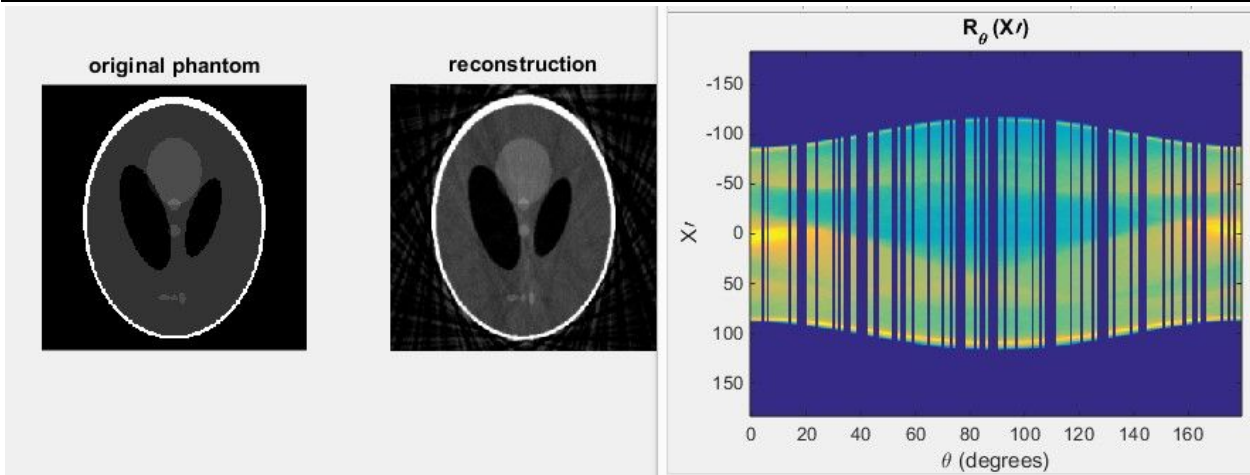




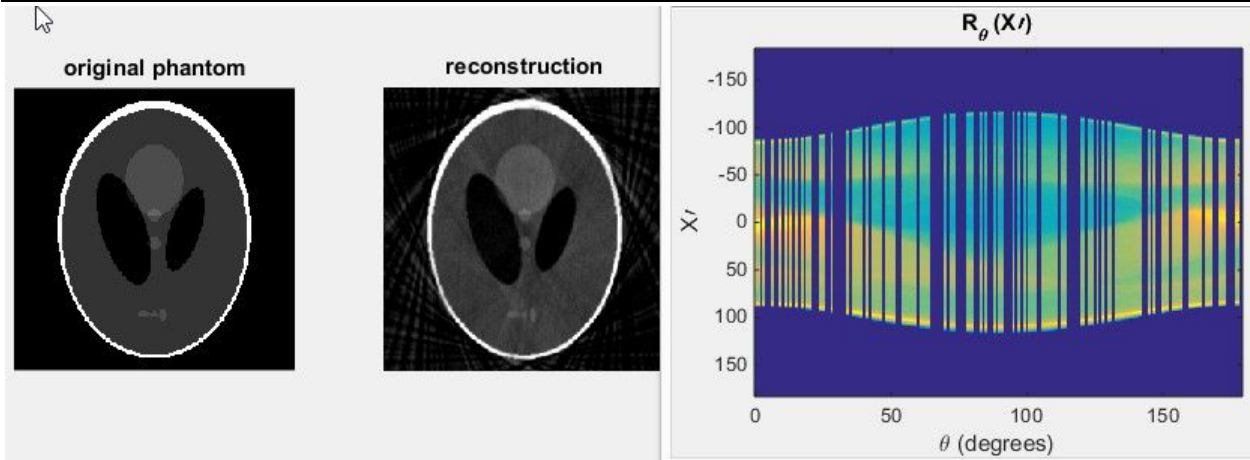
No. of missing projection	SSIM	MSE
70	0.43325	0.0130



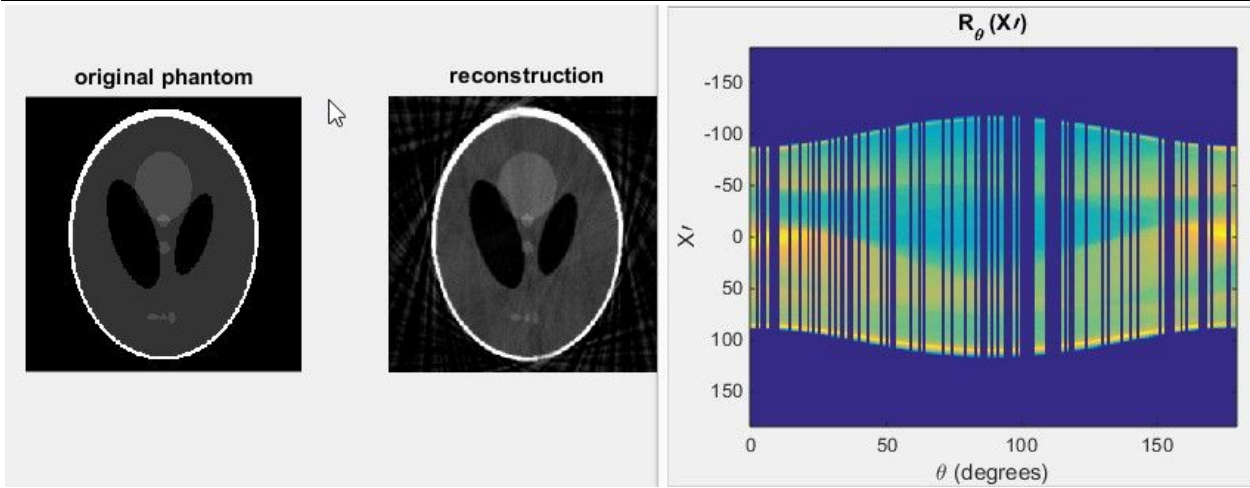
No. of missing projection	SSIM	MSE
80	0.42995	0.0104



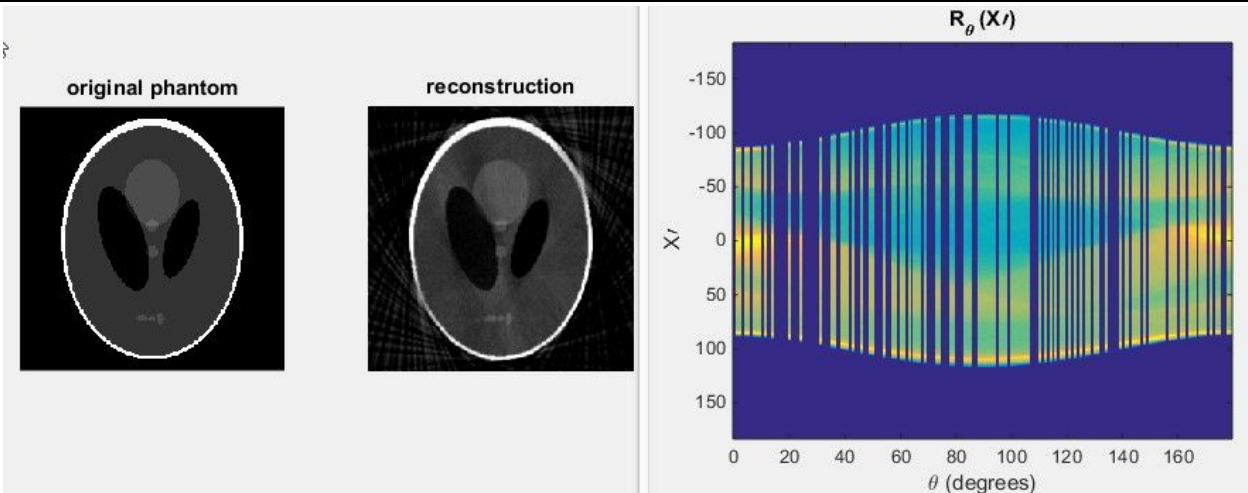
No. of missing projection	SSIM	MSE
90	0.41882	0.0088



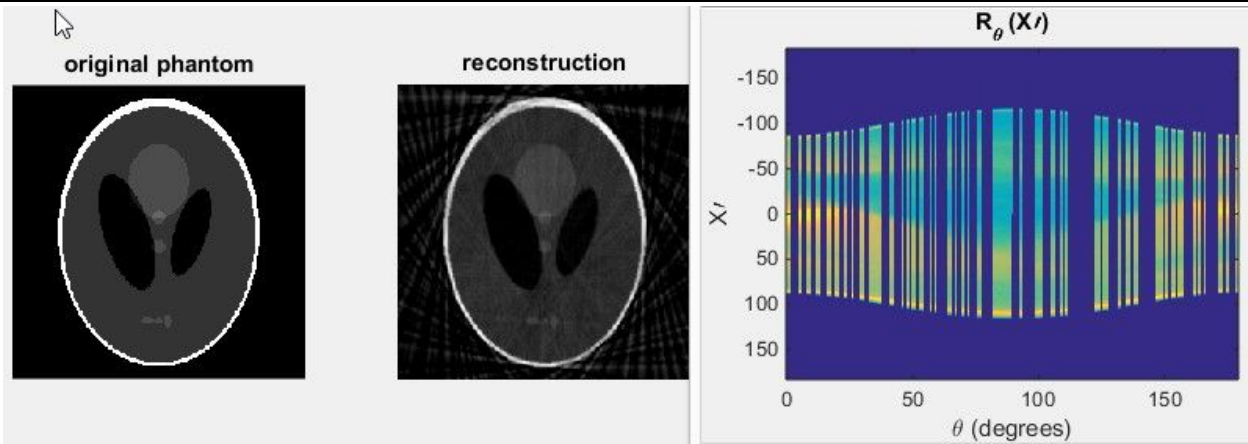
No. of missing projection	SSIM	MSE
100	0.4267	0.0096



No. of missing projection	SSIM	MSE
120	0.40336	0.0091

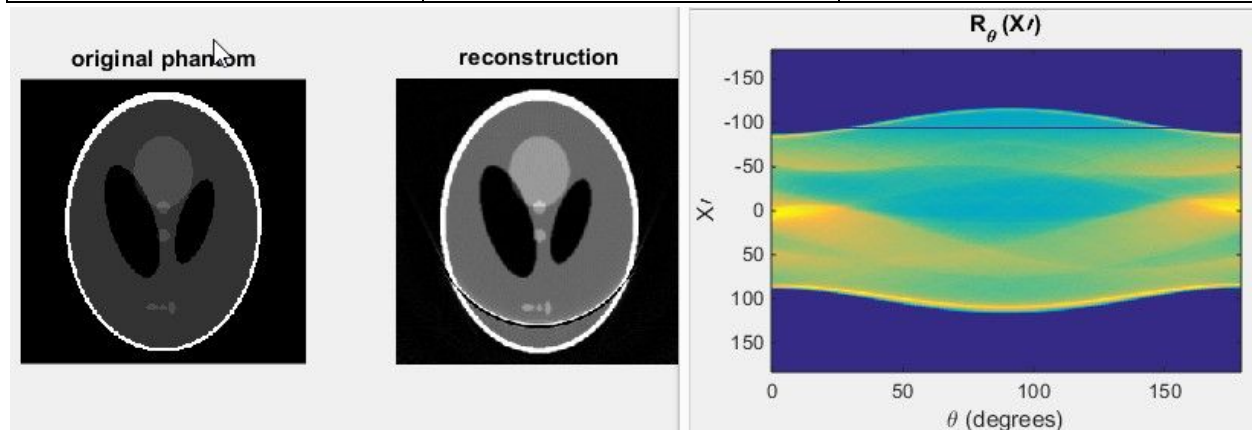


No. of missing projection	SSIM	MSE
150	0.41396	0.0097

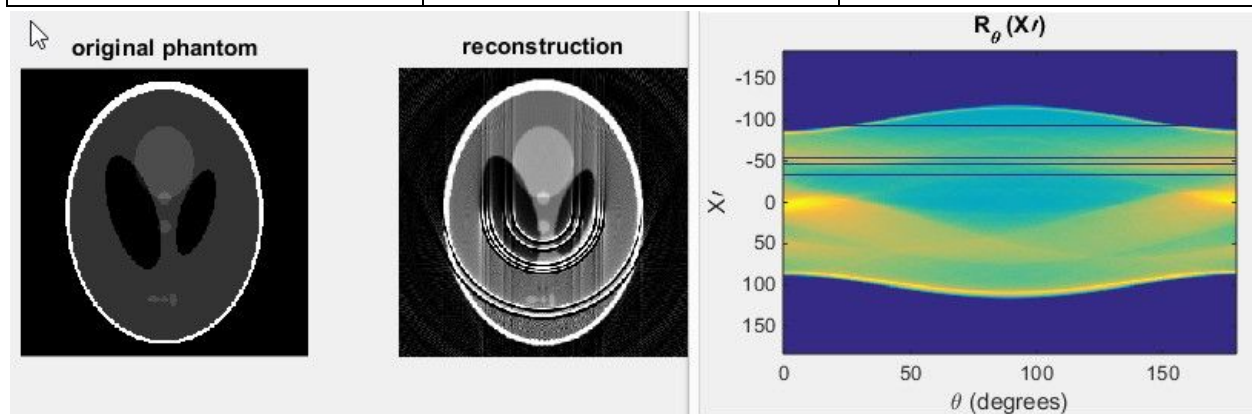


### 3.5.2 Effect of missing detector

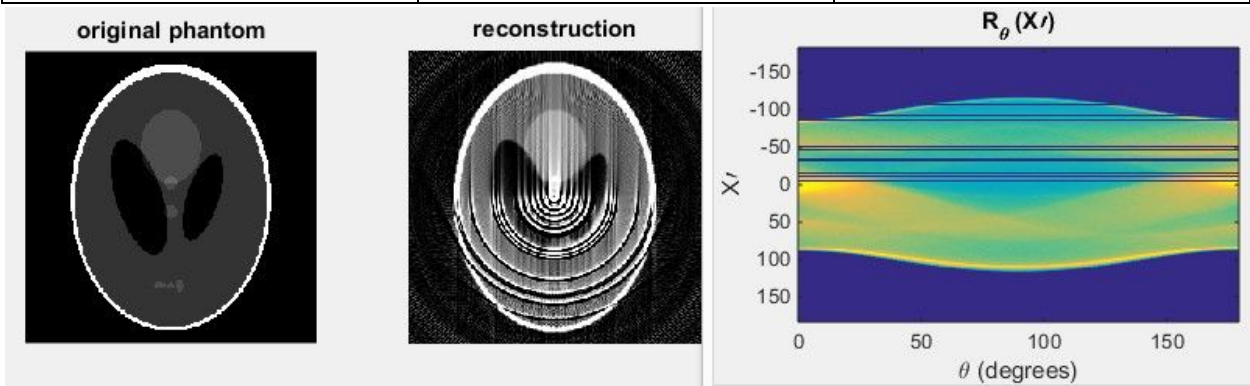
No. of missing detector	SSIM	MSE
1	0.6592	0.0577



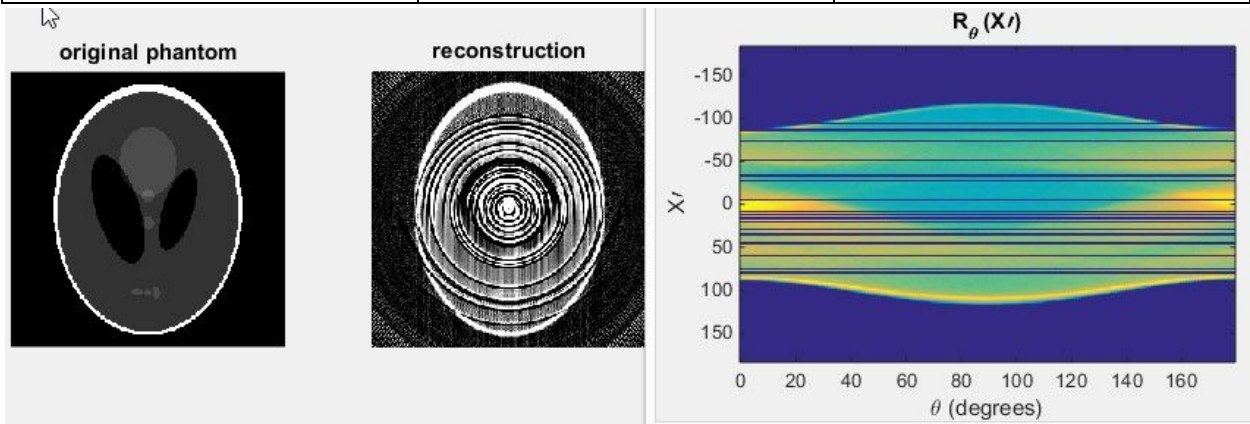
No. of missing detector	SSIM	MSE
10	0.22149	0.1428



No. of missing detector	SSIM	MSE
20	0.1135	0.2176

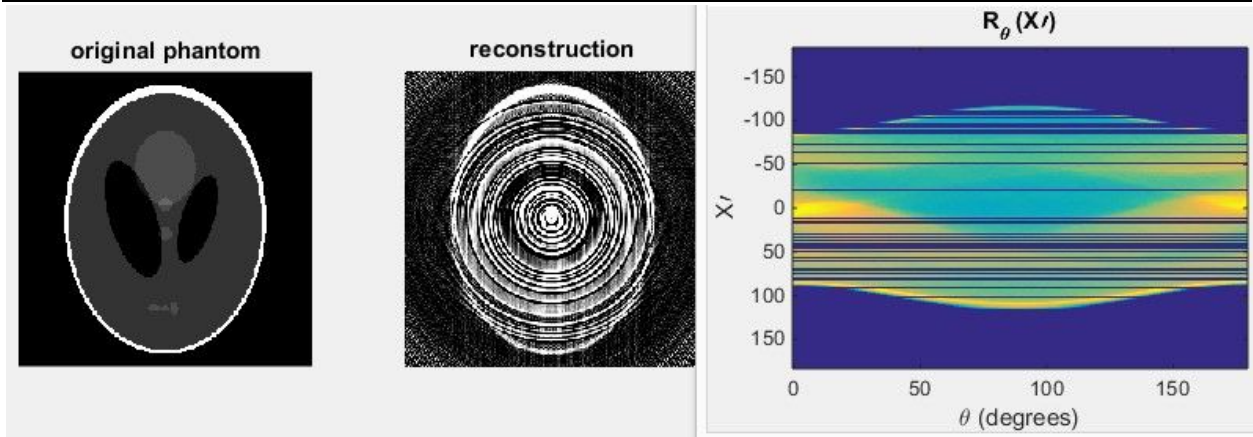


No. of missing detector	SSIM	MSE
50	0.056961	0.5396





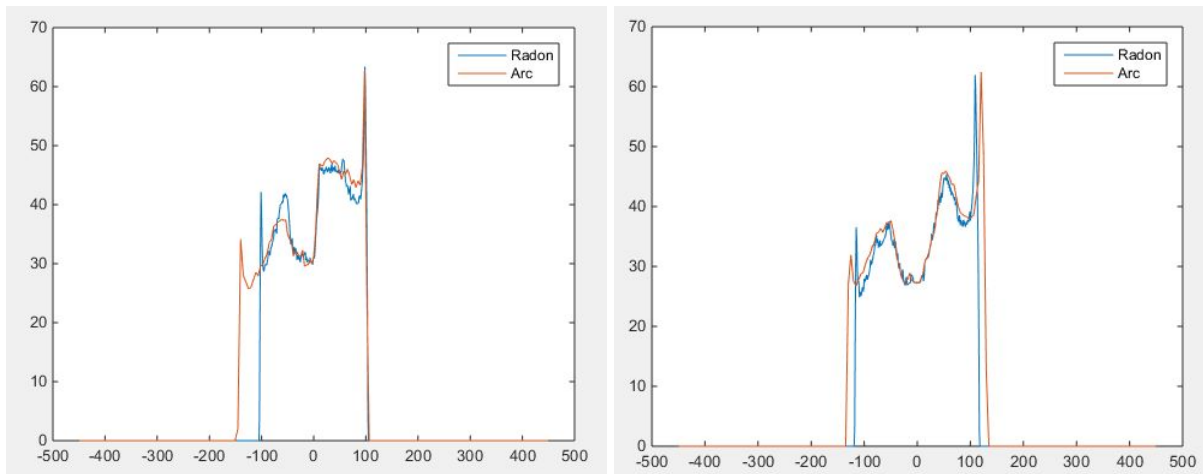
No. of missing detector	SSIM	MSE
100	0.039543	0.5583



## 4. Discussions and Final Reconstructed Object

### 4.1 Pencil-beam and Fan-beam projection

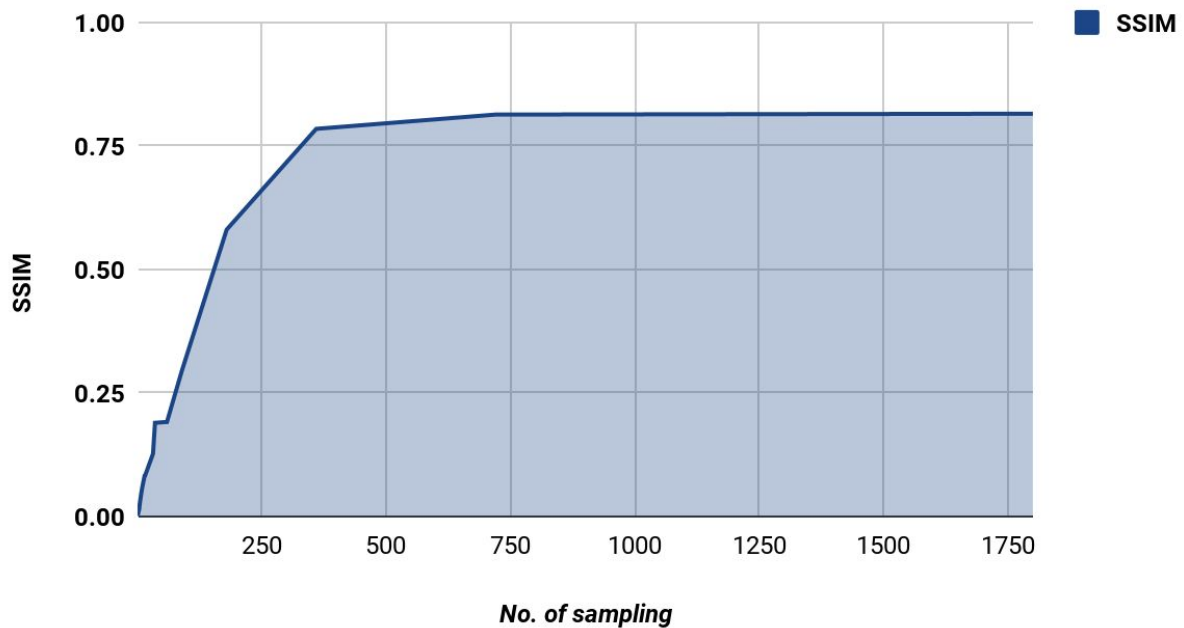
In order to ensure that our project has practical meaning, we compared the projections of the phantom using pencil-beam and fan-beam respectively. The two sets of projection data was obtained by using MATLAB radon and fanbeam functions. Two angles of 45 degrees and 90 degrees were sampled and values in radon domain for different detectors were plotted as below. We can see that the two projections are close to each other in terms of values and shape. The detailed codes are in the appendix.



## 4.2 Effect of sampling

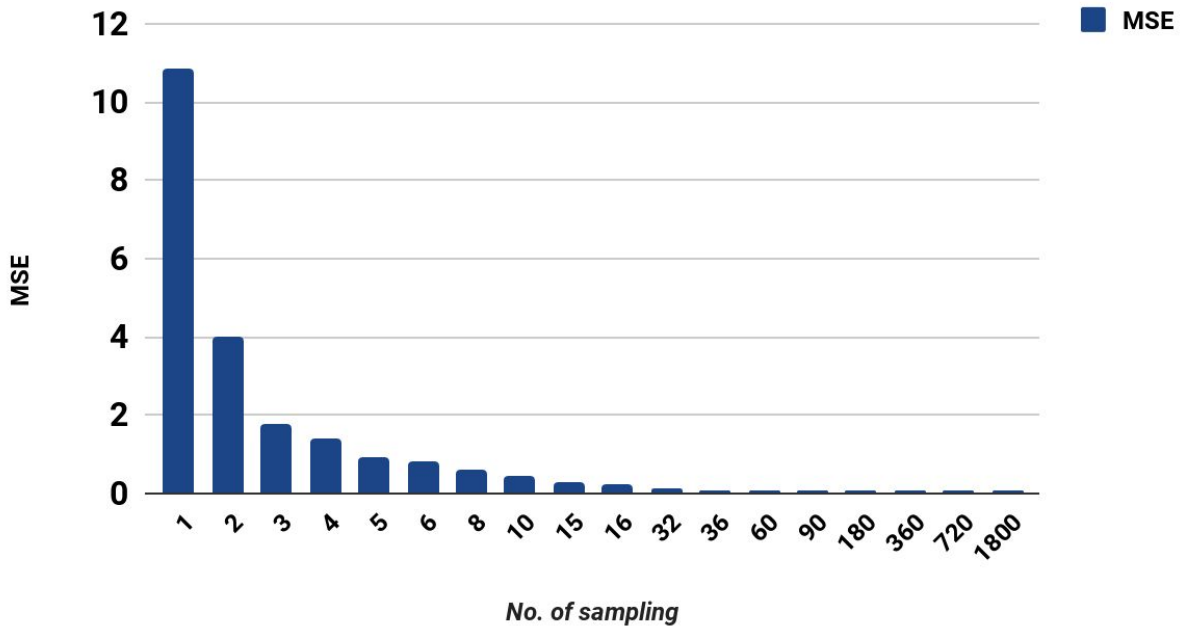
There are total 18 sets of sampling size data, ranging from 1 to 1800, have been taken. From the result we can infer that the quality of the reconstruction improves as more projections are used. When the sampling interval decreases, the SSIM value increases and the MSE value decreases. This is intuitively true as for a single point, more projection means that the sum of backprojection is closer to the complete integral. Therefore, more details can be reconstructed. From the evaluation we can also see that a good MSE is easier to achieve compared with a good SSIM in terms of sampling. This can be verified from two graphs below. For the SSIM, the curve start flattening when no. of sampling reached about 500. But for the MSE, the curve start flattening when the sample size reached around 36 only.

Effect of sampling (SSIM)





### Effect of sampling (MSE)

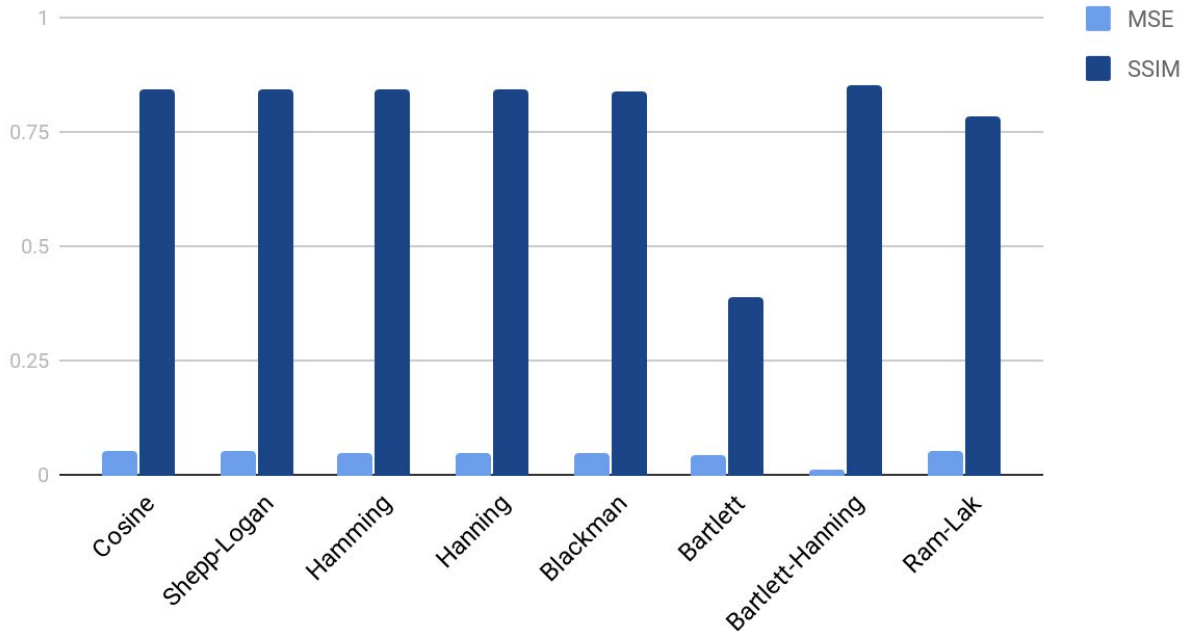


#### 4.3.1 Effects of varying filters on the image quality

In terms of qualitative image quality, it is observed that all filters except for the Bartlett and Bartlett-Hanning filters generated images brighter than the original image and it retained the sharp edges of the images which are seen to have been lost when reconstructing without any filters. The blurred edges proved that low frequency noise was overpowering the reconstructed image. By using high-pass filters, the lower frequencies were eliminated and the image quality is very close to the original phantom image. However, the image with the Bartlett filter looked the dimmest because of which the internal structures are not visible. The image with Bartlett-Hanning filter looked dim as well but all the structures and edges were sharp and visible.

On the other hand, the quantitative values presented different results.

### SSIM and MSE of different filters



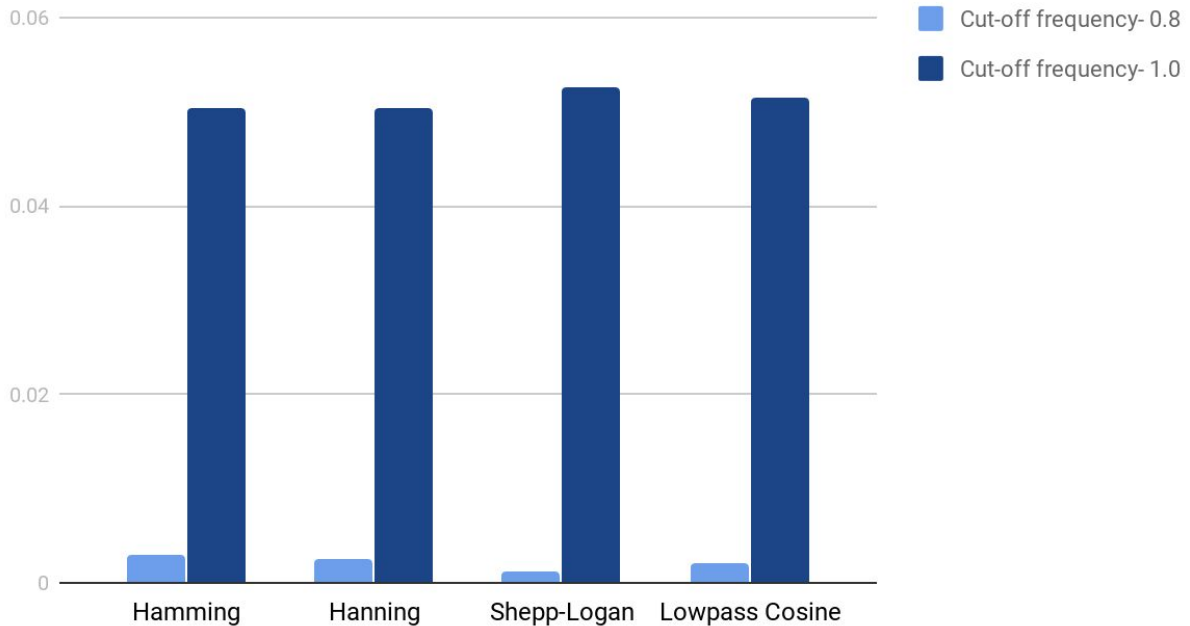
*Figure 1.* The chart shows all the MSE and SSIM values of all the filters. From the graph, it can be deduced that the image with the Bartlett-Hanning filter has the lowest MSE and highest SSIM, showing it has the best image quality. On the other hand, the Bartlett filtered image has the lowest SSIM while the MSE is comparable to the other values, thus, having the worst image quality.

#### 4.3.2 Effects of varying cut-off coefficient on the image quality

When the cut-off coefficient was decreased from 1.0 to 0.8 for the four filters shown before, the images had the same brightness as the original image instead of being brighter as seen in images with cut-off with 1.0. For both types, the edges were sharp and all the structures were clearly visible.

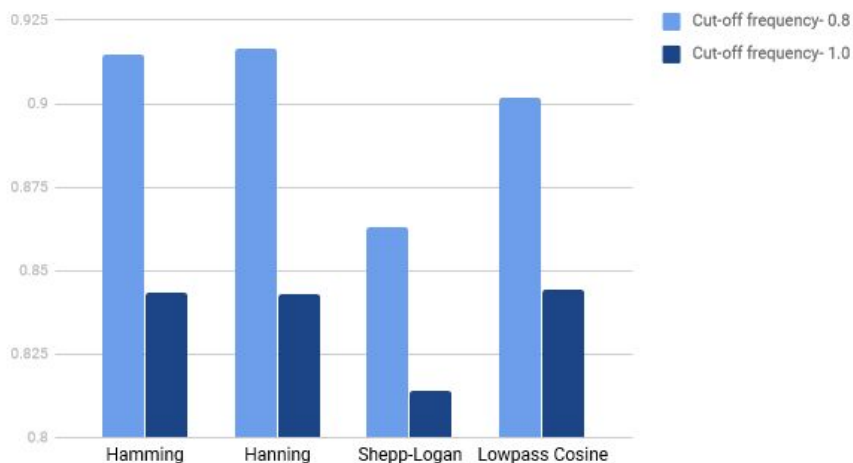
However, there was a big difference in the MSE and SSIM values for different cut-off values which is shown below.

## MSE Comparison between different Cut-off Frequencies



*Figure 2.* compares the MSE values between the two cut-off values between the same type of filters. From here, it is clearly visible that the MSE values are significantly lower at the 0.8 cut-off value than at 1.0, with the Shepp-Logan filter at 0.8 cut-off having the lowest MSE and at 1.0 having the highest MSE.

## SSIM Comparison between different Cut-off Frequencies

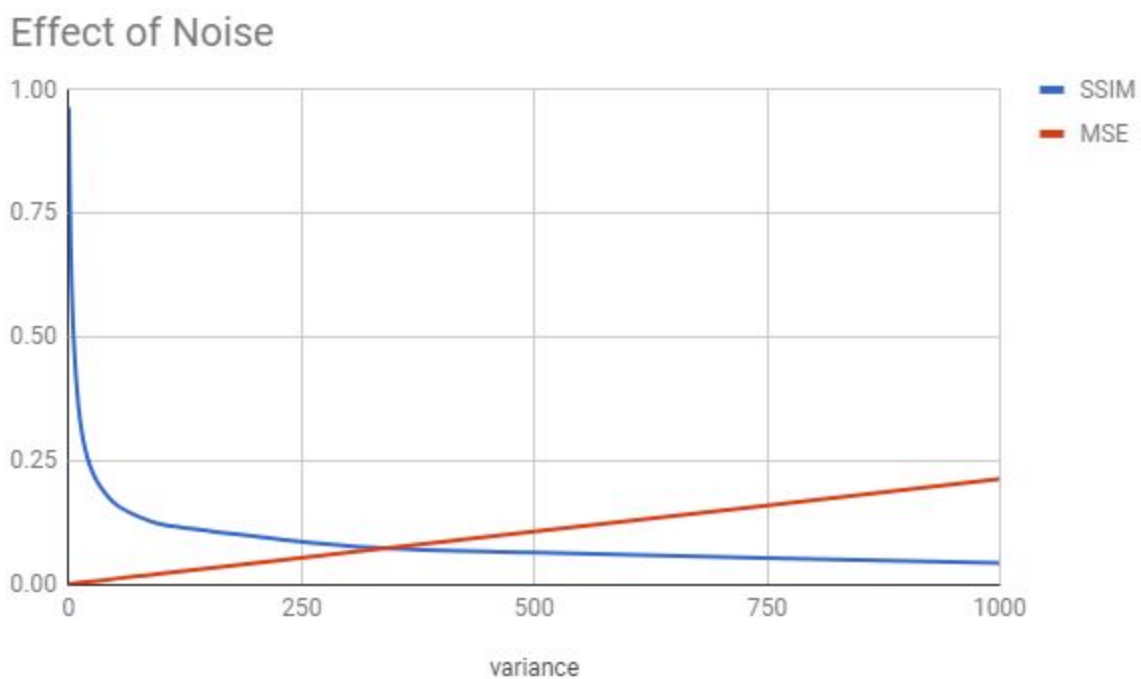


*Figure 3.* compares the SSIM values and this also shows that SSIM values at 0.8 cut-off is significantly higher than at 1.0. From both of these comparisons, it can be concluded that, the image quality is better at 0.8 cut-off coefficients for the given filters. However, for SSIM, the

Shepp-Logan filter has the lowest values for both 0.8 and 1.0 and the Hanning filter had the highest SSIM values in average.

#### 4.4.1 Effect of White Gaussian Noise in Radon Domain

From the graphs shown above, it is shown that the higher degree of noise added, the noisier the reconstructed image is.



SSIM and MSE are employed to evaluate the image qualities of 18 images reconstructed after different degree of noise added. The result is shown by the chart above. The SSIM values show a rapid drop when the variance of Gaussian noise increases from 0 to about 50. The slope becomes less steep afterwards. The MSE value increases with the variance.

#### 4.4.2 Noise Propagation

We first look at the two cases where noise is added to left-half or right-half of the radon transform. The final reconstructions are indeed affected. However, the influence seems to be universal, or at least distributed evenly in the whole picture. The noise added to one half of the radon transform exists in half of the projections, since  $R$  has a shape of [number of detectors,

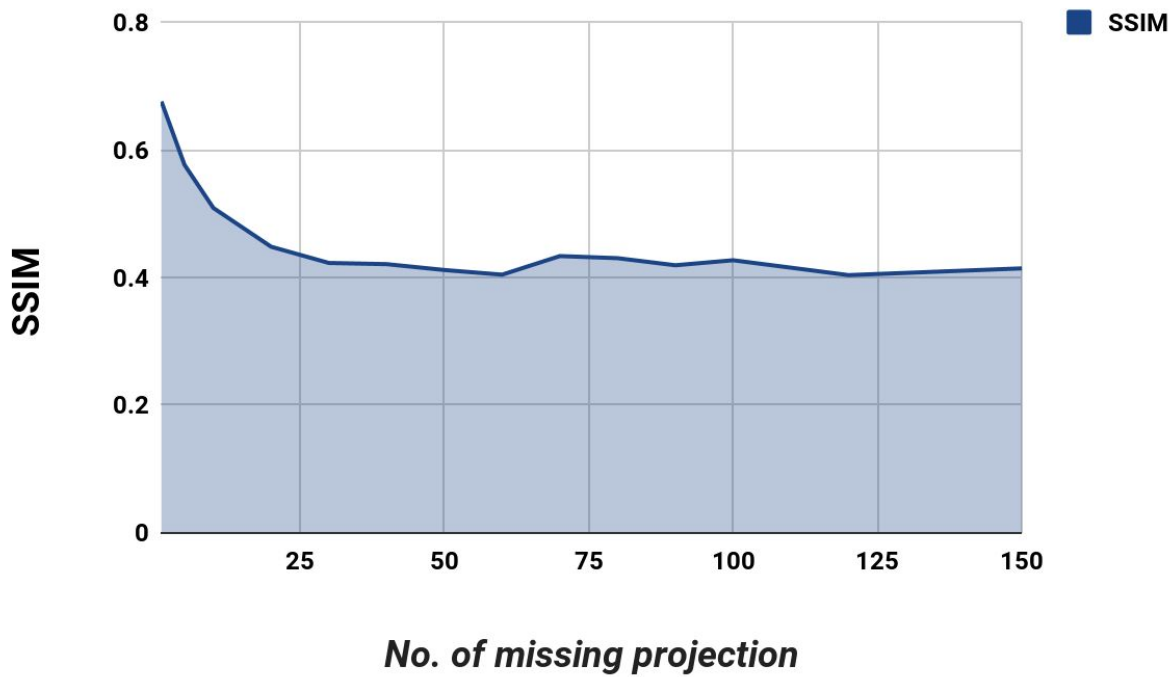
number of angles]. The noise in the final reconstruction is distributed evenly. This is because any projection selected affects all the point reconstruction in the process of backprojection.

Now let's move to the other two cases, where the noise is added to the upper-half and lower-half of the radon transform. The reconstruction shows distinct features in the upper-half and lower-half parts. When we add noise to the upper half or lower half of the radon transform, we are adding noise to one half of the detectors' data. For a certain point in an image, most of its projection will be detected by one half of the detectors only. Therefore the distinct features can be explained.

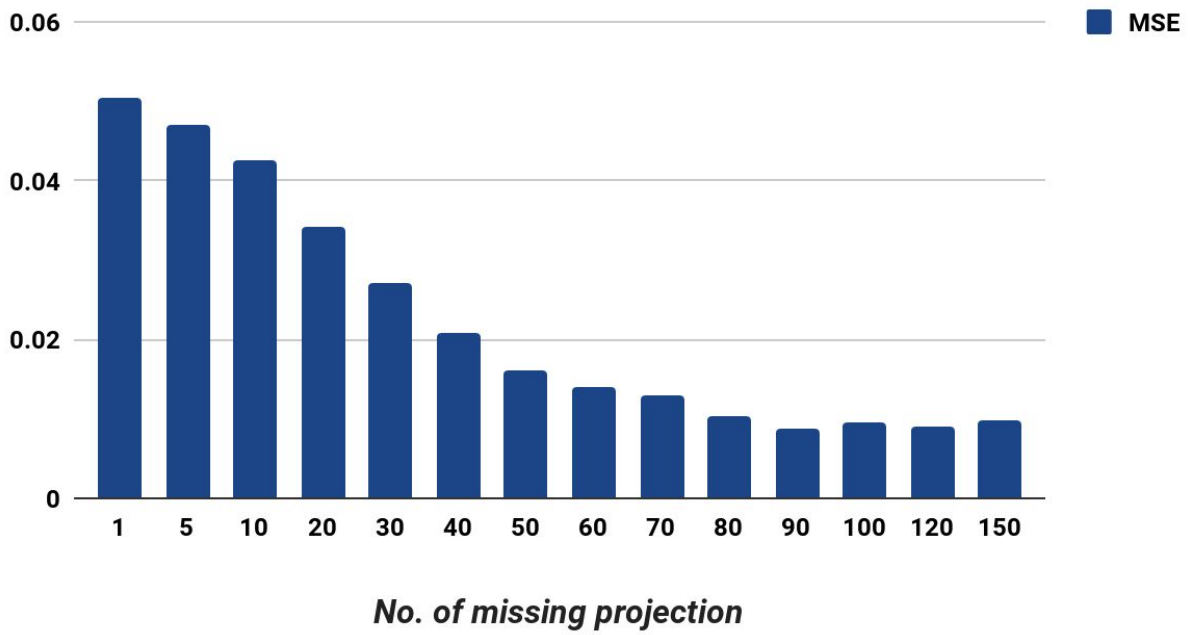
#### **4.5.1 Effect of missing projection**

Total 14 sets of data on no. of missing projection, ranging from 1 to 150, were taken. From the result we can see that the image quality gets lower with higher no. of missing projection. Some cross-lines will appear if the no. of missing projection is high enough. Moreover, the brightness of the image also get lower with higher no. of missing projection. The decreasing of image quality can also be shown by quantification. For the SSIM, it can be seen that the value drop dramatically when the no. of missing projection increase from 1 to 25, and start flattening after that. This indicate that the image quality compared to the original phantom will decrease at first and no longer decrease when the no. of missing projection is large. However, in terms of the MSE, the value surprisingly drops when the no. of missing projection increases, which is contradict to the real case. It could be possibly due to the scaling factor that used for every reconstruction. This scaling factor is strongly related to the number of projection we used. When there are missing projection, the decrease in projection is unconventional in real cases, which do not take into account by the reconstruction. And because MSE is strongly depends on the intensity scaling, therefore the abnormal results happened. This can also be shown that the SSIM is a better evaluation than MSE here.

Effect of missing projection (SSIM)



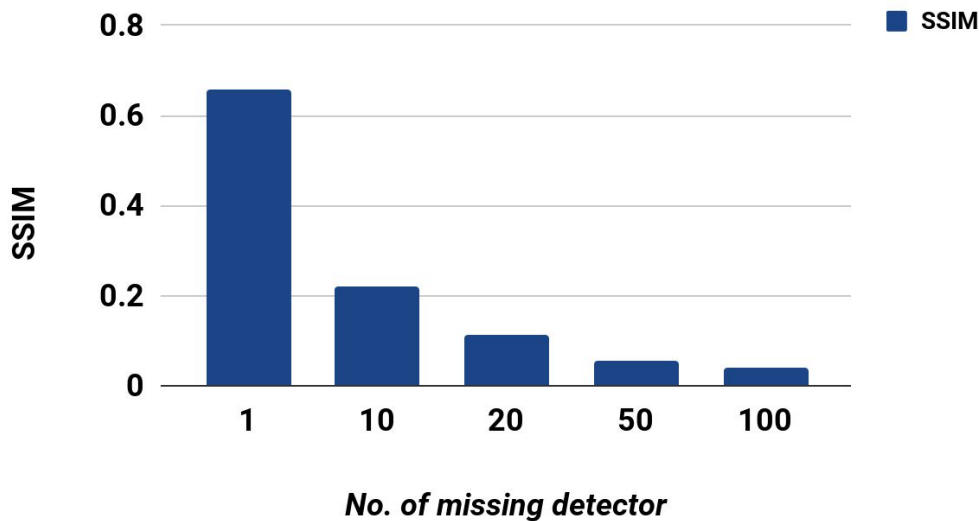
Effect of missing projection (MSE)



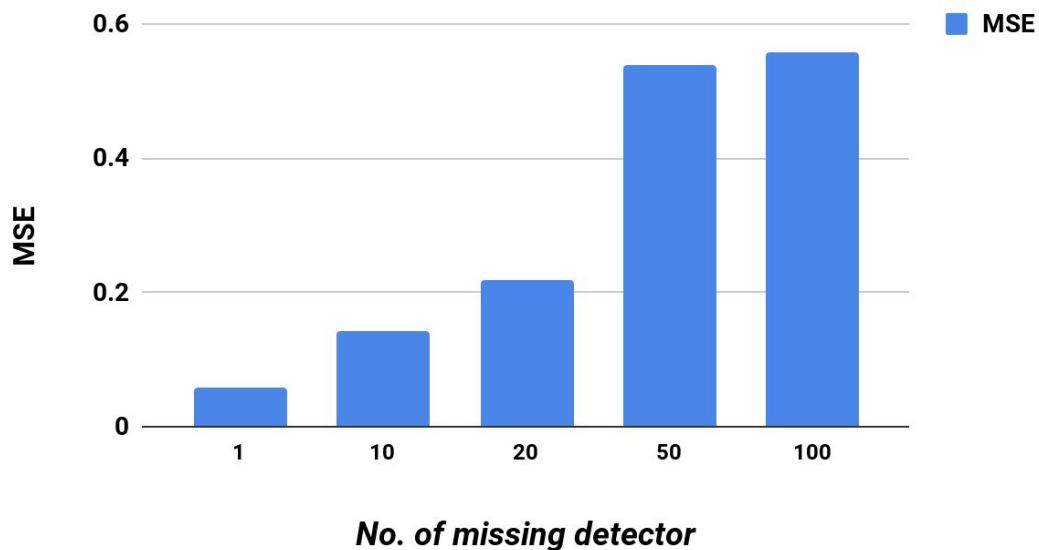
#### 4.5.2 Effect of missing detector

5 set of data were taken. The results show that the image quality drop in the similar way as the effect of missing projection do. From quantify statistics, the SSIM of the image drop rapidly on first two sets of no. of missing detector and start flattening afterwards. On the other hand, the value of MSE increase rapidly from 1 to 50 missing detectors and start to slow down afterward.

**Effect of missing detector (SSIM)**



**Effect of the missing detector (MSE)**





## 4.6 Final Reconstruction



This image is reconstructed using the following parameter:

Projection interval: 0.05

Filter type: Shepp-Logan

Fourier transform coefficient: 6

Cut-off scale: 0.5

The reconstruction evaluations:

MSE: 0.0020

Global SSIM: 0.9670

## 5. Conclusion

Based on the principle of Fourier Filtered Backprojection, we developed a compact algorithm to reconstruct the Shepp-Logan phantom with multiple parameters adjustable. After investigating the effects of filter, noise, sampling, and missing projections on the reconstruction quality, a relatively superlative image was reconstructed. Furthermore, we examined the image quality evaluation and fan-beam projection methods, connecting our project with practical CT scanning.

## 6. Appendix

### Simple proof of the Fourier Filtered Backprojection:

$$f(x, y) = \int_0^{2\pi} \int_0^\infty F_P(w_s, \theta) \exp[iw_s(X \cos \theta + y \sin \theta)] w_s dw_s d\theta \text{ ----- 1}$$

$$f(x, y) = \int_0^\pi \int_0^\infty F_P(w_s, \theta) \exp[iw_s(x \cos \theta + y \sin \theta)] |w_s| dw_s d\theta \text{ ----- 2}$$

$$f(x, y) = \int_0^\pi \left\{ \int_{-\infty}^\infty G(w_s, \theta) \exp[iw_s(x \cos \theta + y \sin \theta)] |w_s| dw_s \right\} d\theta \text{ ----- 3}$$

1 is the polar form of the inverse Fourier Transform

1-2 is by changing limits of integration

2-3 is by using the projection slice theorem

Thus we use equation 3 to reconstruct the image

### Content of group\_project.m:

```
P = phantom(256);

reconstruction = FFB(P, 'shepplogan', 0.05, 6, 0.5);

[global_sim, local_sim] = ssim(reconstruction, P);

%figure, imshow(local_sim,[])
X = ['SSIM VALUE: ', num2str(global_sim)];
disp(X);

err = immse(reconstruction, P);
fprintf('\n The mean-squared error is %0.4f\n', err);

subplot(1,2,1), imshow(P);
title('original phantom');
subplot(1,2,2), imshow(reconstruction);
title('reconstruction');
```

### Content of FFB.m:

```
function final_img = FFB(phantom_img, filter_type, dtheta, coe_transform, cut_off)
    P = phantom_img; %Load the phantom
```

```

[n_row, n_column] = size(P);
theta = 0:dtheta:(180-dtheta); % Set the projection angle

[R, xp] = radon(P, theta); %Perform radon transform, Xp is a vector containing
%the radial coordinates corresponding to each row of R

[num_detectors, num_angles] = size(R);
xp_offset = abs(min(xp)) + 1;

width = 2^nextpow2(num_detectors)*(2^coe_transform); %Get power of 2 for FFT
%at least large enough to fit a column of R
proj_fft = fft(R, width);

ram_lak = 2*[0:(width/2-1), width/2:-1:1]/width;
if (strcmp(filter_type, 'none')==1)
    filter = ones(width, 1);
elseif (strcmp(filter_type, 'ramlak')==1) %Ram-Lak
    filter = ram_lak;
elseif (strcmp(filter_type, 'shepplogan')==1) %Shepp-Logan
    Sinc = abs(sinc(2*(0:width-1)/(2*width)));
    temp = [Sinc(1:(width/2)) Sinc(width/2:-1:1)];
    filter = ram_lak .* temp';
elseif (strcmp(filter_type, 'hamming')==1) %hamming
    Hamming = 0.54 - 0.46*cos(2*pi*(0:width-1)/width);
    temp = [Hamming(width/2:width) Hamming(1:width/2-1)];
    filter = ram_lak .* temp';
elseif (strcmp(filter_type, 'lowpasscosine')==1) %Lowpass Cosine
    Cosine = abs(cos(2*pi*(0:width-1)/(2*width)));
    filter = ram_lak .* Cosine';
elseif (strcmp(filter_type, 'hann')==1) %Hann
    Hann = 0.5 - 0.5*cos(2 * pi * (0:width-1)/width);
    temp = [Hann(width/2:width) Hann(1:width/2-1)];
    filter = ram_lak .* temp';
elseif (strcmp(filter_type, 'blackman')==1) %Blackman
    Blackman = 0.42 - 0.5*cos(2*pi*(0:width-1)/width) + 0.08*cos(4*pi*(0:width-1)/width);
    temp = [Blackman(width/2:width) Blackman(1:width/2-1)];
    filter = ram_lak .* temp';
elseif (strcmp(filter_type, 'bartlett')==1) %Bartlett

```

```

Bartlettcase1= (2*(0:(width-1)/2))/width;
Bartlettcase2= 2-2*(((width-1)/2):width-1)/width);
temp= [Bartlettcase1 Bartlettcase2];
filter= ram_lak .* temp';

elseif (strcmp(filter_type,'barthannwin')==1) %Bartlett-Han
Bartletthan= 0.62-0.48* abs(((0:width-1)/width)-0.5)+0.38*cos(2*pi*(((0:width-1)/width)-0.5);
temp=[Bartletthan(width/2:width) Bartletthan(1:width/2-1)];
filter= ram_lak .* temp';
end

filter(width*cut_off:end)=0;

for w = 1:num_angles
    filtered(:, w) = proj_fft(:, w).*filter;
end

inverse_f = real(ifft(filtered));

final_img = zeros(n_row, n_column);

for iprog=1:num_angles
    G = inverse_f(:, iprog);
    rad = theta(iprog)*pi/180;

    for x=1:n_column
        for y=1:n_row
            t = (x-(n_column/2))*cos(rad) - (y-(n_row)/2)*sin(rad) + xp_offset;

            if (ceil(t) == floor(t))
                g1 = G(ceil(t));
                g2=0;
            else
                g1 = G(ceil(t)) * (t-floor(t));
                g2 = G(floor(t)) * (ceil(t) -t);
            end

            final_img(y,x) = final_img(y,x) +g1 +g2;

```

```

        end
    end
end

```

```

    final_img = (pi/num_angles)*final_img;

```

```

    return
end

```

### **Content of our own radon transform own\_radon.m:**

```

function processed_img = own_radon(img, theta)

[n_row, n_column] = size(img);
Dia = sqrt(n_row + n_column);
row_Pad = ceil(Dia - n_row) + 2;
col_Pad = ceil(Dia - n_column) + 2;
pad = zeros(n_row+row_Pad, n_column+col_Pad);
pad(ceil(row_Pad/2):(ceil(row_Pad/2)+n_row-1), ...
    ceil(col_Pad/2):(ceil(col_Pad/2)+n_column-1)) = img;

angle = length(theta);
processed_img = zeros(size(pad,2), angle);
for i = 1:angle
    tic
    tmpimg = imrotate(pad, 90-theta(i), 'bilinear', 'crop');
    processed_img(:,i) = (sum(tmpimg));
    theta(i)
    toc
end

```

### Content of codes when comparing pencil-beam and fan-beam:

```
I = phantom(256);
D = 200;
dtheta = 1;
[Farc,FposArcDeg,Fangles]
fanbeam(I,D,'FanSensorGeometry','arc','FanRotationIncrement',dtheta);

FposArc = D*tan(FposArcDeg*pi/180);

[R,Rpos]=radon(I,Fangles);

figure
idx = find(Fangles==90);
plot(Rpos,R(:,idx),FposArc,Farc(:,idx))
legend('Radon','Arc')
```

## 7. References

1. MEDE3501 Medical Imaging lecture notes
2. Barkhausen, J., Rody, A. and Schäfer, F. (n.d.). *Digital breast tomosynthesis*.
3. Grangeat, P. (2013). *Tomography*. New York, NY: John Wiley & Sons.
4. Singh, S., Kalra, M., Hsieh, J., Licato, P., Do, S., Pien, H. and Blake, M. (2010). Abdominal CT: Comparison of Adaptive Statistical Iterative and Filtered Back Projection Reconstruction Techniques. *Radiology*, 257(2), pp.373-383.
5. Zhu, Y., Zhao, M., Zhao, Y., Li, H., & Zhang, P. (2012). Noise reduction with low dose CT data based on a modified ROF model. *Optics Express*, 20(16), 17987.