Problems Chapter 15

# Problem 15.1

Load the proj\_prof1 which contains a projection profile. Use an unfiltered inverse radon function. Deblur the image using 2 unsharp filters and display all 3 images.

## Solution

p = load('C:\dev\biomedeng\Associated Files\Chapter 15\proj\_prof1.mat')

p = p.p

[I\_b,f] = iradon(p, 1, "none")% Inverse Radon Transform

I\_b = mat2gray(I\_b)

b = fspecial('unsharp')%Unsharp filter

I\_f1 = imfilter(I\_b, b)

I\_f2 = imfilter(I\_f1, b)%Apply twice

subplot(1, 3, 1)

imshow(p)

title('Original')

subplot(1, 3, 2)

imshow(I\_f1)

title("First Filter")

subplot(1, 3, 3)

imshow(I\_f2)

title("Second Filter")

## Results

After two iterations of the unsharp filter the reconstructed image is still blurry but the square has well defined borders.

A white square with a light in the center

Description automatically generated

# Problem 15.3

Load proj\_prof3 which contains a projection profile. Reconstruct the image using four different filtered inverse radon functions: Ram-Lak, Hamming, Shepp-Logan and cosine filters. Note the differences and filter strengths.

## Solution

p = load('C:\dev\biomedeng\Associated Files\Chapter 15\proj\_prof3.mat')

p = p.p

I\_b1 = iradon(p, 1, "Ram-Lak")% Ram-Lak filter

I\_b1 = mat2gray(I\_b1)

I\_b2 = iradon(p, 1, "Hamming")% Hamming filter

I\_b2 = mat2gray(I\_b2)

I\_b3 = iradon(p, 1, "Shepp-Logan")% Shep-logan filter

I\_b3 = mat2gray(I\_b3)

I\_b4 = iradon(p, 1, "Cosine")% Cosine filter

I\_b4 = mat2gray(I\_b4)

I\_b5 = iradon(p, 1, "none")% defualt filter

I\_b5 = mat2gray(I\_b5)

subplot(2, 3, 1)

imshow(p)

title('Original')

subplot(2, 3, 2)

imshow(I\_b1)

title('Ram-Lak')

subplot(2, 3, 3)

imshow(I\_b2)

title('Hamming')

subplot(2, 3, 4)

imshow(I\_b3)

title('Shepp-Logan')

subplot(2, 3, 5)

imshow(I\_b4)

title('Cosine')

subplot(2, 3, 6)

imshow(I\_b5)

title('none')

## Results

The filters do not produce vastly different images and very little, if any, noise is filtered out. However, using no filter is very blurry and undefined compared to the filtered images.

A collage of images of an alien

Description automatically generated

# Problem 15.8

Simulate an RF pulse and take its FFT and note its narrow bandwidth around the base frequency. Use both 64 and 32 MHz as base frequencies.

## Solution

This uses a simple for loop to avoid writing everything twice, where f1 will be 64 and 32. The time vector goes from -0.5s to 0.5s.

Code:

for f1 = [64 32] %MHz

f2 = f1 / 25 %MHz

fs = 1000 %MHz

t = -0.5:1/fs:0.5 %Time vector, sinc is symmetric at t=0

baseline = cos(2\*pi\*f1\*t)

shapewave = sinc(2\*f2\*t)

shapedpusle = baseline .\* shapewave

sp\_fft = abs(fft(shapedpusle))

subplot(2, 2, 1)

plot(t, baseline)

xlabel('Time (Seconds)'); ylabel('Magnitude')

title(['Baseline Signal (', num2str(f1), ' MHz)'])

subplot(2, 2, 2)

plot(t, shapewave)

xlabel('Time (Seconds)'); ylabel('Magnitude')

title(['Shaping Waveform (', num2str(f2), ' MHz)'])

subplot(2, 2, 3)

plot(t, shapedpusle)

xlabel('Time (Seconds)'); ylabel('Magnitude')

title('Sahped RF Signal')

subplot(2, 2, 4)

plot(sp\_fft)

xlim([0 150])

xlabel('Frequency (MHz)'); ylabel('Magnitude')

title('RF Signal Frequency Domain')

figure;

end

## Results

The Fourier transforms correctly shows a very narrow bandwidth around 64 and 32 MHz

A group of graphs with lines

Description automatically generated with medium confidence

A group of graphs and diagrams

Description automatically generated

# Problem Aliasing

Load frame 18 of mri.tif. Use a 2D FFT and plot it. Downsample the signal by zeroing every second row of the frequency matrix. Inverse it and plot it together with the original image. Note the difference and what happens.

## Solution

The operation fft\_2(2:2:end)=0 sets every column in every second row to zero.

Code:

I = imread('C:/dev/biomedeng/Associated Files/Chapter 15/mri.tif')

I = im2double(I)

fft = fft2(I)% Take 2D FFT

fft\_p = fftshift(abs(fft))% Shift and abs

fft\_2 = fft%Modified fft

fft\_2(2:2:end) = 0 %Zero out every other row

fft\_p2 = fftshift(abs(fft\_2))% Shift and abs

I2 = ifft2(fft\_2)%Inverse of the modified

subplot(2, 2, 1)

imshow(I)

title('Original')

subplot(2,2,2)

imshow(I2)

title('Reconstructed')

subplot(2,2,3)

mesh(fft\_p)

title('2-D Forier Transform')

subplot(2,2,4)

mesh(fft\_p2)

title('Modified 2D FFT')

## Results

The image was downsampled and therefore we can now see the aliasing on the reconstructed image. 2 aliases can be seen, shifted halfway up and down the height of the image.

## A comparison of a graph and a scan Description automatically generated with medium confidence