CT Report

Yuzhou Chen

December 5, 2023

0.1 PART 1

The maximun angle of single projection could be calculated implimenting the following inequality:

$$\Delta\theta \leq \frac{2}{N_{\text{samples}}} = \frac{1}{96}$$

Then the approximate number of projections could be gained:

$$N_\theta = N_{\text{proj}} \geq \frac{\theta_{\max} - \theta_{\max}}{\Delta\theta} = 96\pi \approx 301.592$$

Since part 9. and part 10. request N_{proj} to be a multiple of 4, we can Impliment the following value:

$$N_\theta = 304$$

Thus 236 projections would be enough to fully sample the object for a final image of size 192 by 192.

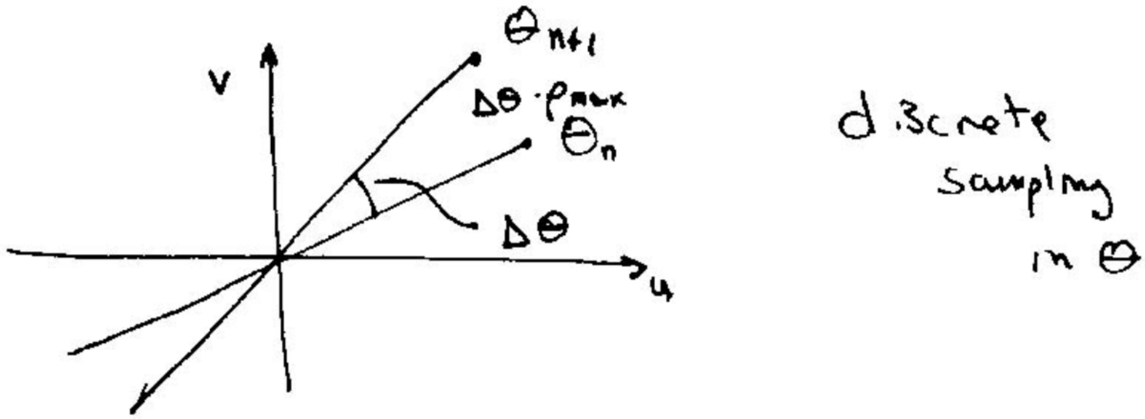


Figure 1: Sampling in θ.

0.2 PART 2

The sinogram could be implemented by the following equation, where $R_0 = 192mm$:

$$g_\theta(R) = \begin{cases} 2\text{amplitude}\sqrt{R_0^2 - (R - x_0\cos\theta - y_0\sin\theta)^2}, & \text{if } |R - x_0\cos\theta - y_0\sin\theta| < R_0 \\ 0, & \text{else} \end{cases}$$

By contrast to the CT-note, we can guess the image is a small cirle located at $x = -53mm$ and a larger circle located at $x = 30mm$, represented as two bright hyperbolic curves. And for the y axis, the small circle looks symmetrical which means $y = 0$. But for the larger circle, it is hard to guess the location.

And the other colored part through the whole image from $x = -75mm$ to $x = -75mm$ is the largest circle which located at the origin of the axis, which shown a symmetrical shape as a rectangle.

In summary, three circle will be represented in the image in the next task.

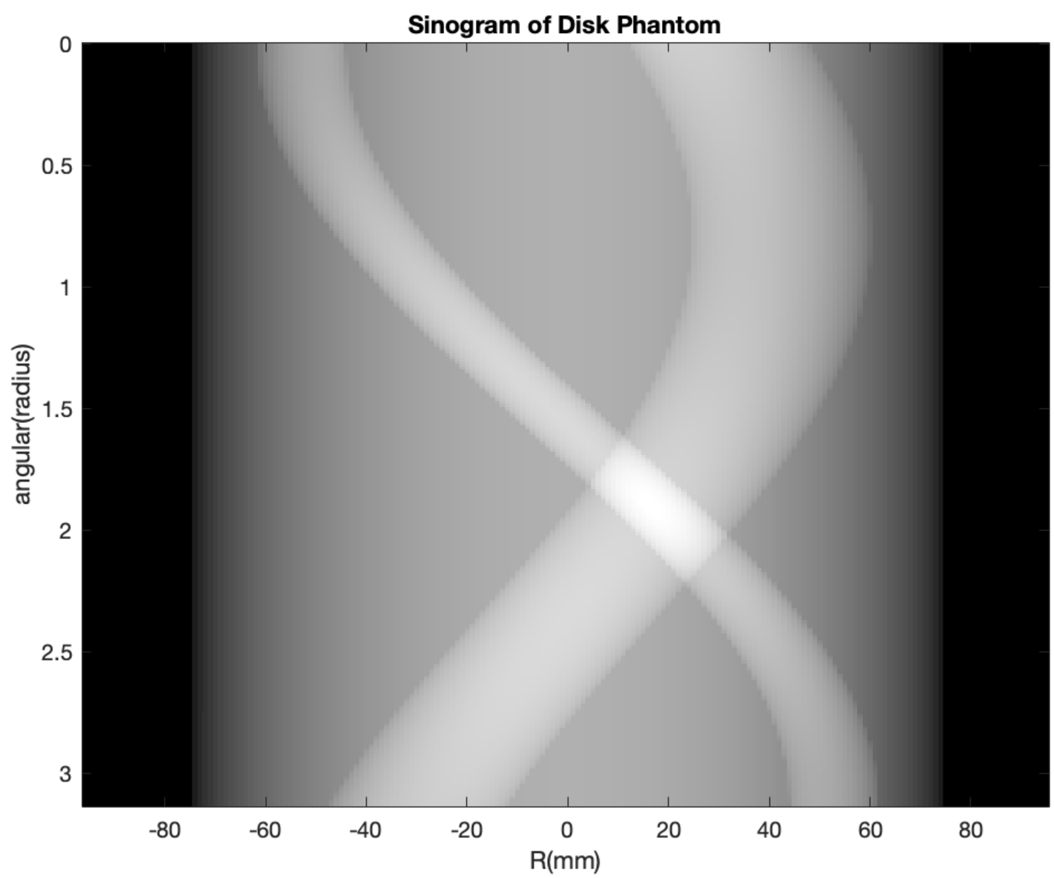


Figure 2: Sinogram.

0.3 PART 3

Impliment the followng equation to finish the rotation of projection at each angular:

$$b_\theta(x, y) = \int_{-\infty}^{\infty} g_\theta(R)\delta(x\cos\theta + y\sin\theta - R)dR$$

Then we need to sum all the projections together by implimenting the following equation:

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

Then the plotting of final image could be shown in the $Figure$.3.The result of such a simple backprojection is typically very blurry and has a starburst artifact due to the high-intensity contributions along the straight paths of the X-rays. This is visible in the second image you provided, where the blurry appearance and overlapping regions of brightness reflect the basic nature of the backprojection without any filtering to refine the result.

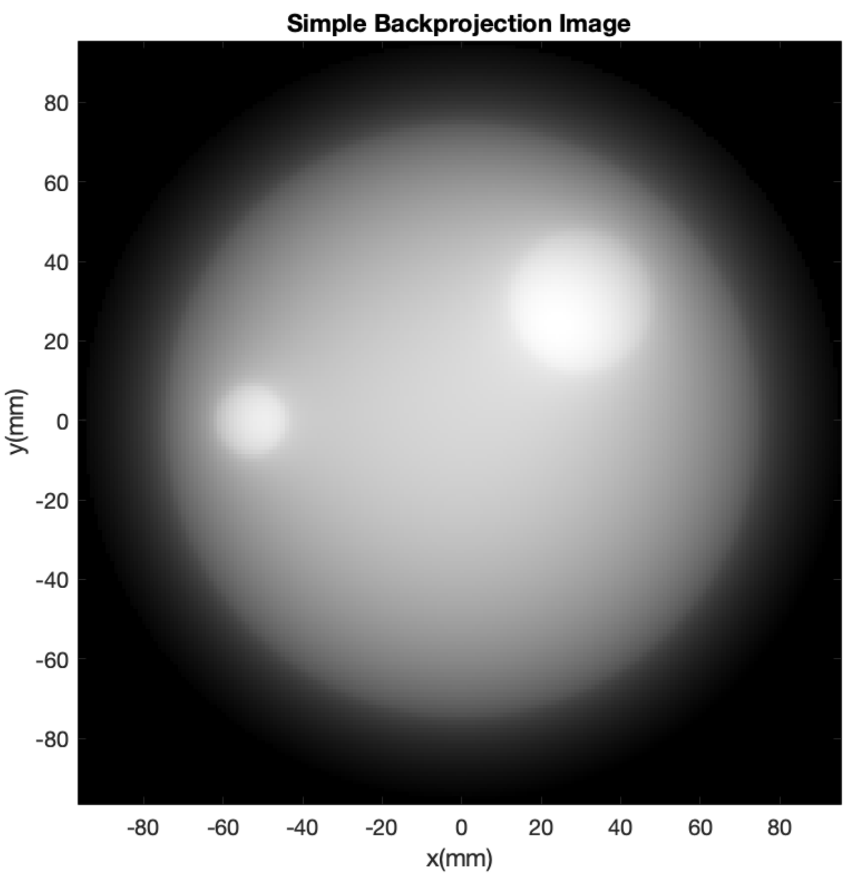


Figure 3: Backprojection Image.

0.4 PART 4

To start the filtering, we can implement the "ramp" filter, which is also a high-pass filter by using the following equation:

$$g'_\theta(R) = \mathcal{F}^{-1}\left\{|\rho|\mathcal{F}_{1D}\{g_\theta(R)\}\right\}$$

From the plot, we can find out that the sharp edges in the filtered projection correspond to the boundaries of the object being imaged.The clear difference between the unfiltered and filtered projections is evident in the steepness and height of the graphs; the filtered projection has much steeper sides and a flat top, which indicates that the filter has enhanced the high-frequency components corresponding to the edges of the disk.

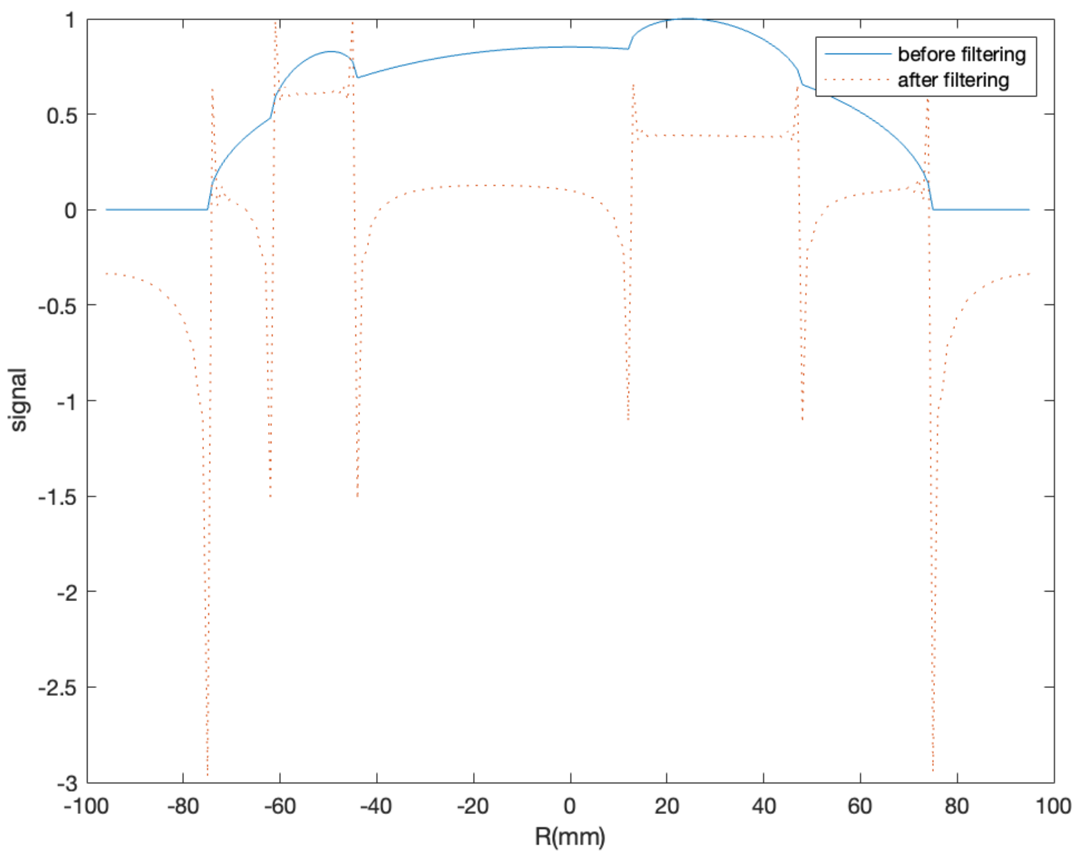


Figure 4: Campared defference before filtered and after filtering.

0.5 PART 5

The filtering process enhances the high-frequency components associated with edges and reduces low-frequency components associated with smooth regions, leading to a clearer and more accurate reconstruction.

And from the reconstruction, here are some different kinds of artifact could be found in the image:

1. **Streak Artifacts**: There may be some faint streaks emanating from the bright areas. These are less pronounced than in an unfiltered image but can still occur if the filter does not fully compensate for all the high-frequency loss during backprojection.
2. **Noise**: There's some level of graininess or noise in the image.
3. **Beam Hardening Artifacts**: These would manifest as dark bands or areas that seem less dense than expected.

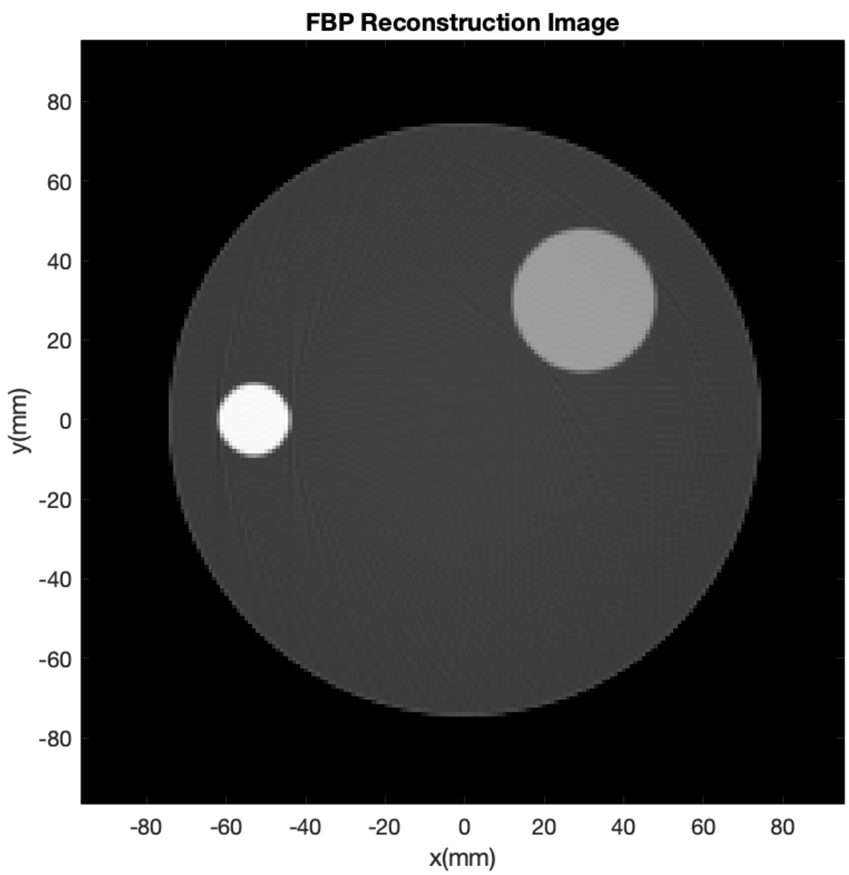


Figure 5: FBP method.

0.6 PART 6

Firstly, take 1D-Fourier transform, which could be done by ft function in $MATLAB$:

$$\mathcal{F}_{1D}\{g_{\theta}(R)\} = G_{\theta}(\rho) = F(\theta, \rho)$$

Then, transfer the scale of image from polar to rectangular coordinates by using *griddat* function:

$$F(\theta, \rho) \Rightarrow \hat{F}(u, v)$$

From the plot, we can notice there are different kinds of artifacts and the image also looks blur with the strip-shape noise which needed further processing.

The following artifacts could be seen:
1. **Streak Artifacts**
2. **Noise**
3. **Unknow Artifact**: Since we are using *griddat* function, there are artifacts generated because of the repeated sampling in the origrin.
4. **Gibbs Phenomenon**: There might be ripples or oscillations near the edges of objects, typically occurring near sharp transitions or boundaries in the image. This is due to the truncation of the Fourier series.

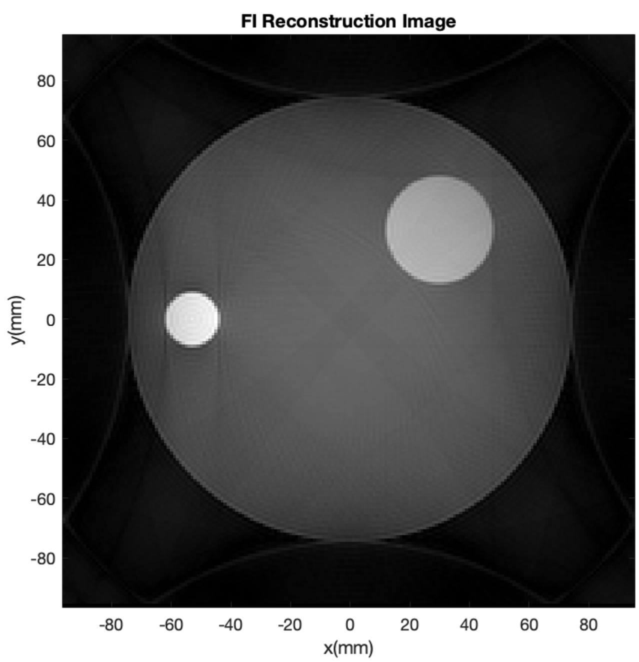


Figure 6: FI Method

0.7 PART 7

By contrast to the FI reconstruction without zero pad, here is some difference:

1. **Noise**: The ZP image seems smoother with less visible noise, which might be due to the interpolation effect of the zero padding that can lead to a smoothing of the image.

2. **Artifacts**: All the tyoes of artifacts was reduced after zero padding

3. **Background**: The zero padding seems to improve the uniformity and consistency of the background, making the objects in the image stand out more clearly against the background.

In summary, zero padding effectively increases the number of samples in the Fourier domain, leading to a finer grid upon inverse transformation. This does not add new information but interpolates existing data, which can improve the image quality by reducing artifacts and enhancing resolution. The improvement in image quality from part 6 to part 7 is consistent with the expected effects of zero padding in Fourier-based image reconstruction methods.

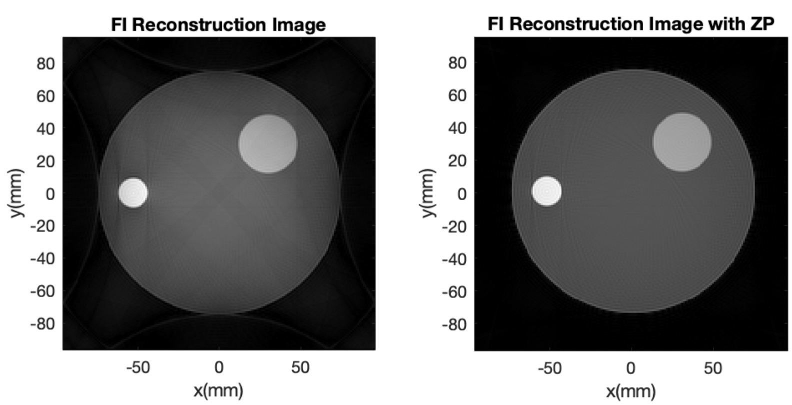


Figure 7: FI method with zero pad.

0.8 PART 8

1. **FBP Profile (Solid Blue Line)**: This profile appears to have the highest overall intensity and shows a clear distinction between the high-intensity areas (presumably corresponding to the more attenuating regions within the phantom) and the background. However, there is some fluctuation in intensity near the edges, which could be indicative of the Gibbs phenomenon or other reconstruction artifacts like noise.

2.**FI Profile (Dashed Orange Line)**: The FI profile without zero padding shows significantly lower intensity, suggesting that the image might be under-scaled compared to the FBP image. The peaks are also less sharp, which could be due to a smoothing effect inherent to the interpolation process.

3.**FI ZP Profile (Yellow Line)**: The profile of the FI image with zero padding seems to have a slightly higher intensity than the FI without ZP and is smoother, indicating that zero padding may help in reducing noise and improving the uniformity of the reconstructed image. However, there is a noticeable peak on the right side that does not align with the FBP profile, which might suggest an artifact introduced by the zero padding or the interpolation process.

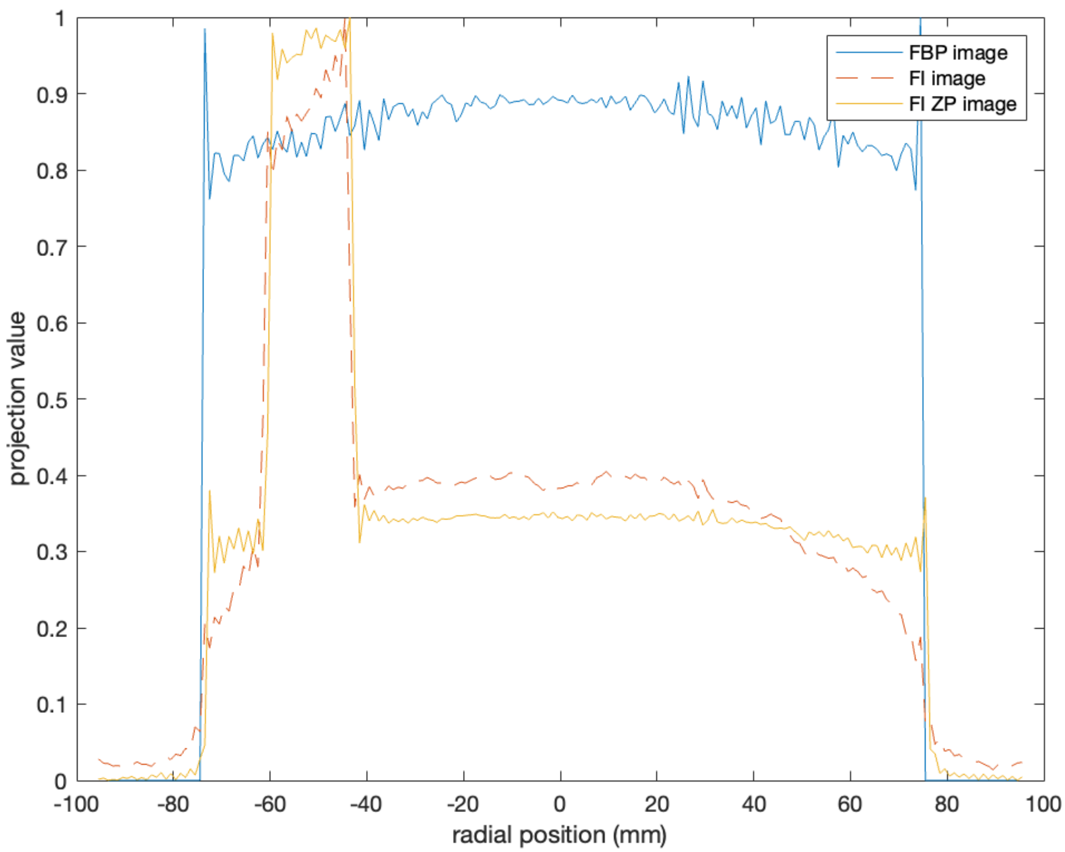


Figure 8: Compared the signals based on different method.

0.9 PART 9

1. **FBP Reconstruction with Subsampling (Left Image)**: This image appears to have streak artifacts. Streaks arise due to the incomplete projection information which causes inconsistencies in the backprojected image. The edges of the objects in the image also appear less sharp, which is an indication of reduced spatial resolution due to subsampling.

2. **FI Reconstruction with ZP and Subsampling (Right Image)**: The FI method with zero padding seems to mitigate some of the streaking seen in the FBP image. The edges of the objects appear smoother, which suggests that the interpolation and zero padding may help compensate for the lower sampling rate to some extent. However, there may still be a loss of detail, and the image could exhibit interpolation artifacts that manifest as blurring or distortions.

In summary, both image looks less detailed with more artifacts due to the subsampling, but the image based on the FI mehod was shown better performance than FBP reconstruction.

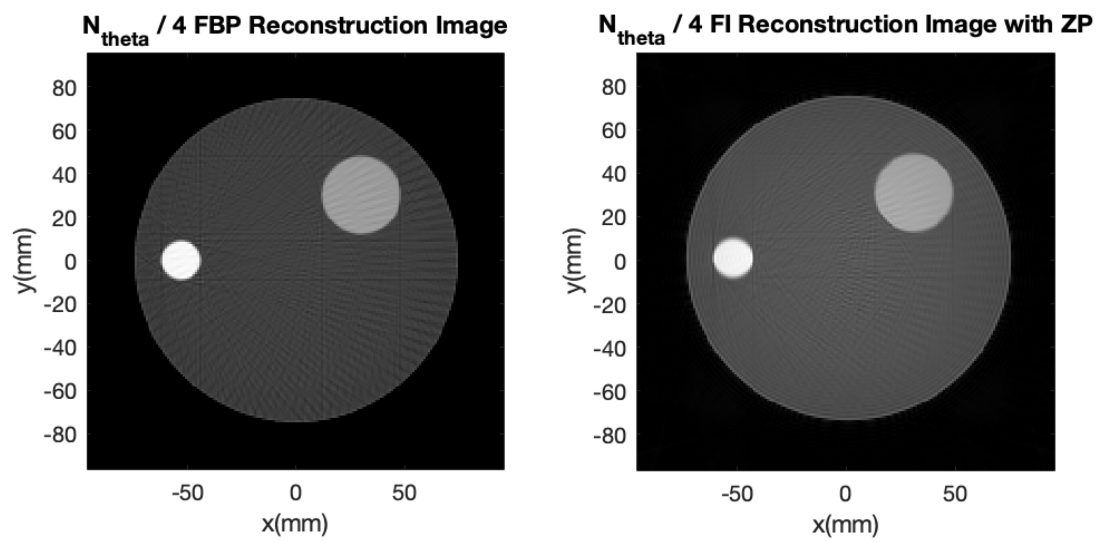


Figure 9: $\frac{1}{4}$ subsampling.

0.10 PART 10

After the incomplete projections, the image looks far incorrect than the expected image and the following artifacts could be detected:

1. **Streak Artifacts**: Due to the limited number of angles, streaking artifacts become much more obvious. These appear as lines radiating from high-density areas to the rest of the image.

2. **Blurring**:The details in the image are likely to be less sharp because fewer projections mean less information to accurately reconstruct the image.

3. **Increased Noise**

In summary, From the provided image, we can observe a significant distortion and streaking, as expected. The limited projection data causes the reconstruction algorithm to inadequately represent the object, resulting in a loss of detail and the introduction of significant streaking artifacts. The boundaries between different structures within the image are also likely less defined, and there might be areas where the reconstruction has failed to represent the actual attenuation values correctly.

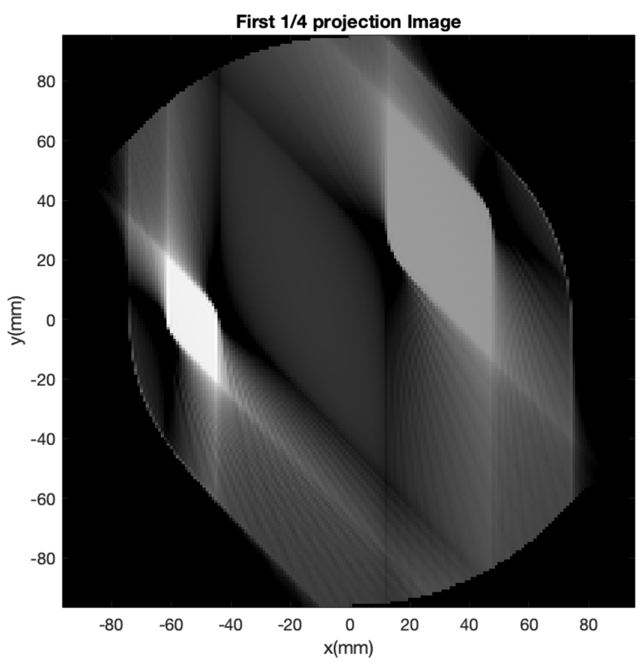


Figure 10: $\frac{1}{4}$ subprojection.

0.11 PART 11

From the figure 11, we can find the image is "Minion" from the "Despicable Me" film series.

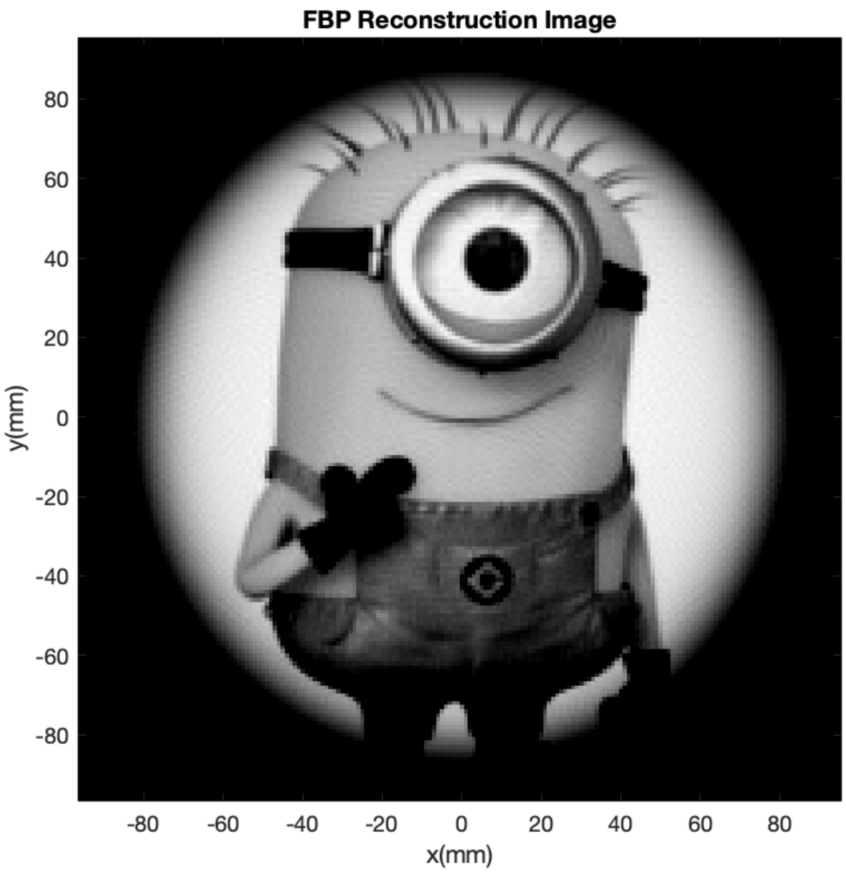


Figure 11: Mystery Image.

From the Sinogram, it is really hard to guess the shape of image after deconstruction. However, there are still some details could be noticed, such as the huge circle in $(R, \theta) = (0, \pi)$, the asymmetric hyperbolic curves starts approximately from $(R, \theta) = (10, 0)$ and ends to $(R, \theta) = (10, 2\pi)$ seems like the boundaries of the single eye and goggles.

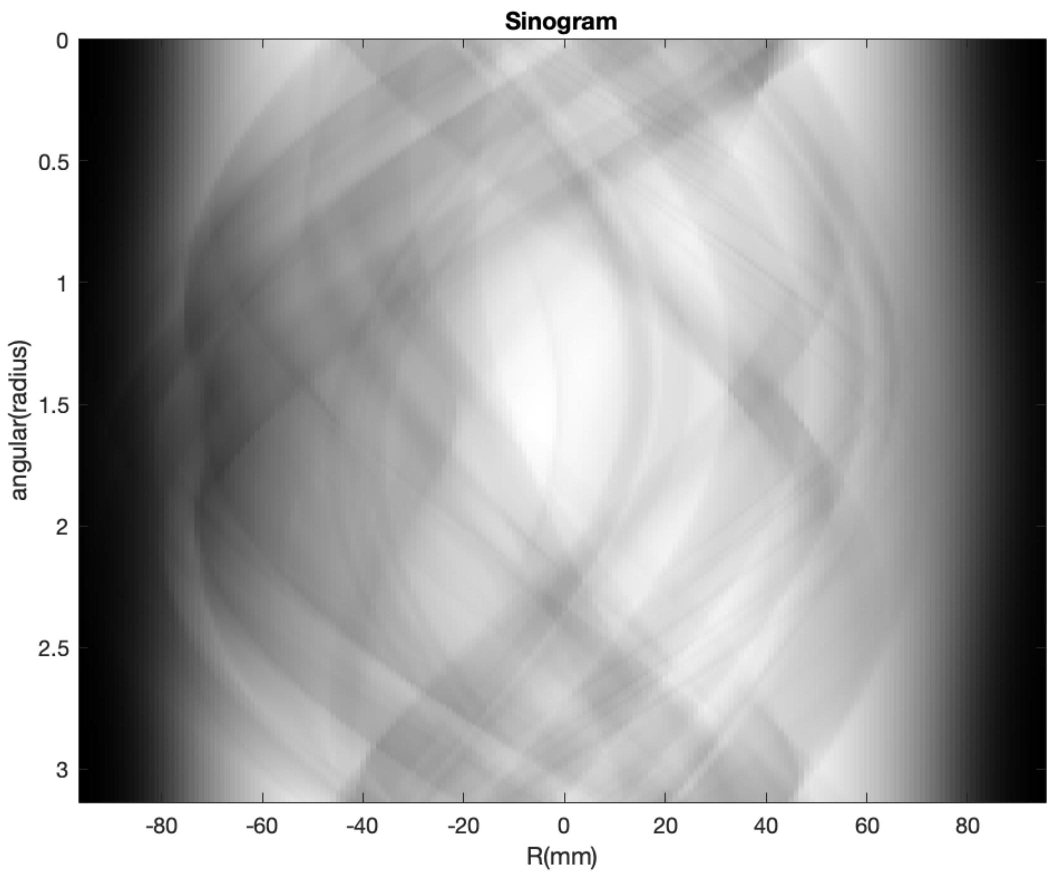


Figure 12: Mystery Sinogram.

```
%
%       template for BME/EECS 516 tomography project
%
%       replace all ?'s
%

%       parameters for the 3 disk phantom
%       x,y center, radius, 'amplitude' (e.g. attenuation coefficient)
clear all
clc
circ = [0 0 75 0.1; 30 30 18 0.2; -53 0 9 0.3];
nobj = size(circ,1);

%
%       image parameters
%

nx = 192; ny = 192;
dx = 1;                          % 1 mm / pixel

%
%       geometry parameters
%

nr = 192;       dr = 1;         % # of radial samples, ray spacing

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Question 1 - determine number of angular samples
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% calculate the number of angular samples needed
na = 304;
r = dr*[-nr/2:nr/2-1]';
ang = [0:(na-1)]'/na * pi;      % angular sample positions
disp(sprintf('number of angles = %g', na))

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Question 2 - compute sinogram for disk phantom (NO CHANGES NEEDED)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  rr = r(:,ones(1,na));

  sinogram1 = zeros(nr, na);
  for ii=1:nobj
    cx = circ(ii,1);            % center of object in x
    cy = circ(ii,2);            % center of object in y
    rad = circ(ii,3);           % radius of object
    amp = circ(ii,4);           % amplitude of object (in atten

    % correct amplitude for overlying objects
    if ii > 1, amp = amp - circ(1,4);, end

    % center location of object for each projection
    tau = cx * cos(ang) + cy * sin(ang);
    tau = tau(:,ones(1,nr))';

    % find all locations where "rr" is within "rad" of "tau"
    t = find( (rr-tau).^2 <= rad.^2 );

    % update the sinogram with length of segment (a bit of geometry...)
    sinogram1(t) = sinogram1(t)+amp*2*sqrt(rad^2-(rr(t)-tau(t)).^2);

  end


%
% Output Image of Singram
%
```

```
figure(1)
close(figure(1))
figure(2)
imagesc(r,ang,sinogram1'); colormap('gray')
title('Sinogram of Disk Phantom')
xlabel('R(mm)')
ylabel('angular(radius)')

sinogram = sinogram1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Question 3 - Implement plain backprojection
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


  bpimage = zeros(nx,ny);
  for ia = 1:na
    %disp(sprintf('angle %g of %g', ia, na))

    % first backproject at theta = 0
    tmpim = repmat(sinogram(:,ia),1,nr);
    % now rotate the projection
    rotim  = imrot3(tmpim, ang(ia,:), 'bilinear');
    bpimage = bpimage + rotim;
  end

%
% Display Image
%\
dy = 1;
x = dx*[-nx / 2 : nx / 2-1];
y = dy*[-ny / 2 : ny / 2-1];
figure(3)
imagesc(x,y,bpimage'); colormap('gray'); axis('image');axis('xy');
title('Simple Backprojection Image')
xlabel('x(mm)')
ylabel('y(mm)')


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Question 4 - Filter Projections
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  % zero pad sinogram (if using Fourier methods)


  % filter the sinogram
  % !!!Warning - fftshift works differently for vectors and matrices!!

%
% Plot Filtered Sinogram
%
projection = zeros([nr na]);
F_projection_filtered = zeros([nr na]);
projection_filtered = zeros([nr na]);
ramp_filter = abs(r);
for ia = 1:na
    projection(:, ia) = sinogram(:, ia);
    F_projection(:, ia) = fftshift(fft(projection(:, ia))); % Move to frequency domain and shift zero frequency to center
    for ir = 1 : nr
        F_projection_filtered(ir,ia) = F_projection(ir,ia) * ramp_filter(ir,:); % Apply the filter
    end
    projection_filtered(:,ia) = ifft(ifftshift(F_projection_filtered(:,ia))); % Move back to spatial domain
end

sinogrampad = real(sinogram);
sinogramfilt = real(projection_filtered);

figure(4)
```

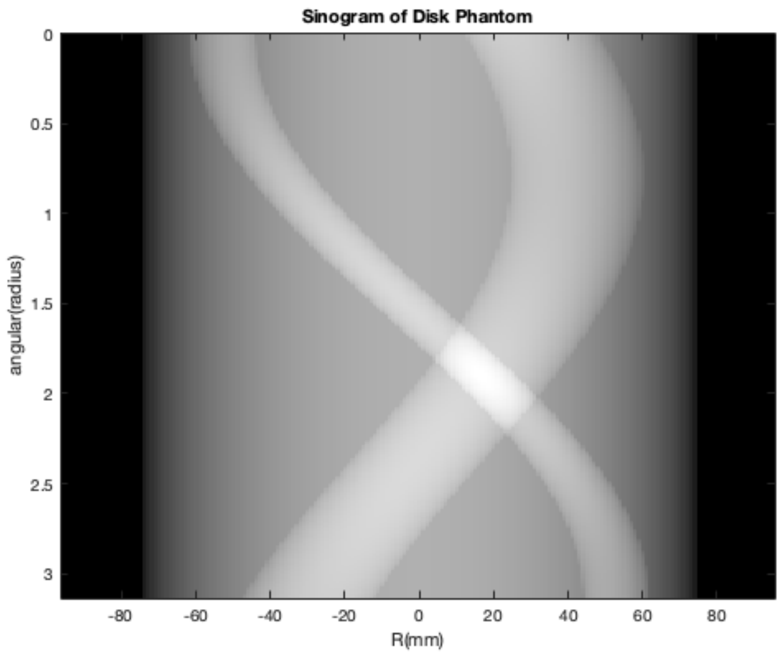
```
plot(r, sinogrampad(:,1)./max(sinogram(:,1)), '-',...
   r, sinogramfilt(:,1)./max(sinogramfilt(:,1)),':');
xlabel('R(mm)');
ylabel('signal');
legend('before filtering','after filtering')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Question 5 - Backproject the filtered sinogram
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

imagefbp = zeros(nx,ny);
for ia = 1:na
    tmpimfbp = repmat(sinogramfilt(:,ia),1,nr);
    rotimfbp  = imrot3(tmpimfbp, ang(ia,:), 'bilinear');
    imagefbp = imagefbp + rotimfbp;
end

for ix = 1 : nx
    for iy = 1 : ny
        if imagefbp(ix,iy) < 0
            imagefbp(ix,iy) = 0;
        end
    end
end


%
% Display Reconstructed Image with Negatives Set to Zero
%

figure(5)
imagesc(x,y,max(imagefbp',0)); colormap('gray'); axis('image');axis('xy');
title('FBP Reconstruction Image')
xlabel('x(mm)')
ylabel('y(mm)')


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Question 6 - Generate Fourier Interpolation Image
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Note that one can use either interp2 or griddata
% template is for interp2, but griddata is fine
% Fourier locations in of gridded data
fov = dr*nr;
kx = 1/fov.*[-nx/2:nx/2-1];
ky = 1/fov.*[-ny/2:ny/2-1];
kr = 1/fov.*[-nr/2:nr/2-1];
[kaa,krr] = meshgrid(ang,kr);
[kxx,kyy] = meshgrid(kx,ky);
kxxin = -krr.*sin(kaa);
kyyin = krr.*cos(kaa);

sinogramft = ft(sinogram);
fdata = griddata(kxxin,kyyin,sinogramft,kxx,kyy,'linear');
t = find(isnan(fdata));
fdata(t) = zeros(size(t));
imagefi = ift2(fdata);
imagefi  = imrot3(imagefi, pi / 2, 'bilinear');

figure(6)

imagesc(x,y,abs(imagefi)); colormap('gray'); axis('image'); axis('xy')
title('FI Reconstruction Image')
xlabel('x(mm)')
ylabel('y(mm)')
```

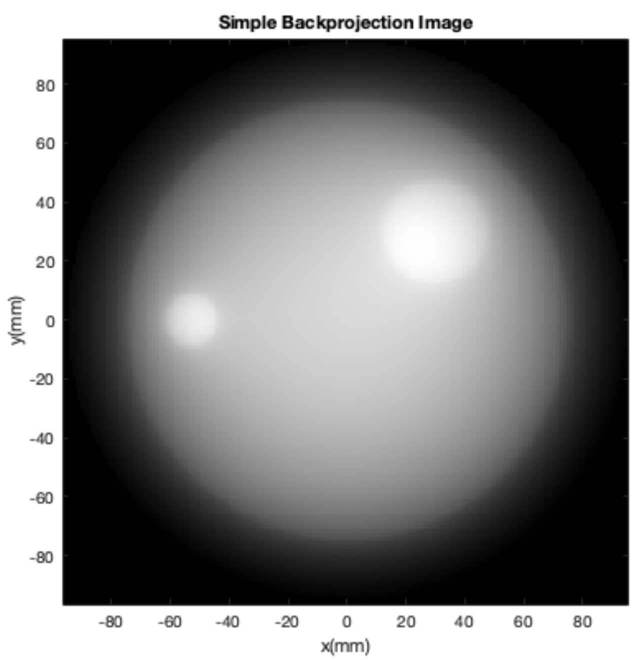
```
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %
% % Question 7 - Generate Fourier Interpolation Image using zeropadding
% %
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fov_zp = 2 * dr * nr;
kx_zp = 1/fov_zp.*[-nx:nx-1];
ky_zp = 1/fov_zp.*[-ny:ny-1];
kr_zp = 1/fov_zp.*[-nr:nr-1];
[kaa_zp,krr_zp] = meshgrid(ang,kr_zp);
[kxx_zp,kyy_zp] = meshgrid(kx_zp,ky_zp);
kxxin_zp = -krr_zp.*sin(kaa_zp);
kyyin_zp = krr_zp.*cos(kaa_zp);

sinogram_zp = padarray(sinogram, [(384-nr)/2 0], 'post');
sinogram_zp = ft(padarray(sinogram_zp, [(384-nr)/2 0], 'pre'));

fdata_zp = griddata(kxxin_zp,kyyin_zp,sinogram_zp,kxx_zp,kyy_zp,'linear');
t = find(isnan(fdata_zp));
fdata_zp(t) = zeros(size(t));
imagefizp = ift2(fdata_zp);
imagefizp  = imrot3(imagefizp, pi / 2, 'bilinear');
imagefizp = imagefizp(nx / 2 : (nx / 2 + nx -1), ny / 2 : (ny / 2 + ny -1));

figure(7)
subplot(1,2,1)
imagesc(x,y,abs(imagefi)); colormap('gray'); axis('image'); axis('xy')
title('FI Reconstruction Image')
xlabel('x(mm)')
ylabel('y(mm)')
subplot(1,2,2)
imagesc(x,y,abs(imagefizp)); colormap('gray'); axis('image'); axis('xy')
title('FI Reconstruction Image with ZP')
xlabel('x(mm)')
ylabel('y(mm)')


% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %
% % Question 8 - Plot profiles through reconstructed images
% %
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
if 1~=exist('linefi')
    y = dx * ([1:ny]'-(ny+1)/2);
    linefbp = abs(imagefbp(ny/2, :));
    linefi = abs(imagefi(ny/2, :));
    linefizp = abs(imagefizp(ny/2, :));
    figure(8)
    plot(y, linefbp/max(linefbp),'-', y, linefi/max(linefi),'--',...
    y, linefizp/max(linefizp),'-');
    xlabel('radial position (mm)');
    ylabel('projection value');
    legend('FBP image', 'FI image','FI ZP image')
end
%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %
% % Question 9 - Do again for subsampled image
% %
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
na = na / 4;
x = dx*[-nx / 2 : nx / 2-1];
y = dy*[-ny / 2 : ny / 2-1];
sinosubsamp = sinogram(:,1:4:end);
ang_9 = ang(1:4:end,:);
projection = zeros([nr na]);
F_projection_filtered = zeros([nr na]);
projection_filtered = zeros([nr na]);
ramp_filter = abs(r);
for ia = 1:na
    projection(:, ia) = sinosubsamp(:, ia);
```

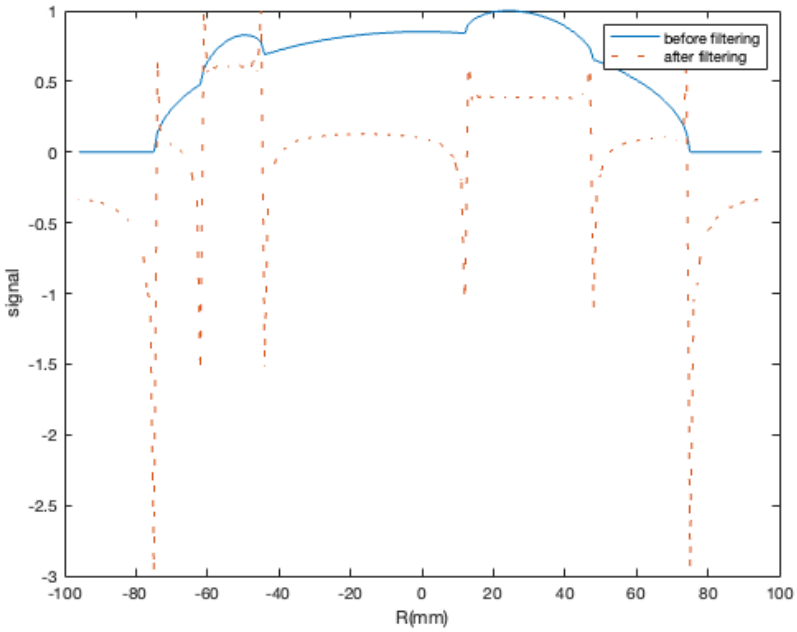
```
    F_projection(:, ia) = fftshift(fft(projection(:, ia)));
    for ir = 1 : nr
        F_projection_filtered(ir,ia) = F_projection(ir,ia) * ramp_filter(ir,:);
    end
    projection_filtered(:,ia) = ifft(ifftshift(F_projection_filtered(:,ia)));
end

sinogramfilt = real(projection_filtered);

imagefbp = zeros(nx,ny);
for ia = 1:na
    tmpimfbp = repmat(sinogramfilt(:,ia),1,nr);
    rotimfbp  = imrot3(tmpimfbp, ang_9(ia,:), 'bilinear');
    imagefbp = imagefbp + rotimfbp;
end

for ix = 1 : nx
    for iy = 1 : ny
        if imagefbp(ix,iy) < 0
            imagefbp(ix,iy) = 0;
        end
    end
end

[kaa_zp,krr_zp] = meshgrid(ang_9,kr_zp);
[kxx_zp,kyy_zp] = meshgrid(kx_zp,ky_zp);
kxxin_zp = -krr_zp.*sin(kaa_zp);
kyyin_zp = krr_zp.*cos(kaa_zp);

sinogram_zp = padarray(sinosubsamp, [nr/2 0], 'post');
sinogram_zp = ft(padarray(sinogram_zp, [nr/2 0], 'pre'));

fdata_zp = griddata(kxxin_zp,kyyin_zp,sinogram_zp,kxx_zp,kyy_zp,'linear');
t = find(isnan(fdata_zp));
fdata_zp(t) = zeros(size(t));
imagefizp = ift2(fdata_zp);
imagefizp  = imrot3(imagefizp, pi / 2, 'bilinear');
imagefizp = imagefizp(nx / 2 : (nx / 2 + nx -1), ny / 2 : (ny / 2 + ny -1));

figure(9)
subplot(1,2,1)
imagesc(x,y,max(imagefbp',0)); colormap('gray'); axis('image');axis('xy');
title('N_{theta} / 4 FBP Reconstruction Image')
xlabel('x(mm)')
ylabel('y(mm)')
subplot(1,2,2)
imagesc(x,y,abs(imagefizp)); colormap('gray'); axis('image'); axis('xy')
title('N_{theta} / 4 FI Reconstruction Image with ZP')
xlabel('x(mm)')
ylabel('y(mm)')


% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %
% % Question 10 - Do again for lmited view angles
% %
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sino4 = sinogram(:,1:na);
ang_10 = ang(1:na,:);
projection = zeros([nr na]);
F_projection_filtered = zeros([nr na]);
projection_filtered = zeros([nr na]);
ramp_filter = abs(r);
for ia = 1:na
    projection(:, ia) = sino4(:, ia);
    F_projection(:, ia) = fftshift(fft(projection(:, ia)));
    for ir = 1 : nr
        F_projection_filtered(ir,ia) = F_projection(ir,ia) * ramp_filter(ir,:);
    end
    projection_filtered(:,ia) = ifft(ifftshift(F_projection_filtered(:,ia)));
end

sinogramfilt = real(projection_filtered);
```

```
imagefbp = zeros(nx,ny);
for ia = 1:na
    tmpimfbp = repmat(sinogramfilt(:,ia),1,nr);
    rotimfbp  = imrot3(tmpimfbp, ang_10(ia,:), 'bilinear');
    imagefbp = imagefbp + rotimfbp;
end

for ix = 1 : nx
    for iy = 1 : ny
        if imagefbp(ix,iy) < 0
            imagefbp(ix,iy) = 0;
        end
    end
end

figure(10)
imagesc(x,y,max(imagefbp',0)); colormap('gray'); axis('image');axis('xy');
title('First 1/4 projection Image')
xlabel('x(mm)')
ylabel('y(mm)')


% %%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %
% % Question 11 - load and reconstruct mystery object
% %
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load mys23;
na = 360;
projection = zeros([nr na]);
F_projection_filtered = zeros([nr na]);
projection_filtered = zeros([nr na]);
ramp_filter = abs(r);
for ia = 1:na
    projection(:, ia) = sinogram(:, ia);
    F_projection(:, ia) = fftshift(fft(projection(:, ia)));
    for ir = 1 : nr
        F_projection_filtered(ir,ia) = F_projection(ir,ia) * ramp_filter(ir,:);
    end
    projection_filtered(:,ia) = ifft(ifftshift(F_projection_filtered(:,ia)));
end
sinogrampad = real(sinogram);
sinogramfilt = real(projection_filtered);

imagefbp = zeros(nx,ny);
for ia = 1:na
    tmpimfbp = repmat(sinogramfilt(:,ia),1,nr);
    rotimfbp  = imrot3(tmpimfbp, ang(:,ia), 'bilinear');
    imagefbp = imagefbp + rotimfbp;
end

for ix = 1 : nx
    for iy = 1 : ny
        if imagefbp(ix,iy) < 0
            imagefbp(ix,iy) = 0;
        end
    end
end

figure(11)
imagesc(x,y,max(imagefbp',0)); colormap('gray'); axis('image');axis('xy');
title('FBP Reconstruction Image')
xlabel('x(mm)')
ylabel('y(mm)')

figure(12)
imagesc(r,ang,sinogram'); colormap('gray')
title('Sinogram')
xlabel('R(mm)')
ylabel('angular(radius)')
```

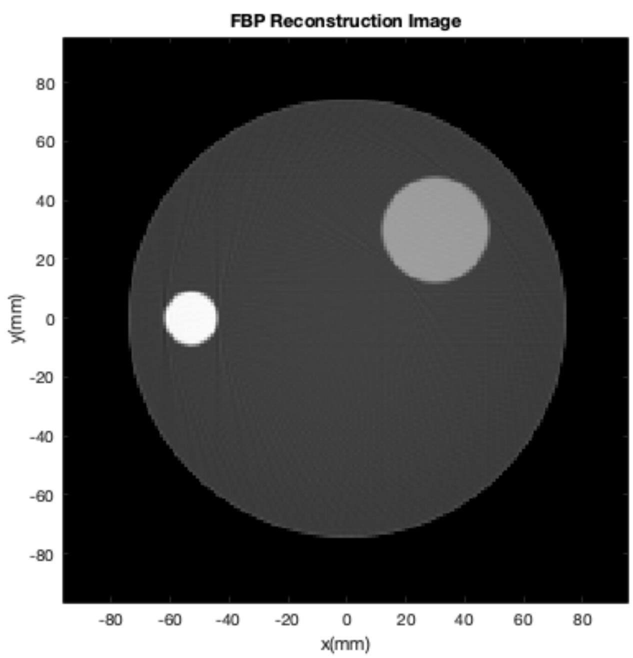
```
number of angles = 304
Warning: Duplicate data points have been detected and removed - corresponding
values have been averaged.
Warning: Duplicate data points have been detected and removed - corresponding
values have been averaged.
Warning: Duplicate data points have been detected and removed - corresponding
values have been averaged.
```



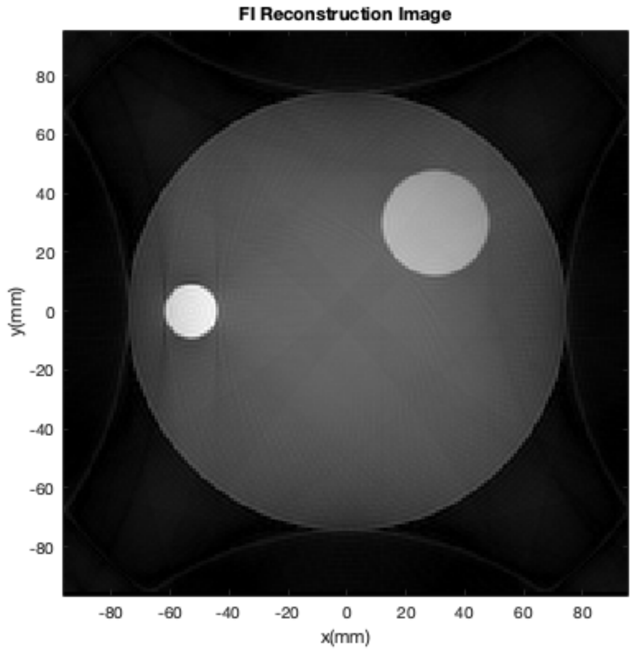




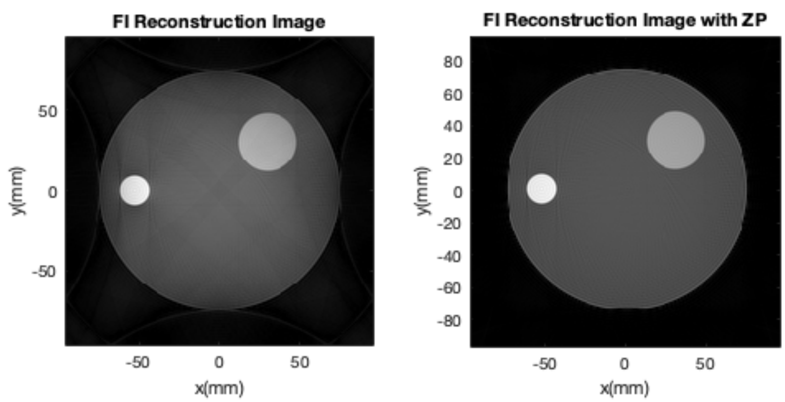
before filtering
after filtering
signal
R(mm)



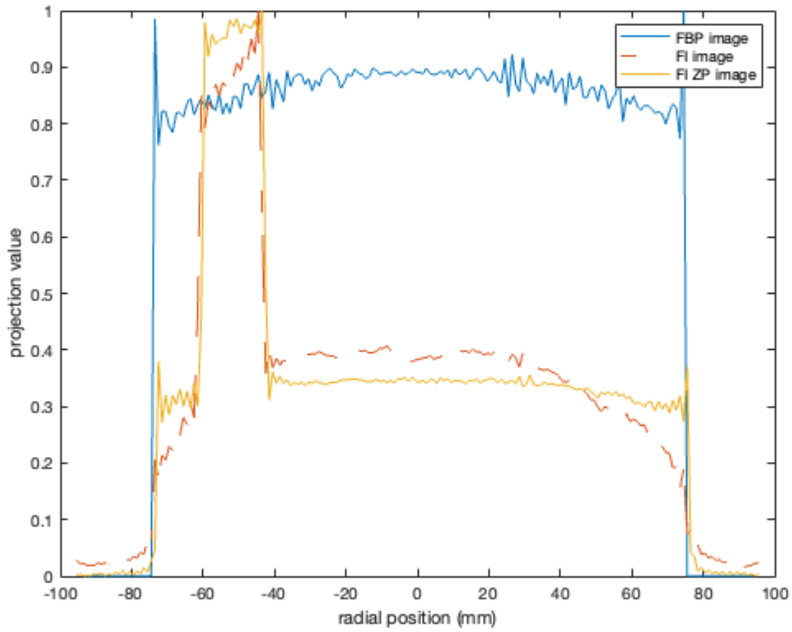
FBP Reconstruction Image
y(mm)
x(mm)



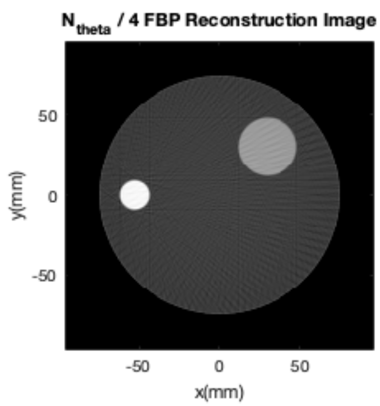
FI Reconstruction Image
y(mm)
x(mm)



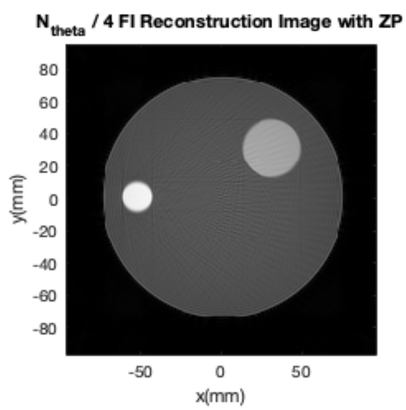
FI Reconstruction Image
FI Reconstruction Image with ZP
y(mm)
x(mm)



FBP image
FI image
FI ZP image
projection value
radial position (mm)

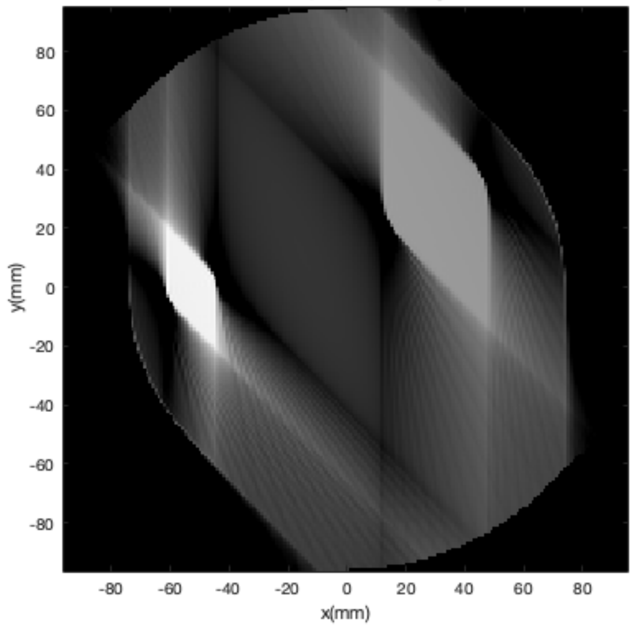


N_{theta} / 4 FBP Reconstruction Image
y(mm)
x(mm)

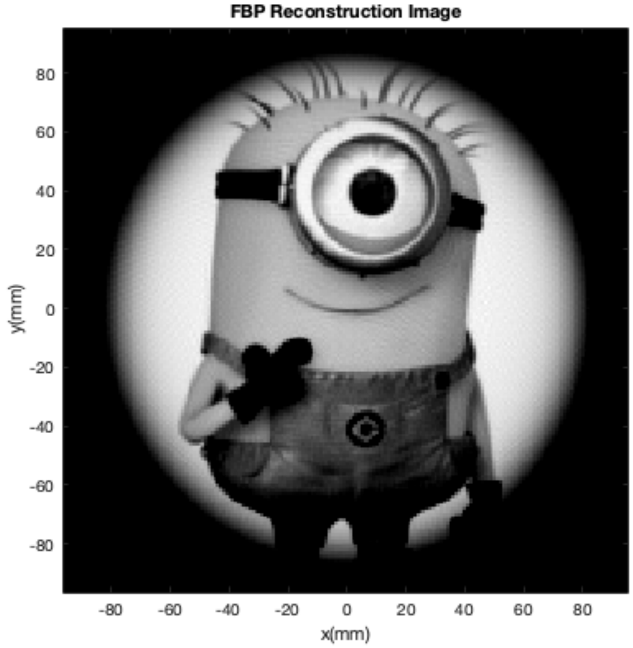


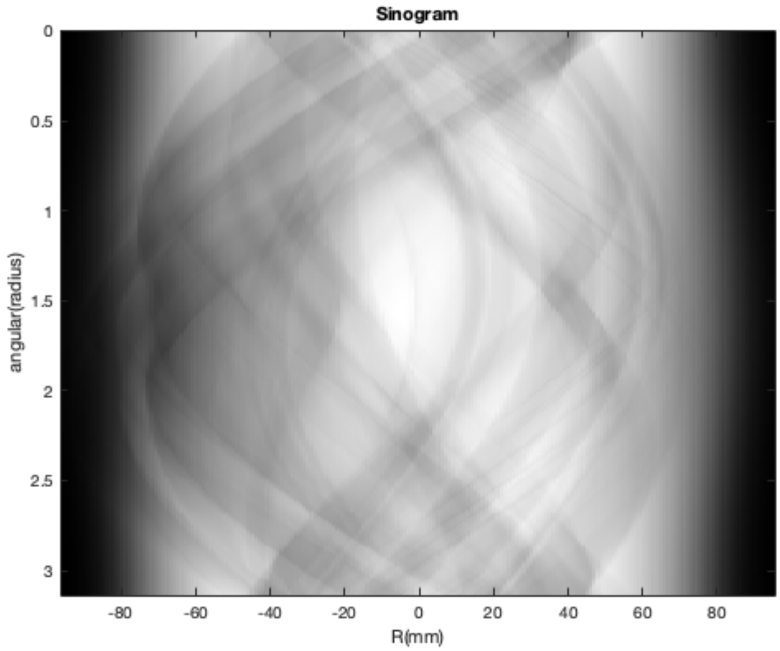
N_{theta} / 4 FI Reconstruction Image with ZP
y(mm)
x(mm)

First 1/4 projection Image



FBP Reconstruction Image




Sinogram
angular(radius)
R(mm)
0
0.5
1
1.5
2
2.5
3
-80
-60
-40
-20
0
20
40
60
80