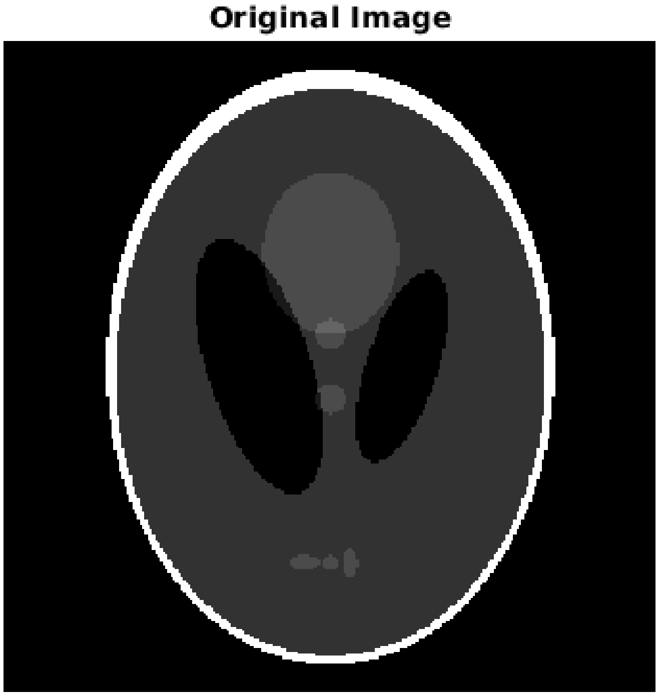
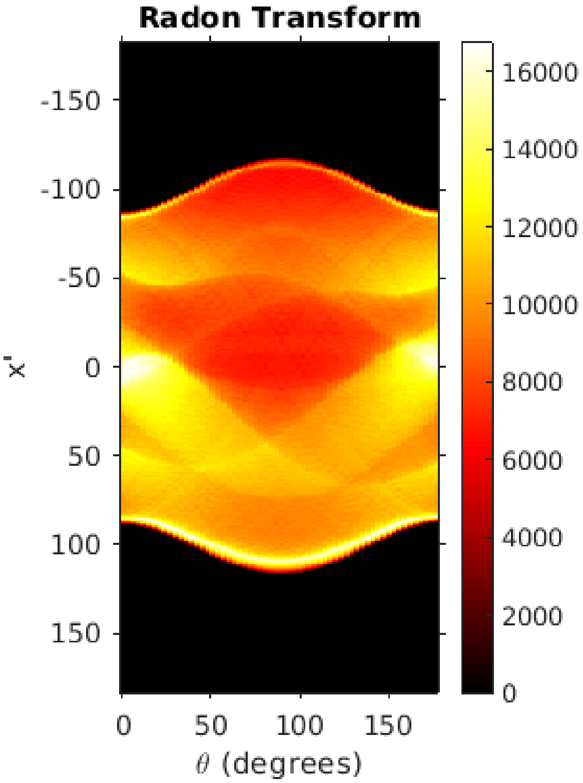
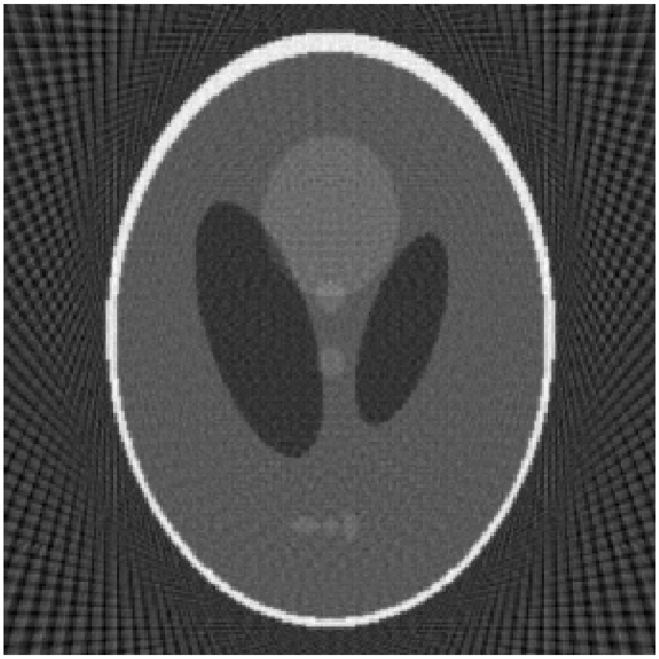
## Table of Contents

# Inbuilt filtered backprojection

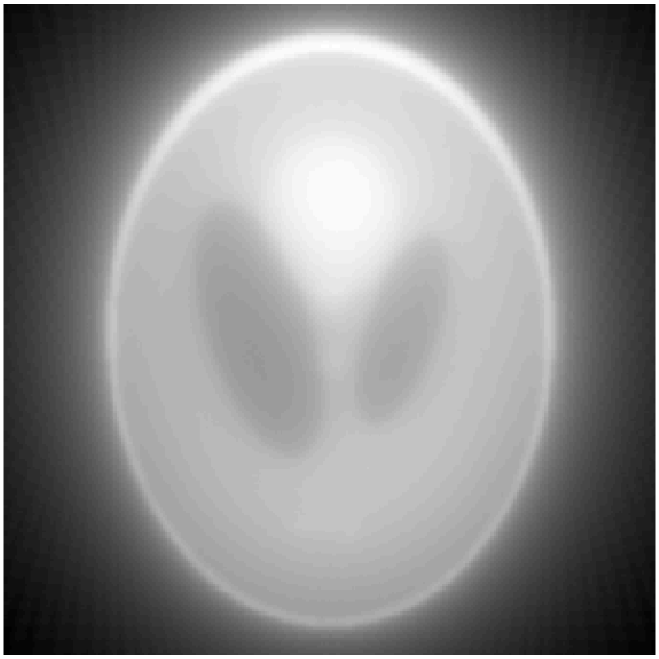
```
clear;

A = double(imread("SheppLogan256.png"));
B = phantom(128);
theta=[0:3:179];

f1=figure(1);
imshow(A,[]);
title('Original Image');

[R,xp]=radon(A,theta);
I1=iradon(R,theta,'linear', 256);
I2=iradon(R,theta,'linear','none', 256);

iptsetpref('ImshowAxesVisible','on');
f2=figure(2);
imshow(R,[],'Xdata',theta,'Ydata',xp,'InitialMagnification','fit');
xlabel('\theta (degrees)');
ylabel('x''');
title('Radon Transform');
colormap(gca,hot), colorbar;

iptsetpref('ImshowAxesVisible','off');
f3=figure(3);
imshow(I1,[]);
title('Filtered Backprojection (in-built)');

f4=figure(4);
imshow(I2,[]);
title('Unfiltered Backprojection (in-built)');
```


Original Image


Radon Transform
-150
-100
-50
0
50
100
150
x'
0
50
100
150
$\theta$ (degrees)
16000
14000
12000
10000
8000
6000
4000
2000
0



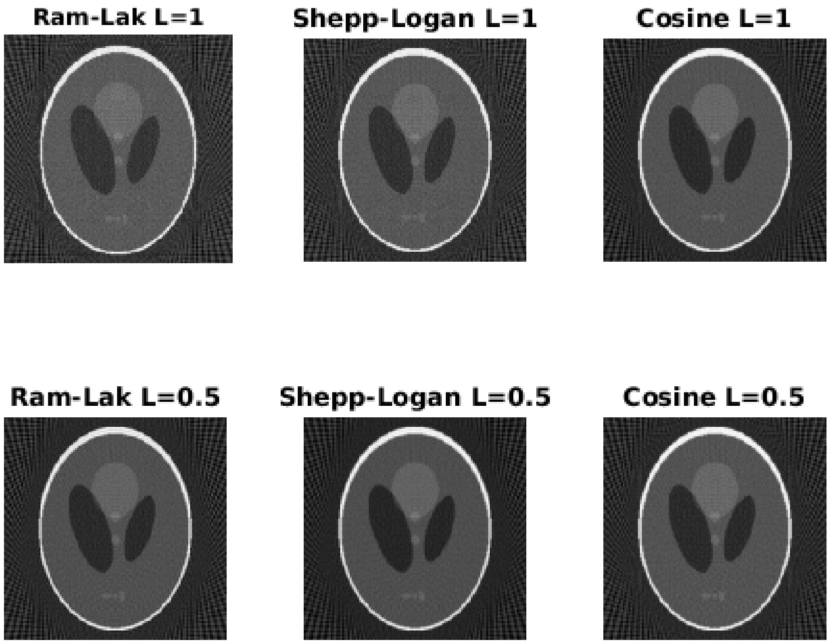
Filtered Backprojection (in-built)



Unfiltered Backprojection (in-built)

# myFilter images

```
% Ram-Lak
R_filt = myFilter(R,theta,xp,1,1);
bkproj_ramlak_1 = iradon(R_filt,theta,'linear','none',1,256);
R_filt = myFilter(R,theta,xp,1,0.5);
bkproj_ramlak_2 = iradon(R_filt,theta,'linear','none',1,256);

% Shepp-Logan
R_filt = myFilter(R,theta,xp,2,1);
bkproj_shepp_1 = iradon(R_filt,theta,'linear','none',1,256);
R_filt = myFilter(R,theta,xp,2,0.5);
bkproj_shepp_2 = iradon(R_filt,theta,'linear','none',1,256);

% Cosine
R_filt = myFilter(R,theta,xp,3,1);
bkproj_cos_1 = iradon(R_filt,theta,'linear','none',1,256);
R_filt = myFilter(R,theta,xp,3,0.5);
bkproj_cos_2 = iradon(R_filt,theta,'linear','none',1,256);

f5=figure(5);
subplot(2,3,1)
imshow(bkproj_ramlak_1, []);
title('Ram-Lak L=1');
% colorbar;

subplot(2,3,4)
imshow(bkproj_ramlak_2, []);
title('Ram-Lak L=0.5');
% colorbar;

subplot(2,3,2)
imshow(bkproj_shepp_1, []);
title('Shepp-Logan L=1');
% colorbar;

subplot(2,3,5)
imshow(bkproj_shepp_2, []);
title('Shepp-Logan L=0.5');
% colorbar;

subplot(2,3,3)
imshow(bkproj_cos_1, []);
title('Cosine L=1');
% colorbar;

subplot(2,3,6)
imshow(bkproj_cos_1, []);
title('Cosine L=0.5');
% colorbar;
```



```
disp(["The cosine filter gives the best results in this case, among
 the three."]);
disp(["Across all filters, the images at L=0.5 seem to be sharper with
 better edges than at L=1"]);
```

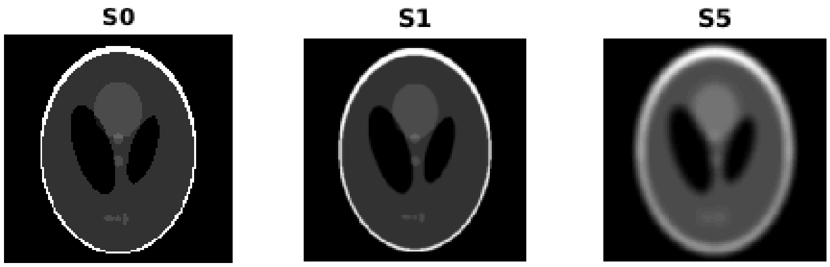
```
The cosine filter gives the best results in this case, among the
 three.
Across all filters, the images at L=0.5 seem to be sharper with better
 edges than at L=1
```

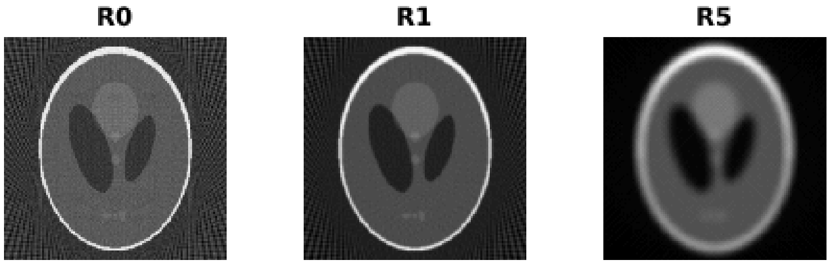
# Part B

```
S0 = A;
S1 = imgaussfilt(A, 1);
S5 = imgaussfilt(A, 5);

[RR0,xp0]=radon(S0,theta);
[RR1,xp1]=radon(S1,theta);
[RR5,xp5]=radon(S5,theta);

RRfilt0 = myFilter(RR0,theta,xp0,1,1);
RRfilt1 = myFilter(RR1,theta,xp1,1,1);
RRfilt5 = myFilter(RR5,theta,xp5,1,1);

R0 = iradon(RRfilt0,theta,'linear','none',1,256);
R1 = iradon(RRfilt1,theta,'linear','none',1,256);
R5 = iradon(RRfilt5,theta,'linear','none',1,256);
```

```
RRMSE0 = RRMSE(S0, R0);
RRMSE1 = RRMSE(S1, R1);
RRMSE5 = RRMSE(S5, R5);

disp(["using myFilter for backprojection:"]);
disp(['RRMSE for S0: ', num2str(RRMSE0)]);
disp(['RRMSE for S1: ', num2str(RRMSE1)]);
disp(['RRMSE for S5: ', num2str(RRMSE5)]);

f6=figure(6);
subplot(2,3,1);
imshow(S0,[]);
title('S0');
% colorbar;

subplot(2,3,2);
imshow(S1,[]);
title('S1');
% colorbar;

subplot(2,3,3);
imshow(S5,[]);
title('S5');
% colorbar;

subplot(2,3,4);
imshow(R0,[]);
title('R0');
% colorbar;

subplot(2,3,5);
imshow(R1,[]);
title('R1');
% colorbar;

subplot(2,3,6);
imshow(R5,[]);
title('R5');
% colorbar;
```

```
using myFilter for backprojection:
RRMSE for S0: 0.32697
RRMSE for S1: 0.20792
RRMSE for S5: 0.20129
```





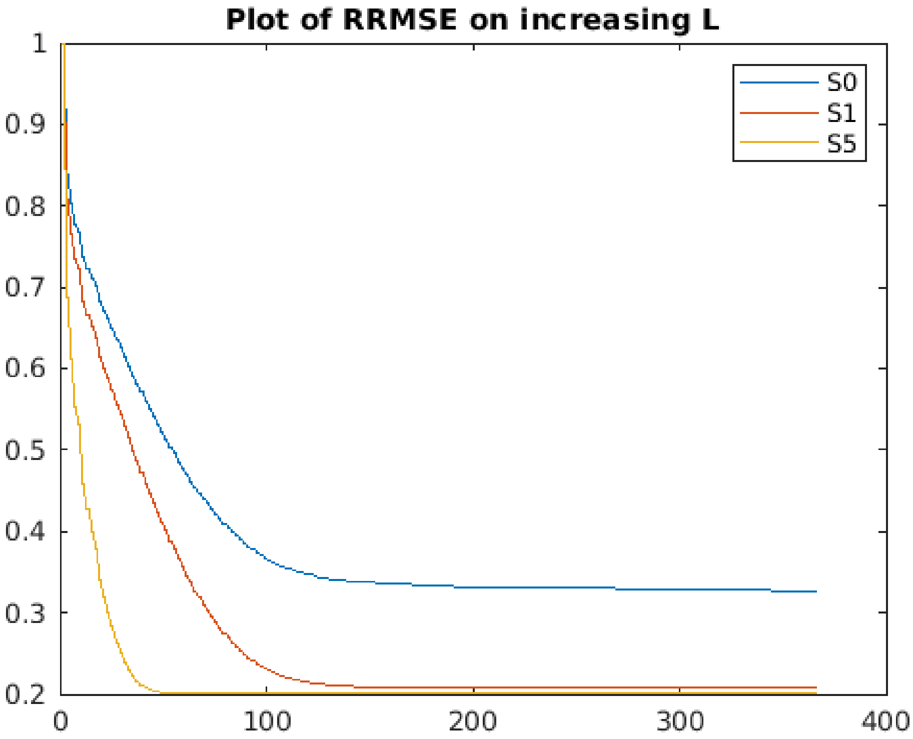
```
disp(["The most blurred image, that is S5 has the least RRMSE, while
 S0 has the highest."]);
disp(["WE observe that RRMSE is decreasing on increasing the Gaussian
 blur."]);
disp(["Blurring results in attenuating high frequency components in
 the image."]);
disp(["Since the most blurred image will have the most attenuated high
 frequency components, the reconstruction error is the least for it
 after filtered backprojection"]);
disp(["Hence the observation"]);
```

```
The most blurred image, that is S5 has the least RRMSE, while S0 has
 the highest.
WE observe that RRMSE is decreasing on increasing the Gaussian blur.
Blurring results in attenuating high frequency components in the
 image.
Since the most blurred image will have the most attenuated high
 frequency components, the reconstruction error is the least for it
 after filtered backprojection
Hence the observation
```

# Part C

```
N=length(xp);
L_arr = linspace((1/N), 1, N);

for i=1:N
```

```
 RRfilt0 = myFilter(RR0,theta,xp0,1,L_arr(i));
 RRfilt1 = myFilter(RR1,theta,xp1,1,L_arr(i));
 RRfilt5 = myFilter(RR5,theta,xp5,1,L_arr(i));
 R0 = iradon(RRfilt0,theta,'linear','none',1,256);
 R1 = iradon(RRfilt1,theta,'linear','none',1,256);
 R5 = iradon(RRfilt5,theta,'linear','none',1,256);
 RRMSE_S0(i) = RRMSE(S0,R0);
 RRMSE_S1(i) = RRMSE(S1,R1);
 RRMSE_S5(i) = RRMSE(S5,R5);
end

f7=figure(7);
plot(RRMSE_S0);
hold on;
plot(RRMSE_S1);
hold on;
plot(RRMSE_S5);
legend('S0','S1','S5');
title('Plot of RRMSE on increasing L');
hold off;
```



```
disp(["Again as the increase in blurring corresponds to attenuation
 of higher frequency components, reconstruction of S5 is more steeply
 dependent on lower L values"]);
disp(["Its RRMSE thus decreases strongly for lower values of L and
 isn't affected further."]);
disp(["We observe that the higher the blur, the more steep is the
 curve."])
```

```
Again as the increase in blurring corresponds to attenuation of higher
 frequency components, reconstruction of S5 is more steeply dependent
 on lower L values
Its RRMSE thus decreases strongly for lower values of L and isn't
 affected further.
We observe that the higher the blur, the more steep is the curve.
```

*Published with MATLAB® R2020a*