HW 4: Medical Imaging Systems

Jake Bergquist, u6010393

December 4, 2019

Q1

a: Attached is the drawn sinogram for (a). All of the profiles are either semicircular or semi oval with a smooth transition between the different angles.

b: Attached is the drawn sinogram for (b). All profiles have either a semicircle, or two, or some overlap of two. The 0 degree projection has double the peak intensity of any of the projections where the two semicircles do not overlap. One of the circles is closer to the center than the other leading to the kind of mirroring effect seen in the sinogram profile.

c: Attached is the sinogram for (c). All profiles are either trapezoids, rectangles or triangles. Assuming a uniform diffusion coefficient the max intensity of the 45 degree projections (ones that cross the diagonal of the square perfectly) is $\sqrt{2}$ larger than the max intensity of the 0 or 90 degree projections (the ones that are parallel to the sides of the square). Here max intensity meaning the maximum projection on the sinogram which corresponds to the maximal attenuation of the X ray beam through the sample.

Q2

a: The predicted object for part (a) is attached. The pattern of a small rectangle profile with high intensity at 0 and a large rectangular profile with less intensity at 90 or -90 suggests an object like a rectangle that is wider on the 90 degree projections than the 0 degree. Additionally there is a triagle profile at some angle less than 45 degrees which is the angle at which the projection would pass directly along the diagonal of the rectangle.

b: The drawn object prediction is attached. For this one there seems to be a circle of lower attenuation coefficient centered at the origin . The remaining profile looks like an ofset oval like shape along the y axis as shown.

Q3

a: Below is my function. Figure 1 shows the results of 0, 30 and 45 degree projections of a rectangualr profile.

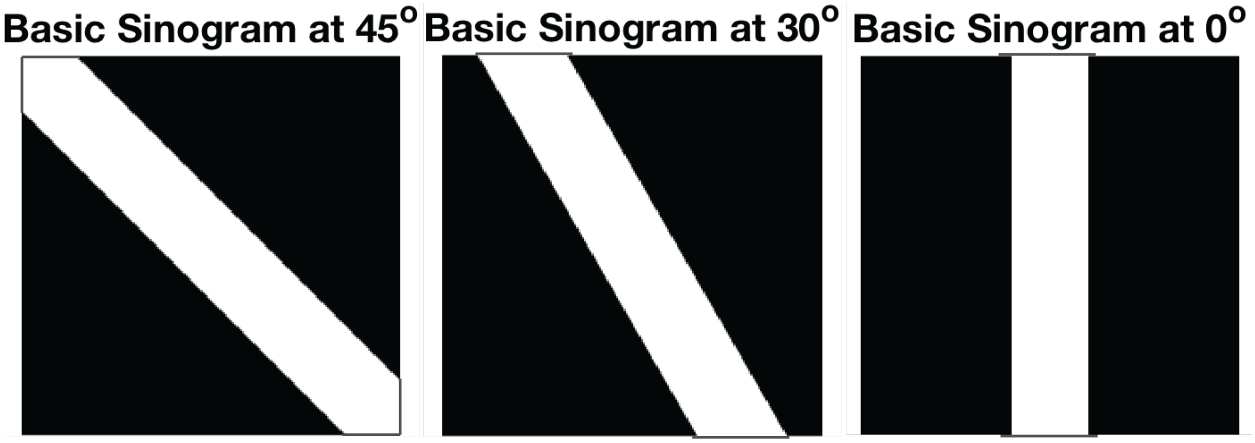
Basic Sinogram at 45^o Basic Sinogram at 30^o Basic Sinogram at 0^o

Figure 1: Basic sinogram projections for 0, 30, and 45 degrees for a rectangular profile.

```
function [Btheta] = BackPropSinogram(Rtheta,theta,interp)
%Input: Rtheta: the inogram projection at theta
%       Theta: the angle of the current projection
%       Interp: The interpolation method fromt eh fit function
%               Default: 'linearinterp'
%Assume image center is at the origin
%Assume Rtheta has an odd number of measurements

if ~exist(interp)
        interp = 'linearinter';
end


imageLim = floor(length(Rtheta)/2);
[X,Y] = meshgrid([-imageLim:imageLim],[imageLim:-1:-imageLim]);
RotCoords1 = Y.*sin(theta) + X.*cos(theta);
RotCoords2 = Y.*sin(pi/2-theta) + X.*cos(pi/2-theta);
Val = repmat(Rtheta,1,size(X,2));
sf = fit([X(:),Y(:)],Val(:),interp);
Btheta = sf(RotCoords2,RotCoords1);
Btheta(isnan(Btheta)) = 0;

%Below is the minimum line version of above (4 lines total)
%I chose to go with what is above because it is easier to read
%Both version produce the same output
%    [X,Y] = meshgrid([-floor(length(Rtheta)/2):floor(length(
   Rtheta)/2)],[floor(length(Rtheta)/2):-1:-floor(length(Rtheta
   )/2)]);
%    sf = fit([X(:),Y(:)],reshape(repmat(Rtheta,1,size(X,2)),[
   length(Rtheta)*size(X,2),1]),interp);
%    Btheta = sf(Y.*sin(pi/2-theta) + X.*cos(pi/2-theta),Y.*sin
   (theta) + X.*cos(theta));
%    Btheta(isnan(Btheta)) = 0;
end
```

b: Figure 2 shows the provided sinogram with 0, 90, and 180 degrees marked.

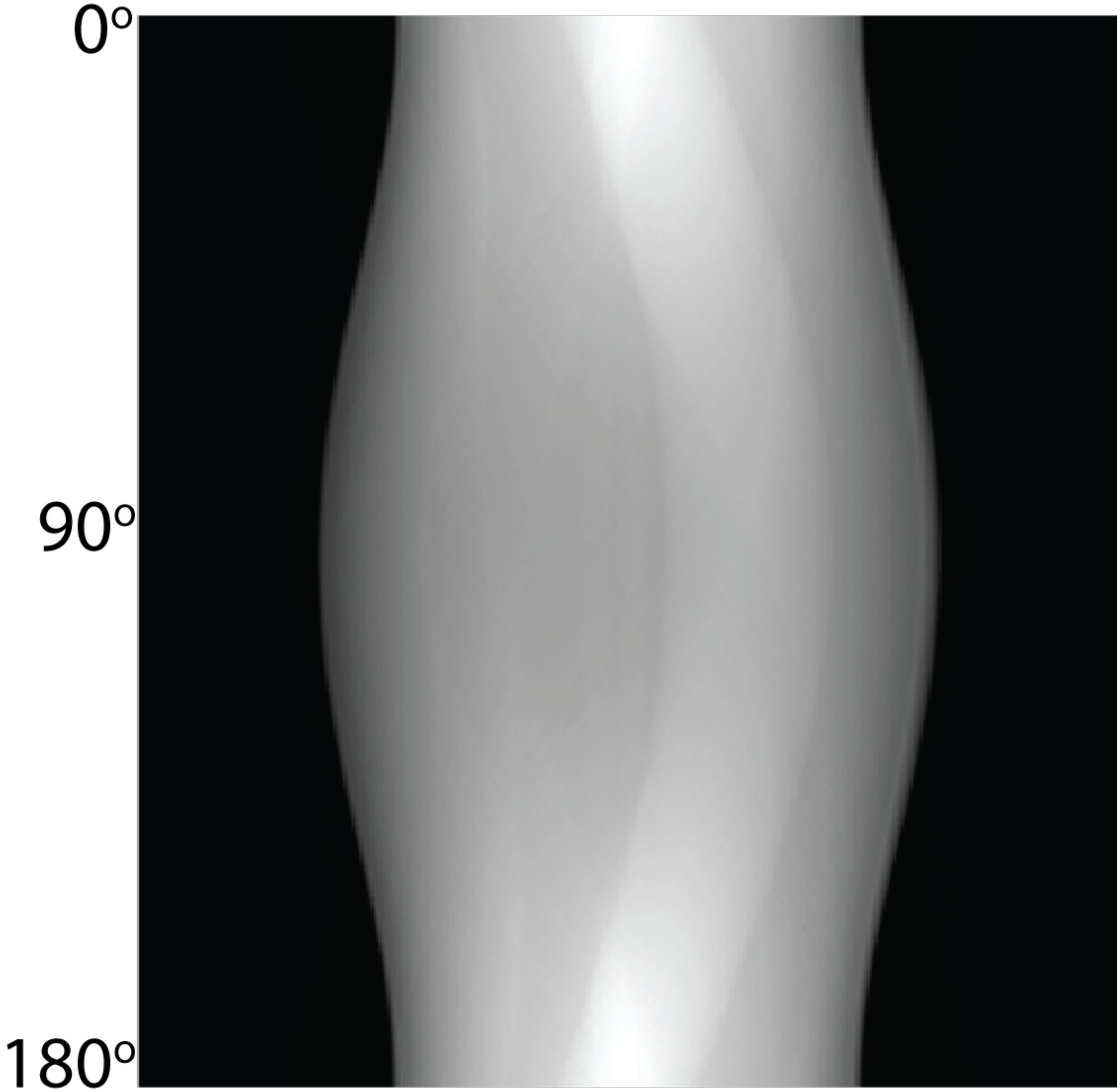

Figure 2: Provided sinogram with 0, 90, and 180 degrees marked.

c&d: In Figure 3 we see the results of backprojection. This is a very blurred image but we can barely make out that there seems to be some circular shape with perhaps another circle of different consistency within it. The Rho filtered results however are much clearer (Figure 4), where several detailed structures can be resolved. There are many more details such as different attenuation coefficients of material at teh border, several regions of large lower and higher attenuation and even some very small circles near the front that were completely unrecognizable in the direct backprojection results. The filter used is shown in Figure 5 where the center point is the each of the sinogram profiles.

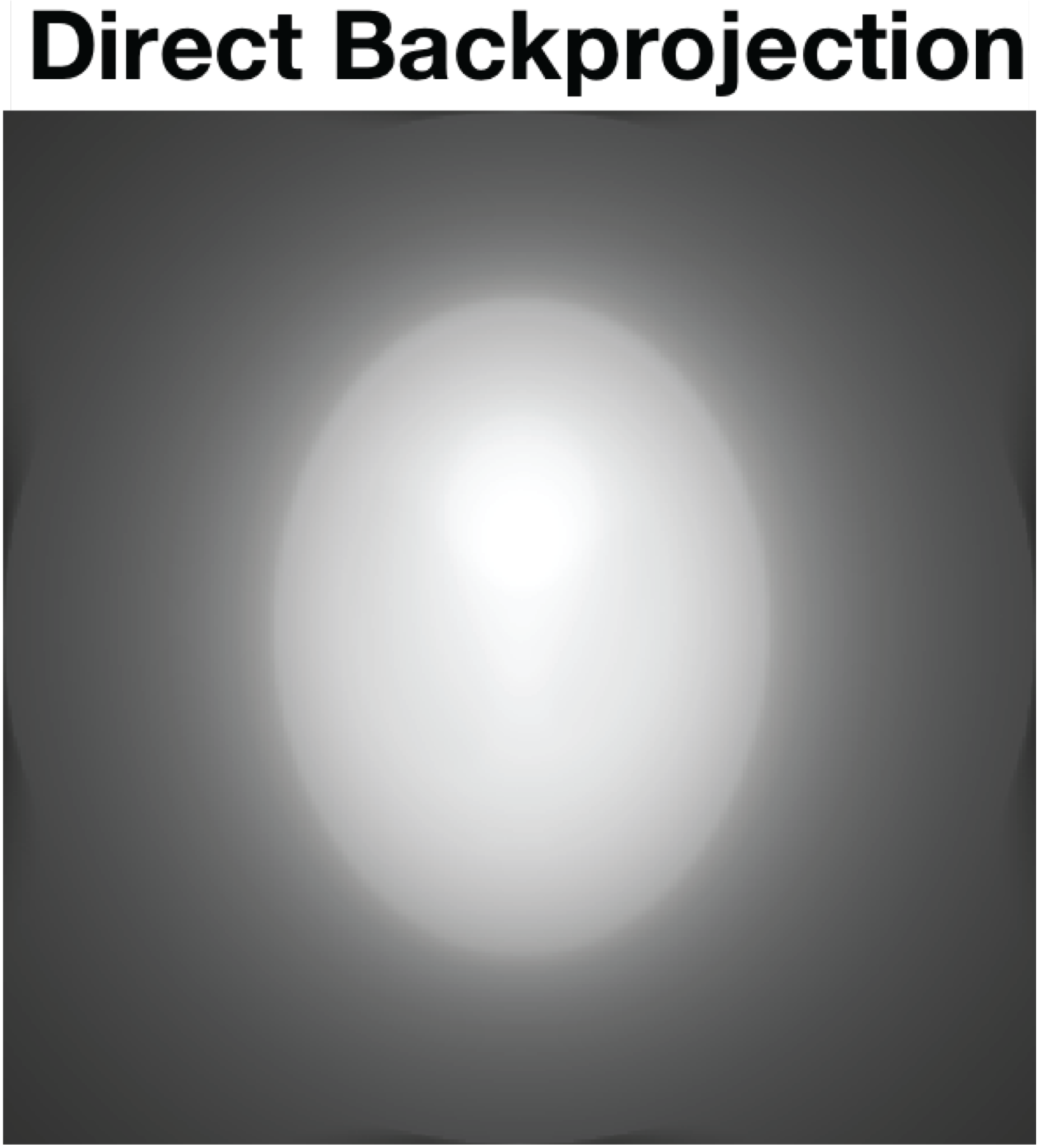

Figure 3: Reconstructed image using direct backprojection.

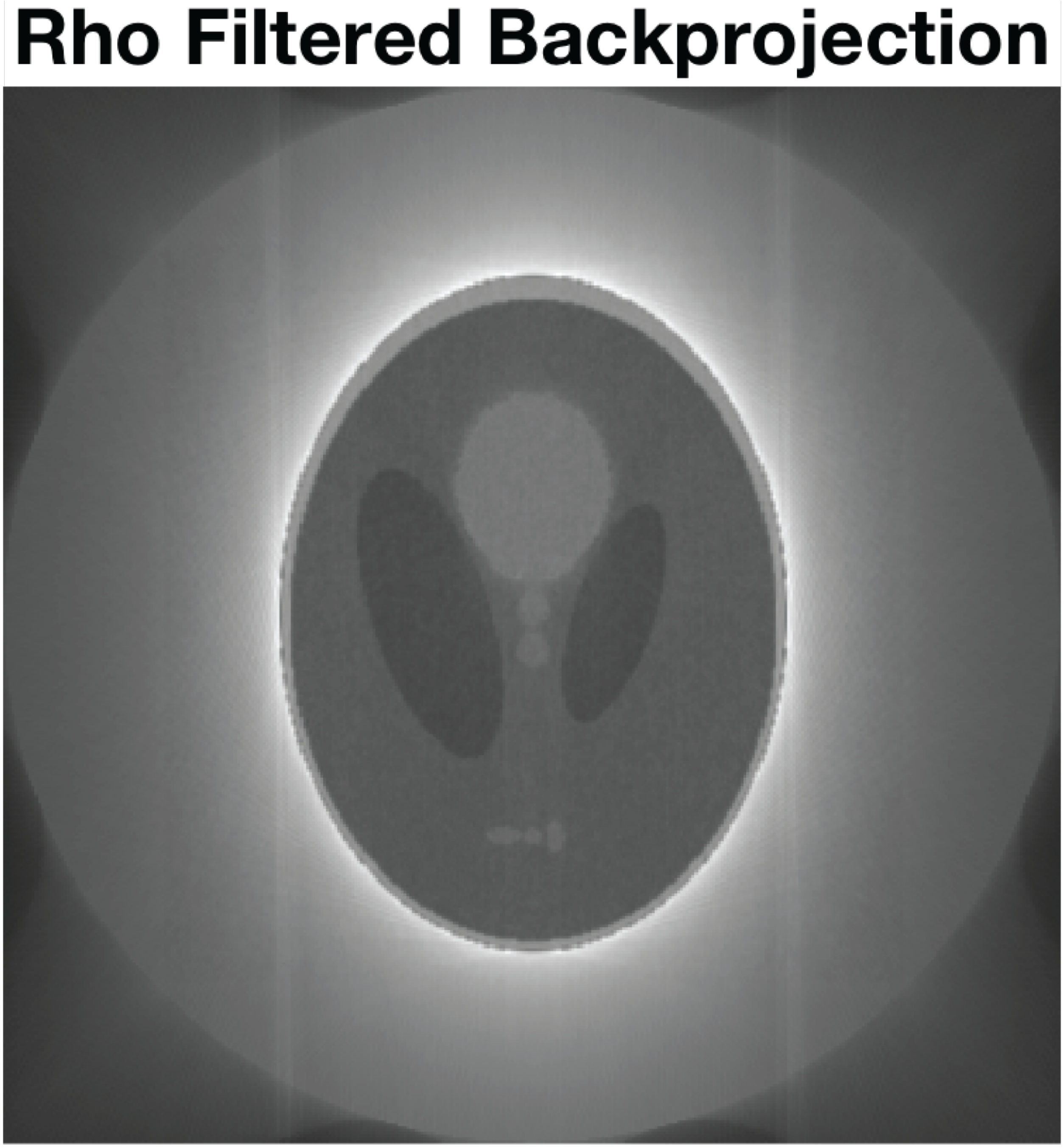

Figure 4: Reconstructed image using Rho filtered backprojection.

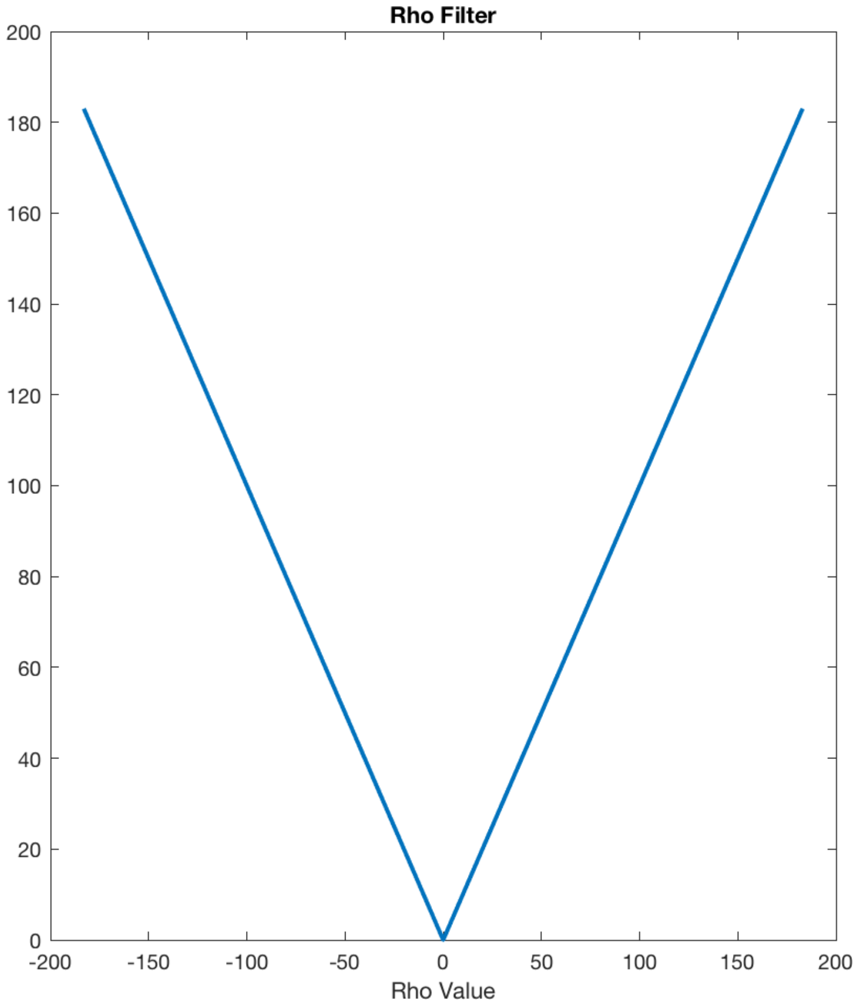

Figure 5: Filter used for Rho filtering.

Q4

The results of downsampling the number of projections (thetas) is shown in Figure 6. With this method of downsampling the number of pixels is not changed because the size of rho, the detector, is not affected. Thus the main artifact of this downsampling is projection artifact from fewer and fewer projections. This can be seen in a sort of radial banding. More projections smooths out this effect.

By contrast the downsamlping of rho, the "number of detectors", causes the image to shrink and become more pixelated. This reduces the ability to resolve some of the smaller, closer together objects.

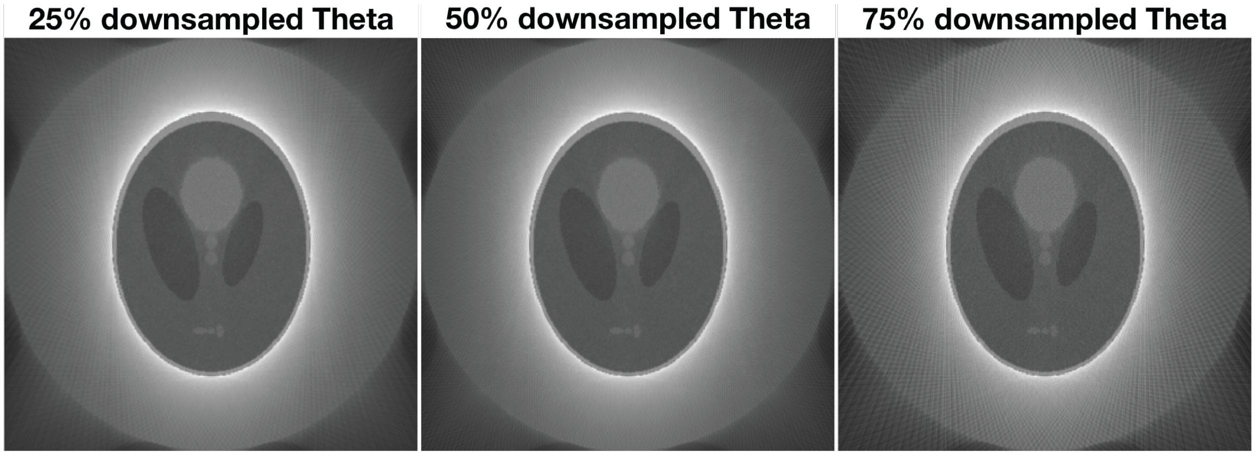

Figure 6: Reconstructions using progressively more downsampled theta.

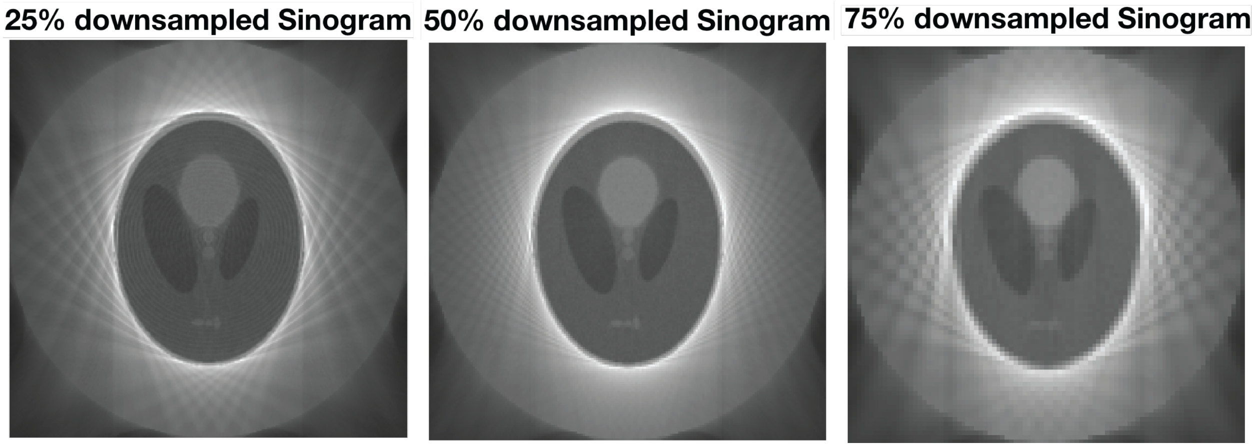

Figure 7: Reconstructions using progressively more downsampled rho.

Q5

Using the given values for water and air which should be at 0 and -1000 respctivly we can construct a mapping from emasured to tru Hounsfield number. In the form $T = am + b$ is a linear equation where a is the "slope" and b is the "y" intercept (here the Y axis is true values). Thus:
$a = \frac{W_t - A_t}{W_o - A_o} = \frac{0 - -1000}{-20 - -1200} = \frac{1000}{1820} \approxeq 0.85$
$b = T - am = 0 - a * -20 = -16.95$

Now using these values we can solve for the true Hounsfield number for an observed value of 1850:
$T = am + b = 0.85 * 1850 - 16.95 = 1585$

Appendix:

Below is the matlab code used for this assignment to generate figures. Figures were post-processed and arranged in illustrator.

```
%%
clear
close all
clc


%%
%Testing Backprojection

testThetas = deg2rad([0:5:175]);
simpleSinogram = zeros(201,1);
simpleSinogram(81:121,:) = 1;

t = 1;
figure(1);imshow(BackPropSinogram(simpleSinogram,testThetas(t),
   'linearinterp'));
title(sprintf('Basic Sinogram at %g^o',rad2deg(testThetas(t))))
   ;
set(gca,'fontsize',20)
t = 7;
figure(2);imshow(BackPropSinogram(simpleSinogram,testThetas(t),
   'linearinterp'));
title(sprintf('Basic Sinogram at %g^o',rad2deg(testThetas(t))))
   ;
set(gca,'fontsize',20)
t = 10;
figure(3);imshow(BackPropSinogram(simpleSinogram,testThetas(t),
   'linearinterp'));
title(sprintf('Basic Sinogram at %g^o',rad2deg(testThetas(t))))
   ;
set(gca,'fontsize',20)
[X,Y] = meshgrid(1:201,1:201);
b = zeros(length(simpleSinogram));
for t = 1:length(testThetas)
b = b + BackPropSinogram(simpleSinogram,testThetas(t),'
   linearinterp');
figure(4);imagesc(b./max(b(:)));title(sprintf('Proj %d',t));
   drawnow();
figure(5);surf(X,Y,b);zlim([0 40]);drawnow();
end



%%
b = zeros(length(simpleSinogram));
```

```
filter = abs([-floor(length(simpleSinogram)/2):floor(length(
   simpleSinogram)/2)])';
filteredSinogram = abs(ifft(fftshift(fft(simpleSinogram)).*
   filter));
for t = 1:length(testThetas)
b = b + BackPropSinogram(filteredSinogram,testThetas(t),'
   linearinterp');
figure(4);imagesc(b./max(b(:)));title(sprintf('Proj %d',t));
   drawnow();
figure(5);surf(X,Y,b);drawnow();
end

%%
figure(1);clf();
imshow(sinogram'/max(sinogram(:)))
%%
%Direct Backprojection
b = zeros(size(sinogram,1));
for t = 1:length(thetas)
b = b+BackPropSinogram(sinogram(:,t),deg2rad(thetas(t)),'
   linearinterp');
figure(1);imshow(b/max(b(:)));title(sprintf('Proj %d',t));
   drawnow();
end
figure(1);title('Direct Backprojection');set(gca,'fontsize',25)
%%
%Rho filtering
b = zeros(size(sinogram,1));
numSamples = size(sinogram,1);
rho = abs([-floor(numSamples/2):floor(numSamples/2)])';
Ft = fftshift(fft(sinogram),1);
M = abs(ifft(Ft.*rho));
%%
for t = 1:length(thetas)
b = b+BackPropSinogram(M(:,t),deg2rad(thetas(t)),'linearinterp'
   );
figure(1);imshow(b/max(b(:)));title(sprintf('Proj %d',t));
   drawnow();
end
figure(1);title('Rho Filtered Backprojection');set(gca,'
   fontsize',25)
%%
%downsampling
%25% downsampled, skip every 4th angle
skipInds = [1:4:length(thetas)];
thetaInds = setdiff([1:length(thetas)],skipInds);
downSampleTheta = thetas(thetaInds);

b = zeros(size(sinogram,1));
```

```
for t = 1:length(downSampleTheta)
b = b+BackPropSinogram(M(:,thetaInds(t)),deg2rad(
   downSampleTheta(t)),'linearinterp');
figure(1);imshow(b/max(b(:)));title(sprintf('Proj %d',t));
   drawnow();
end
figure(1);title('25% downsampled Theta');set(gca,'fontsize',25)

%50% downsampeld uses every other
downSampleTheta = thetas(1:2:end);
thetaInds = [1:2:length(thetas)];
b = zeros(size(sinogram,1));
for t = 1:length(downSampleTheta)
b = b+BackPropSinogram(M(:,thetaInds(t)),deg2rad(
   downSampleTheta(t)),'linearinterp');
figure(2);imshow(b/max(b(:)));title(sprintf('Proj %d',t));
   drawnow();
end
figure(2);title('50% downsampled Theta');set(gca,'fontsize',25)
%75% downsampled uses every 4th

downSampleTheta = thetas(1:4:end);
thetaInds = [1:4:length(thetas)];
b = zeros(size(sinogram,1));
for t = 1:length(downSampleTheta)
b = b+BackPropSinogram(M(:,thetaInds(t)),deg2rad(
   downSampleTheta(t)),'linearinterp');
figure(3);imshow(b/max(b(:)));title(sprintf('Proj %d',t));
   drawnow();
end
figure(3);title('75% downsampled Theta');set(gca,'fontsize',25)
%%
%Downsample sinogram
%25%
skipInds = [1:4:size(sinogram,1)];
sinogramInds = setdiff([1:size(sinogram,1)],skipInds);
downsampleSinogram = sinogram(sinogramInds,:);
downsampleSinogram = abs(ifft((fftshift(fft(downsampleSinogram)
   ,1).*abs([-floor(size(downsampleSinogram,1)/2):floor(size(
   downsampleSinogram,1)/2)]')))));
b = zeros(size(downsampleSinogram,1));
for t = 1:length(thetas)
b = b+BackPropSinogram(downsampleSinogram(:,t),deg2rad(thetas(t
   )),'linearinterp');
figure(1);imshow(b/max(b(:)));title(sprintf('Proj %d',t));
   drawnow();
end
figure(1);title('25% downsampled Sinogram');set(gca,'fontsize'
   ,25)
```

```
%50%
sinogramInds = [1:2:size(sinogram,1)-1];
downsampleSinogram = sinogram(sinogramInds,:);
downsampleSinogram = abs(ifft((fftshift(fft(downsampleSinogram)
   ,1).*abs([-floor(size(downsampleSinogram,1)/2):floor(size(
   downsampleSinogram,1)/2)]')))));
b = zeros(size(downsampleSinogram,1));
for t = 1:length(thetas)
b = b+BackPropSinogram(downsampleSinogram(:,t),deg2rad(thetas(t
   )),'linearinterp');
figure(2);imshow(b/max(b(:)));title(sprintf('Proj %d',t));
   drawnow();
end
figure(2);title('50% downsampled Sinogram');set(gca,'fontsize'
   ,25)


%75%
sinogramInds = [1:4:size(sinogram,1)-3];
downsampleSinogram = sinogram(sinogramInds,:);
downsampleSinogram = abs(ifft((fftshift(fft(downsampleSinogram)
   ,1).*abs([-floor(size(downsampleSinogram,1)/2):floor(size(
   downsampleSinogram,1)/2)]')))));
b = zeros(size(downsampleSinogram,1));
for t = 1:length(thetas)
b = b+BackPropSinogram(downsampleSinogram(:,t),deg2rad(thetas(t
   )),'linearinterp');
figure(3);imshow(b/max(b(:)));title(sprintf('Proj %d',t));
   drawnow();
end
figure(3);title('75% downsampled Sinogram');set(gca,'fontsize'
   ,25)
%%
%Hounsfield number fitting
water = [-20,0];
air = [-1200,-1000];

a = (water(2)-air(2))/(water(1)-air(1));
b = water(2)-a*water(1);

MysteryItem(1)=1850;
MysteryItem(2) = a*MysteryItem(1) + b;

figure(1);clf();
hold on;
scatter([water(1),air(1),MysteryItem(1)],[water(2),air(2),
   MysteryItem(2)]);
plot([-1300:1900],a.*[-1300:1900]+b);
```