Assignment 5 - Signal and Image Processing 2014

Martin Jørgensen, tzk173

October 1, 2014

Contents

1 Task 1 **2**

2 Task 2 **2**

3 Task 3 **3**

4 Task 4 **4**

A Appendix **8**

Code for Task 1 8

`sinogram` 9

Code for Task 2 10

`backprojection` 11

Code for Task 3 12

`fbackprojection` 13

Code for Task 4 14

1 Task 1

The code for creating Figure 1 can be found in the appendix.

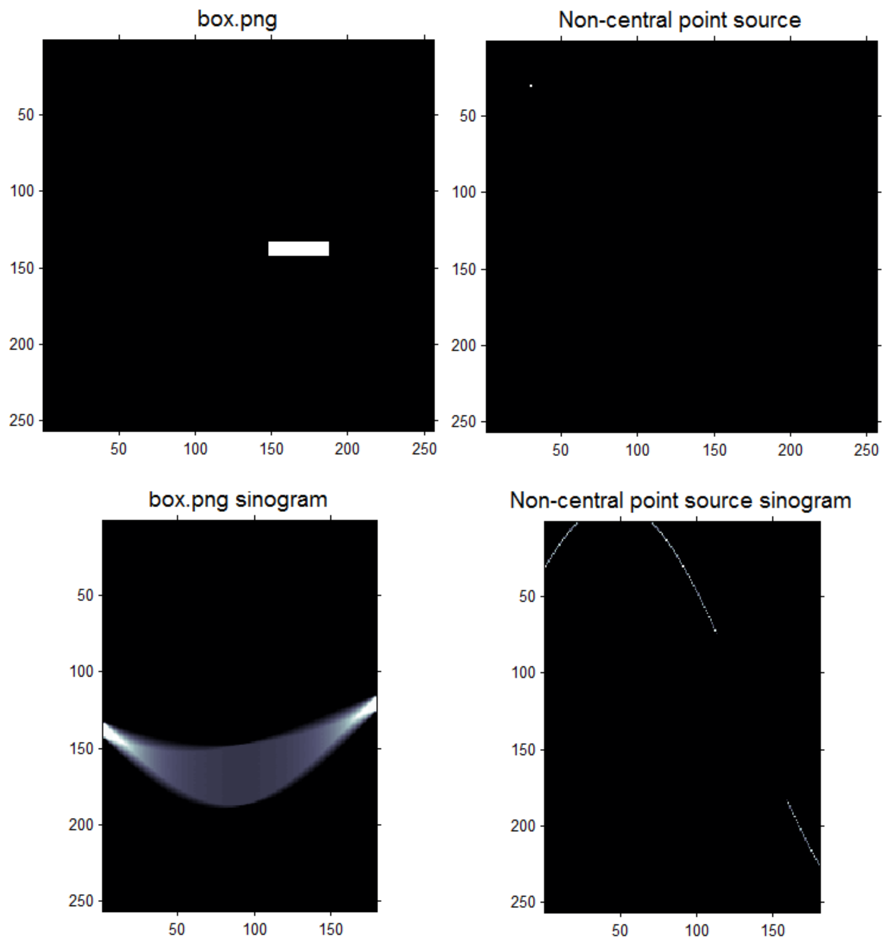

Figure 1: Two different images and their respective sinograms.

The creation of the sinograms is made using the Radon transform

$$p(\xi, \phi) = \int f(x, y)\delta(x\cos\phi + y\sin\phi - \xi)dxdy$$

2 Task 2

The code for creating Figure 2 can be found in the appendix.

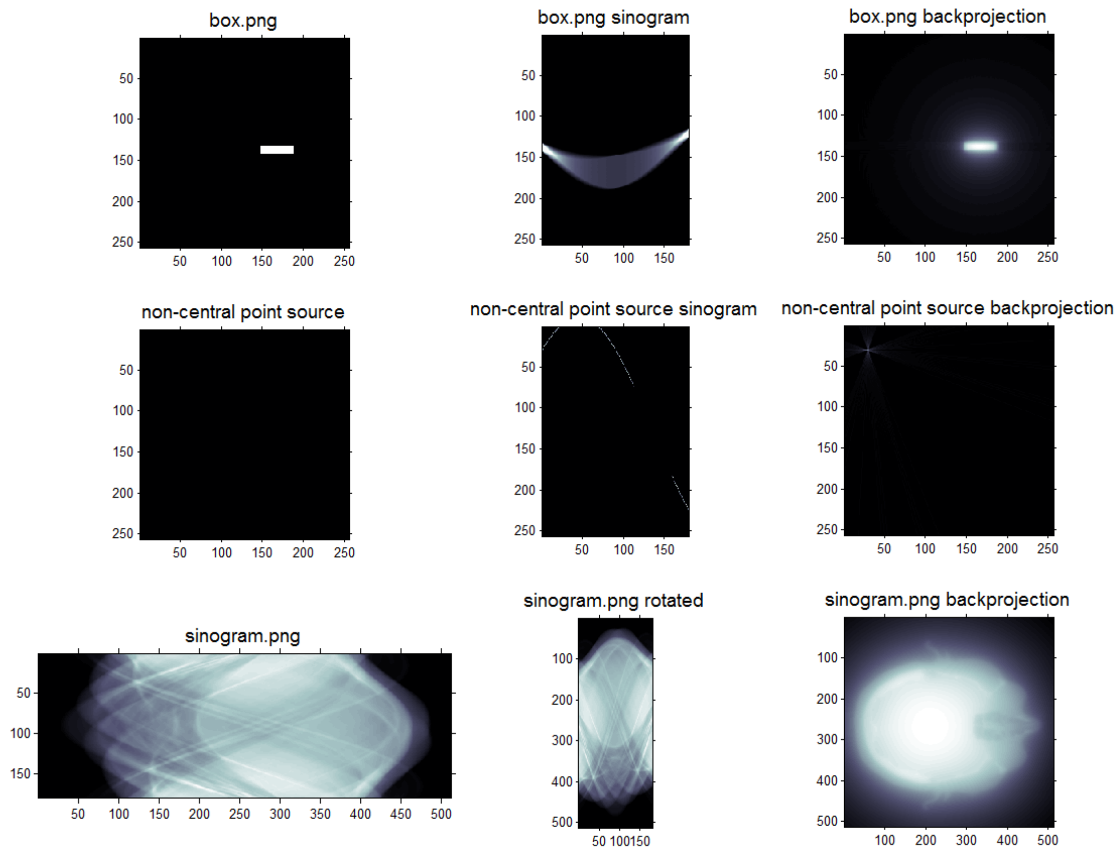

Figure 2: The two upper rows are images, their sinograms and the backprojection of their sinograms. The lower row is a sinogram, the same sinogram rotated and finally the back projection of the rotated sinogram.

The backprojection is written as the formula

$$f_{bp}(x, y) = \int_0^{\pi} p(x \sin \phi + y \cos \phi, \phi) d\phi$$

3 Task 3

The code for creating Figure 3 can be found in the appendix.

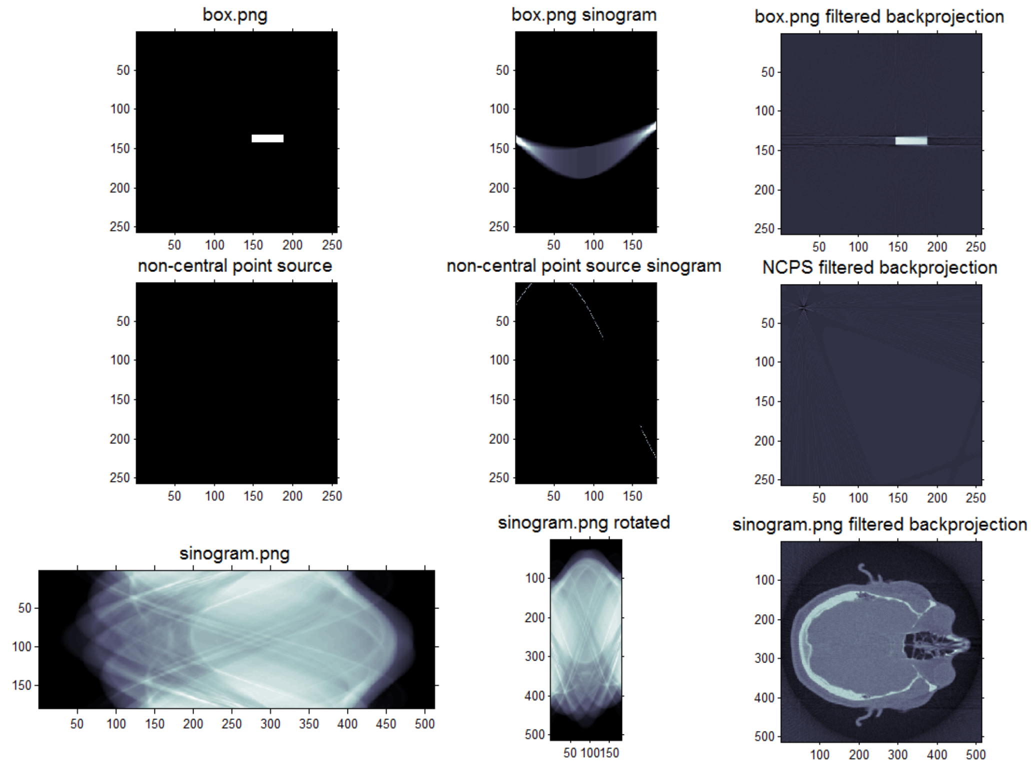

Figure 3: The two upper rows are images, their sinograms and the filtered backprojection of their sinograms. The lower row is a sinogram, the same sinogram rotated and finally the filtered backprojection of the rotated sinogram.

For this task a ramp filter is applied using FFT convolution during the backprojecting function in order to remove some of the "blur" that occurs during the normal backprojection. This decrease in blur comes a a price since some of the original colour get lots and is replaced by radiating lines from the projection angles.

4 Task 4

The code for creating Figure 4 and Figure 5 can be found in the appendix.

Figure 4 and Figure 5 shows the difference the M value have on both the sinogram, and the reconstruction afterwards. For my example I used the *box.png* image as instructed. I chose the *mir.tif* image as a supplement since this technique is primarily used for medicinal imaging, and as such a medicinal image would be a good test case.

We can see that as M increase, the sinograms grow bigger and bigger, but also that the reconstructions gets better and better. At lower M values it's possible to see the different projection angles around the objects.

When the M value crosses a certain threshold we do not gain any more new information for the resulting reconstruction. (I attempted with both $M = 512, M = 1024$, $M = 2048$

and $M = 4096$, there was no visible difference from $M = 256$) This is because the projection angles start to overlap with each other and infer the same information multible times. For a bigger image (higher resolution) more angles will make a bigger difference since there will be further between data points along the edges, but for these sizes it does not matter.

Please note that the code in Figure 12 will only produce on of the figures, to produce the other, you will need to change the value of the `file` variable.

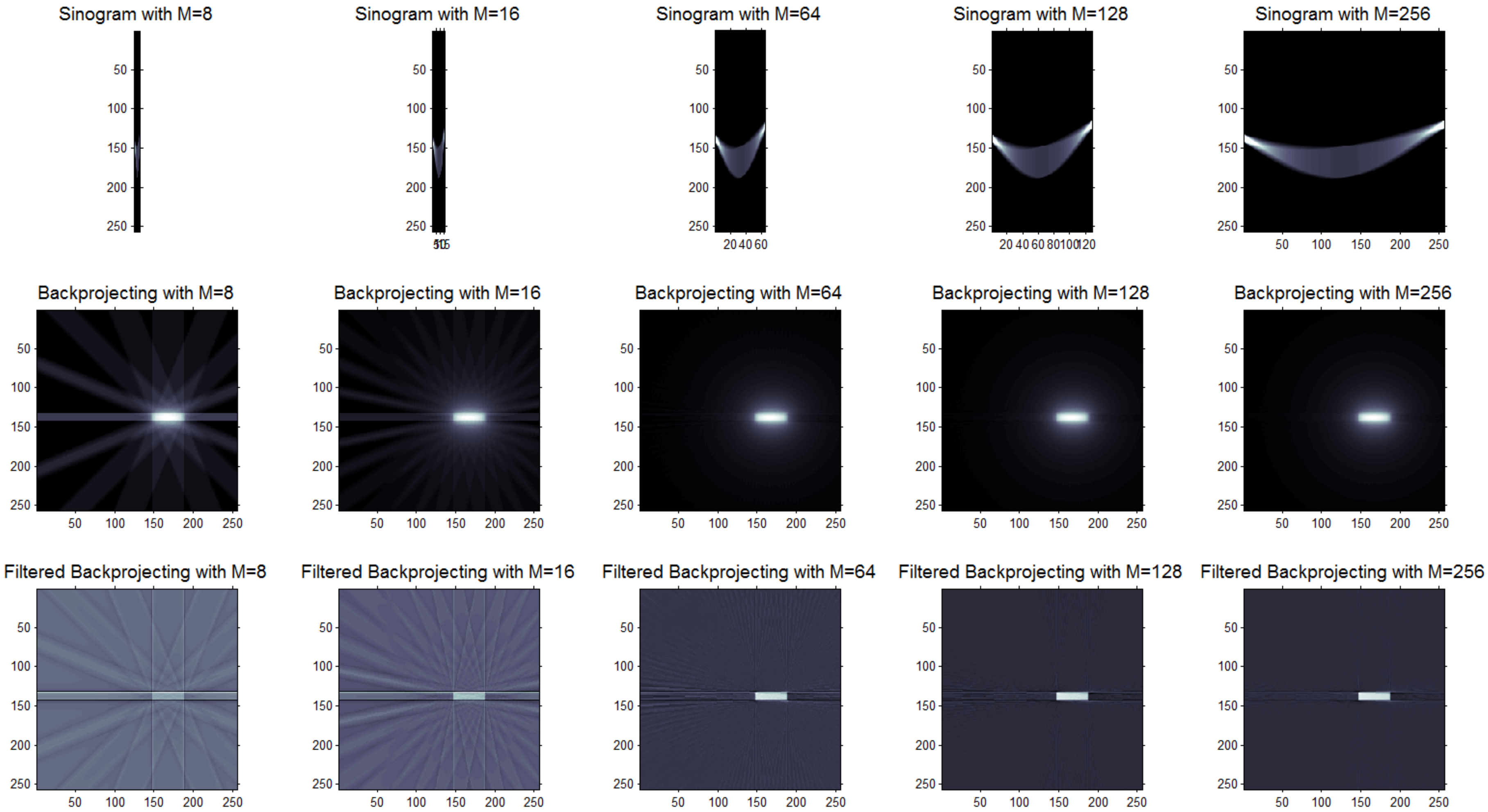

Figure 4: The image *box.png* using different values for M.

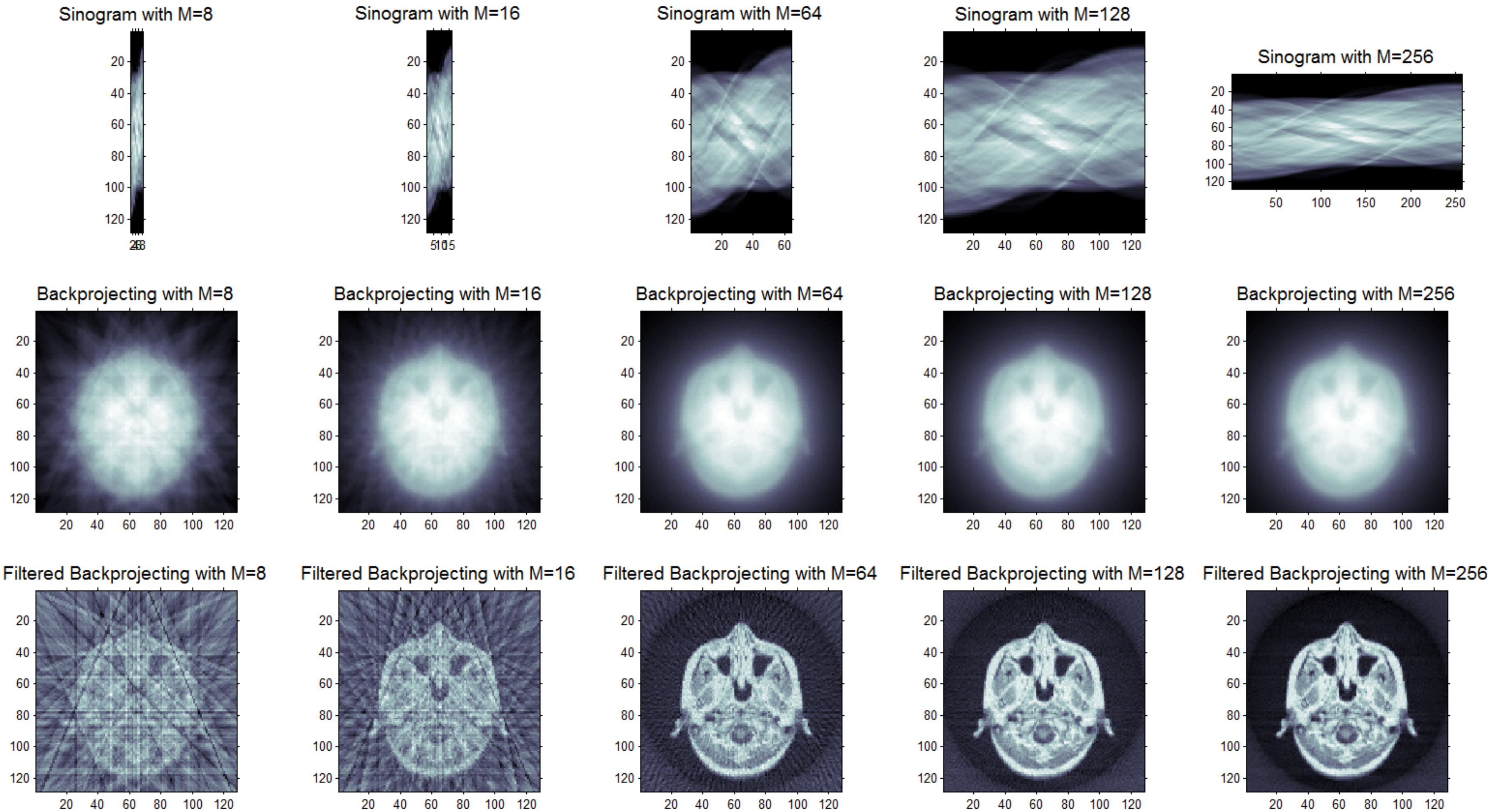

Figure 5: The image *mir.tif* using different values for M.

A Appendix

Code for Task 1

```
% Solution for part 1 of Assignment 5.
% Written by: Martin Jrgensen, tzk173

clear all;

% Read image(s).
I1 = double(imread('box.png'));
I2 = double(zeros(256,256));
I2(30,30) = 255;

I3 = sinogram(I1, 180);
I4 = sinogram(I2, 180);

h = figure(511); set(h,'Color','White'); colormap(bone);
subplot(2,2,1); imagesc(I1); axis image; set(gca,'TickDir','out');
title('box.png','FontSize',14);
subplot(2,2,2); imagesc(I2); axis image; set(gca,'TickDir','out');
title('Non-central point source','FontSize',14);
subplot(2,2,3); imagesc(I3); axis image; set(gca,'TickDir','out');
title('box.png sinogram','FontSize',14);
subplot(2,2,4); imagesc(I4); axis image; set(gca,'TickDir','out');
title('Non-central point source sinogram','FontSize',14);
```

Figure 6: Code for producing the figure for the first task. (../p1.m)

sinogram

```
function [g] = sinogram(f, M)
% sinogram Construct the sinogram of an image "f" using "M" projection lines.

    Phi = zeros(M,1);
    N = size(f,1);
    g = zeros(N,M);

    for i = 1:M
        Phi(i) = 180/M * (i-1);
        tmp = imrotate(f,-Phi(i), 'bilinear', 'crop');
        g(:,i) = sum(tmp,2);
    end
end
```

Figure 7: Code the `sinogram` function. (../sinogram.m)

Code for Task 2

```
% Solution for part 2 of Assignment 5.
% Written by: Martin Jrgensen, tzk173

clear all;

% Read image(s).
I1 = double(imread('box.png'));
I2 = double(zeros(256,256));
I2(30,30) = 255;
I3 = double(imread('sinogram.png'));
I4 = imrotate(I3,90,'bilinear','loose');

% Create sinograms
I1s = sinogram(I1,180);
I2s = sinogram(I2,180);

% Recover that sh...picture!
I1r = backprojection(I1s, 180);
I2r = backprojection(I2s, 180);
I3r = backprojection(I4, 180);

h = figure(521); set(h,'Color','White'); colormap(bone);
% For box.png.
subplot(3,3,1); imagesc(I1); axis image; set(gca,'TickDir','out');
title('box.png','FontSize',14);
subplot(3,3,2); imagesc(I1s); axis image; set(gca,'TickDir','out');
title('box.png sinogram','FontSize',14);
subplot(3,3,3); imagesc(I1r); axis image; set(gca,'TickDir','out');
title('box.png backprojection','FontSize',14);

% For non-central point source.
subplot(3,3,4); imagesc(I2); axis image; set(gca,'TickDir','out');
title('non-central point source','FontSize',14);
subplot(3,3,5); imagesc(I2s); axis image; set(gca,'TickDir','out');
title('non-central point source sinogram','FontSize',14);
subplot(3,3,6); imagesc(I2r); axis image; set(gca,'TickDir','out');
title('non-central point source backprojection','FontSize',14);

% For sinogram.png
subplot(3,3,7); imagesc(I3); axis image; set(gca,'TickDir','out');
title('sinogram.png','FontSize',14);
subplot(3,3,8); imagesc(I4); axis image; set(gca,'TickDir','out');
title('sinogram.png rotated','FontSize',14);
subplot(3,3,9); imagesc(I3r); axis image; set(gca,'TickDir','out');
title('sinogram.png backprojection','FontSize',14);
```

Figure 8: Code for producing the figure for the second task. (../p2.m)

backprojection

```
function [ f ] = backprojection( g, M )
% backprojection takes a sinogram "g" and the number of projection angles
% "M" and recreates the image of the sinogram.

    Phi = zeros(M,1);
    N = size(g,1);
    f = zeros(N,N);
    mid = floor(N/2) + 1;

    [x,y] = meshgrid(ceil(-N/2):ceil(N/2-1));

    for i = 1:M
        Phi(i) = 180/M * (i-1) * (pi/180);

        rCoords = round(mid + x*sin(Phi(i)) + y*cos(Phi(i)));
        idx = find((rCoords > 0) & (rCoords <= N));
        newCoords = rCoords(idx);
        f(idx) = f(idx) + g(newCoords, i) ./ M;
    end

end
```

Figure 9: Code the `backprojection` function. (../backprojection.m)

Code for Task 3

```
% Solution for part 3 of Assignment 5.
% Written by: Martin Jrgensen, tzk173

clear all;

% Read image(s).
I1 = double(imread('box.png'));
I2 = double(zeros(256,256));
I2(30,30) = 255;
I3 = double(imread('sinogram.png'));
I4 = imrotate(I3,90,'bilinear','loose');

% Create sinograms
I1s = sinogram(I1,180);
I2s = sinogram(I2,180);


% Recover that sh...picture!
I1r = fbackprojection(I1s, 180);
I2r = fbackprojection(I2s, 180);
I3r = fbackprojection(I4, 180);



h = figure(531); set(h,'Color','White'); colormap(bone);
% For box.png.
subplot(3,3,1); imagesc(I1); axis image; set(gca,'TickDir','out');
title('box.png','FontSize',14);
subplot(3,3,2); imagesc(I1s); axis image; set(gca,'TickDir','out');
title('box.png sinogram','FontSize',14);
subplot(3,3,3); imagesc(I1r); axis image; set(gca,'TickDir','out');
title('box.png filtered backprojection','FontSize',14);

% For non-central point source.
subplot(3,3,4); imagesc(I2); axis image; set(gca,'TickDir','out');
title('non-central point source','FontSize',14);
subplot(3,3,5); imagesc(I2s); axis image; set(gca,'TickDir','out');
title('non-central point source sinogram','FontSize',14);
subplot(3,3,6); imagesc(I2r); axis image; set(gca,'TickDir','out');
title('NCPS filtered backprojection','FontSize',14);

% For sinogram.png
subplot(3,3,7); imagesc(I3); axis image; set(gca,'TickDir','out');
title('sinogram.png','FontSize',14);
subplot(3,3,8); imagesc(I4); axis image; set(gca,'TickDir','out');
title('sinogram.png rotated','FontSize',14);
subplot(3,3,9); imagesc(I3r); axis image; set(gca,'TickDir','out');
title('sinogram.png filtered backprojection','FontSize',14);
```

Figure 10: Code for producing the figure for the third task. (../p3.m)

fbackprojection

```
function [ f ] = fbackprojection( g, M )
% fbackprojection Performs filetered back projection
% of a sinogram to recreate the image. Uses a ramp filter.
    Phi = zeros(M,1);
    N = size(g,1);
    f = zeros(N,N);
    mid = floor(N/2) + 1;

    [x,y] = meshgrid(ceil(-N/2):ceil(N/2-1));
    h = [floor(N/2):-1:0 1:ceil(N/2-1)]';

    for i = 1:M
        Phi(i) = 180/M * (i-1) * (pi/180);
        rCoords = round(mid + x*sin(Phi(i)) + y*cos(Phi(i)));
        idx = find((rCoords > 0) & (rCoords <= N));
        newCoords = rCoords(idx);

        fpr = real(ifft(ifftshift(h.*fftshift(fft(g(:,i))))));

        f(idx) = f(idx) + fpr(newCoords) ./ M;
    end

end
```

Figure 11: Code the `fbackprojection` function. (../fbackprojection.m)

Code for Task 4

```
% Solution for part 4 of Assignment 5.
% Written by: Martin Jrgensen, tzk173

clear all;

% Read image(s).
%file = 'box.png';
file = 'mir.tif';
I1 = double(imread(file));

i = 1;
h = figure(541); set(h,'Color','White'); colormap(bone);
for M=[8 16 64 128 256]
    I2 = sinogram(I1,M);
    I3 = backprojection(I2,M);
    I4 = fbackprojection(I2,M);
    subplot(3,5,i); imagesc(I2); axis image; set(gca,'TickDir','out');
    title(strcat('Sinogram with M=',num2str(M)),'FontSize',14);
    subplot(3,5,i+5); imagesc(I3); axis image; set(gca,'TickDir','out');
    title(strcat('Backprojecting with M=',num2str(M)),'FontSize',14);
    subplot(3,5,i+10); imagesc(I4); axis image; set(gca,'TickDir','out');
    title(strcat('Filtered Backprojecting with M=',num2str(M)),'FontSize',14);

    i = i + 1;
end
```

Figure 12: Code for producing the figures for the fourth task. The `file` string variable can be set to any image. I used *mir.tif* and *box.png* for my figures. (../p4.m)