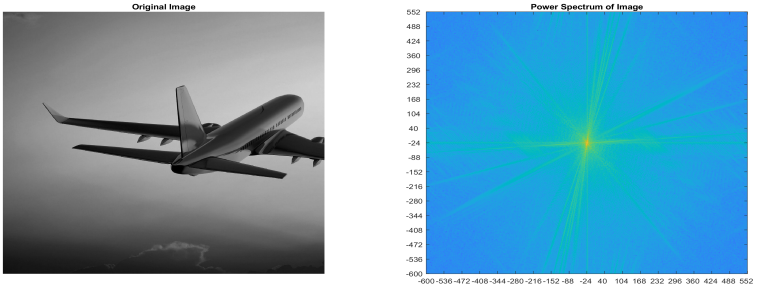
Homework 2

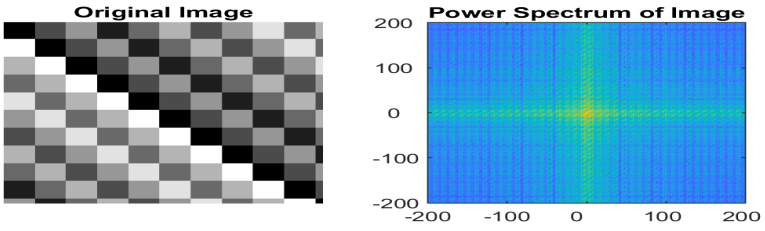
ECE 6790 - Information Processing Models in Neural Systems

Harsh Bhate (903424029)

(a). FFT of Images

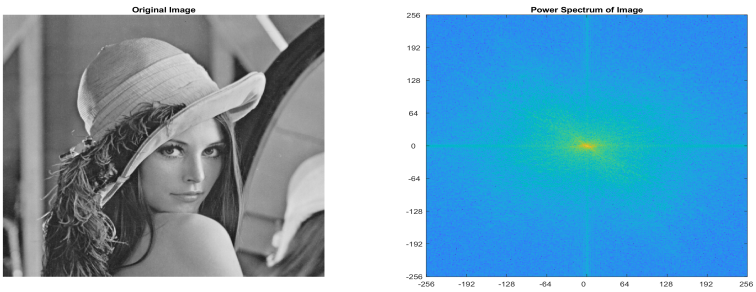


(a) Airplane

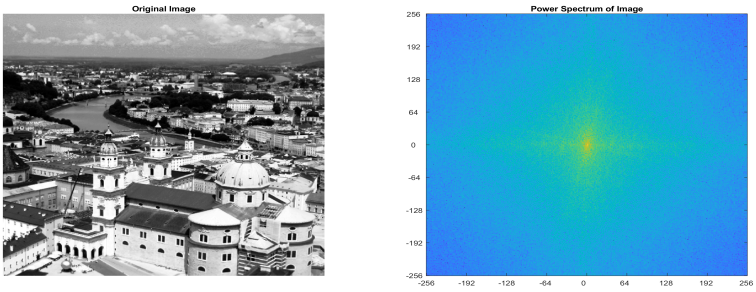


(b) Checkerboard

Figure 1: The FFT power spectrum of a collection of Images



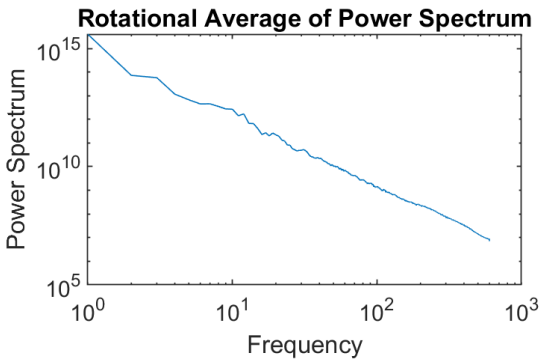
(c) Lena



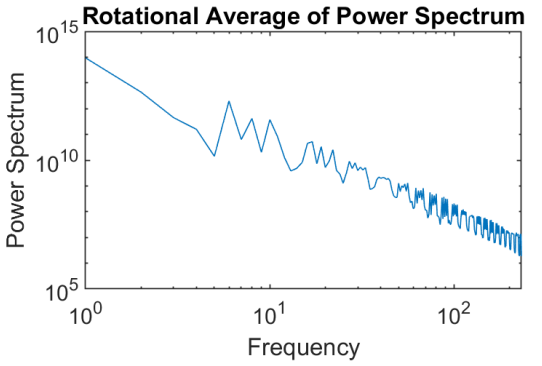
(d) Salzburg

Figure 1: The FFT power spectrum of a collection of Images

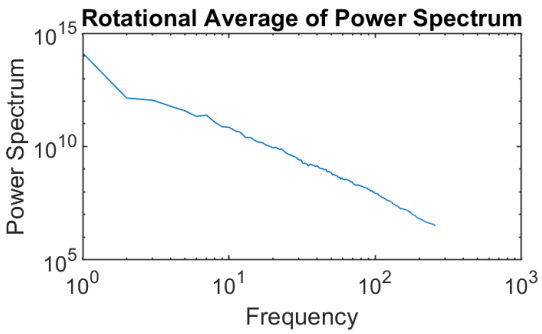
(b). Rotational Average of Power Spectrum



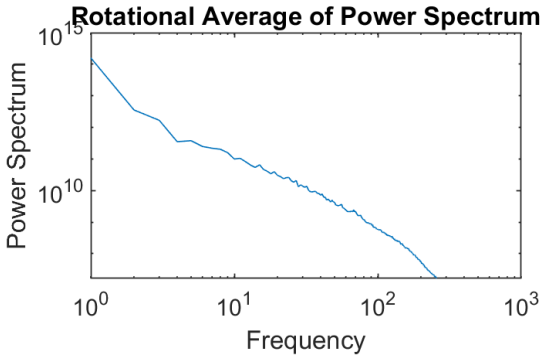
(a) Airplane (Slope=-2.907)



(b) Checkerboard (Slope = -2.918)



(c) Lena (Slope = -3.07)

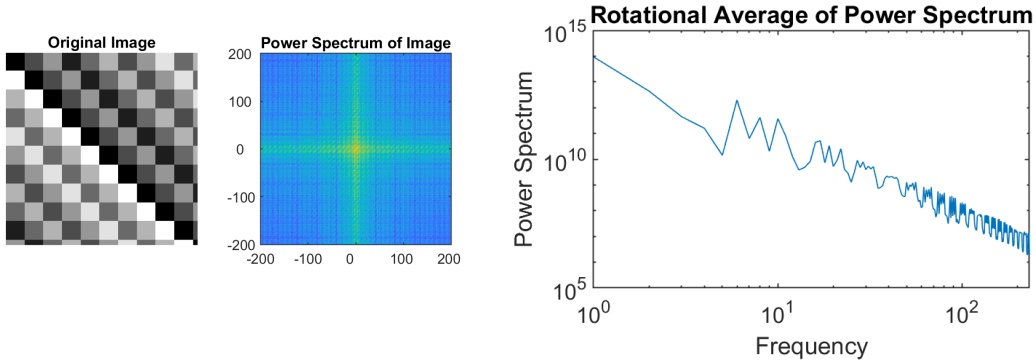


(d) Salzburg (Slope = -2.708)

Figure 2: The FFT power spectrum of a collection of Images

(c). Power Spectrum Discrepancy

Consider the Rotational Average for the following image:



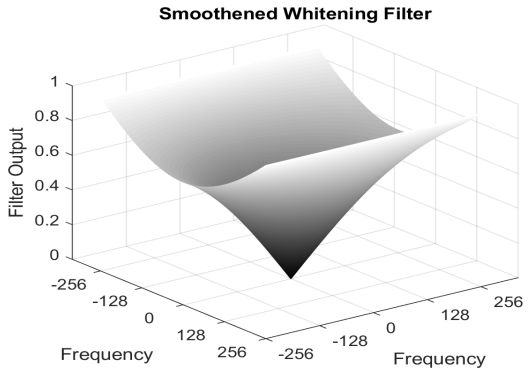
(a) Originnal Image and its Spectrum

(b) Rotational Average

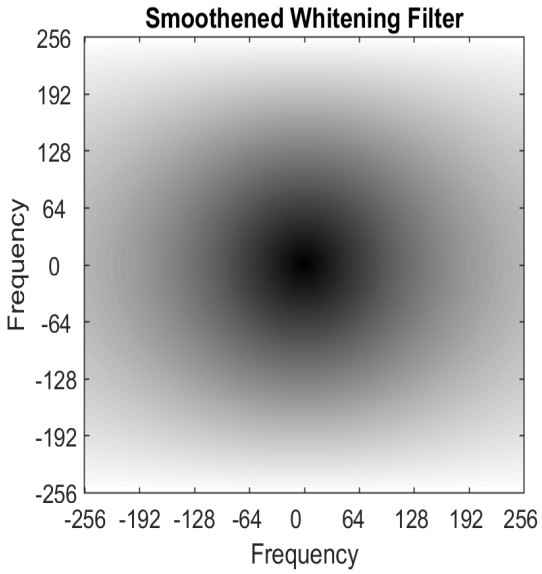
Figure 3: The Frequency Response of the Whitening Filter

As evident from the graph, for patterns where there is rapid change in color such as in a checkboard, the general rule of power decaying inversely to the square of frequency ($\frac{1}{f^2}$) does not hold.

(d). Whitening Filter

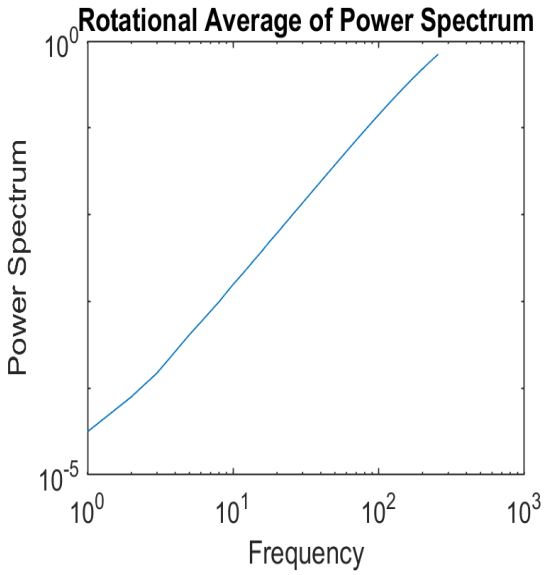


(a) Whitening Filter



(b) 2-D Plot of Smoothened Whitening Filter

Figure 4: Whitening Filter (*cont.*)

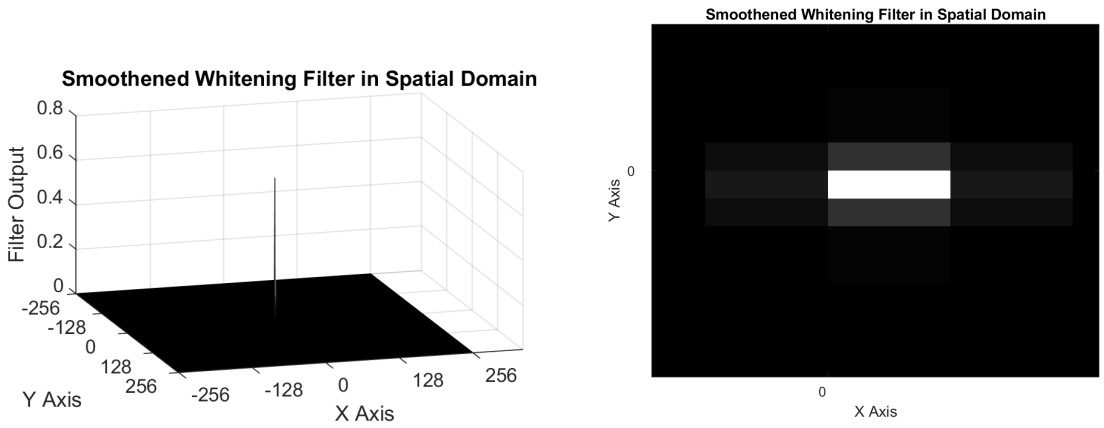


(c) Rotational Power Spectrum Average of the Filter

Figure 4: Whitening Filter

(e). Spatial Whitening Filter

The Spatial Repsonse of the filter is:



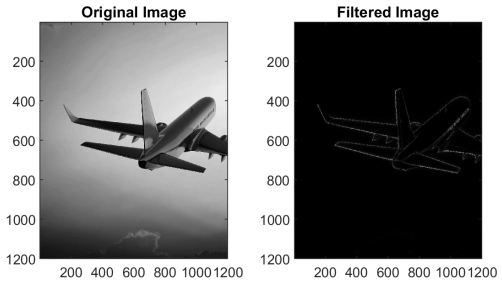
(a) 3-D Spatial Response

(b) Zoomed 2-D Spatial Response

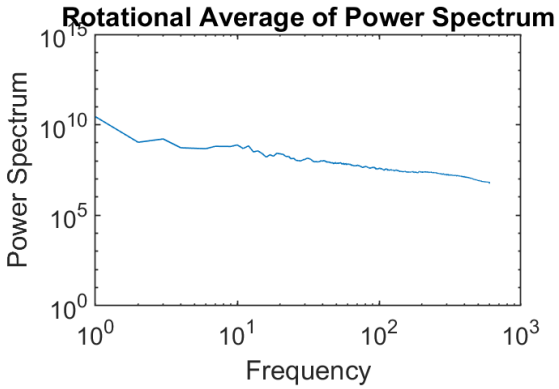
Figure 5: Whitening Filter (*cont.*)

As evident from the spatial response, the whitening filter is analogous to the receptive field model of the visual Ganglion cell. The centre part (white) corresponds to the excitory region while the dark part corresponds to the inhibitory region.

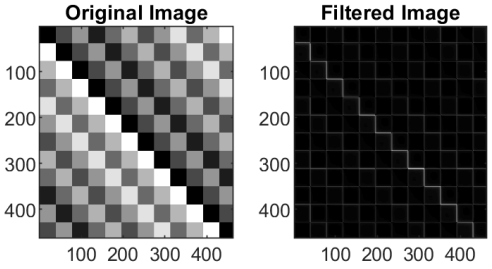
(f). Whitening Filter Applied to Sample Images



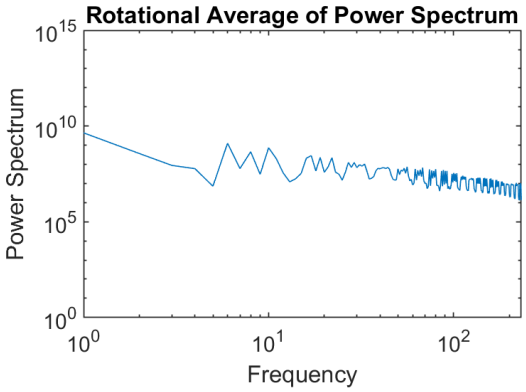
(a) Filter Output



(b) Output Power Output (Slope = -0.958)

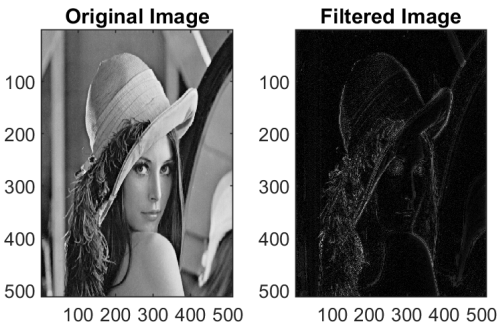


(c) Filter Output

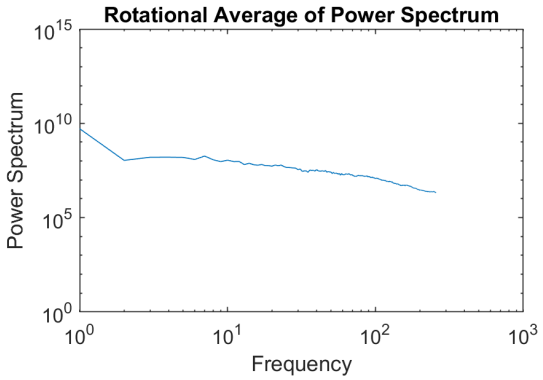


(d) Output Power Output (Slope = -1.01)

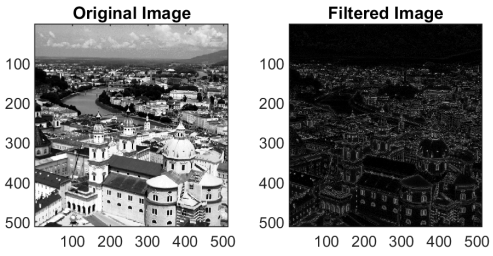
Figure 6: Whitening Filters applied to example Images



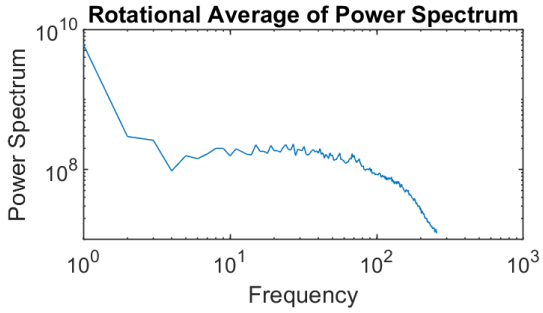
(a) Filter Output



(b) Output Power Output (Slope = -1.917)



(c) Filter Output



(d) Output Power Output (Slope = -0.805)

Figure 7: Whitening Filters applied to example Images

Appendix

```
%% ECE 8833 (Homework 2)
clear all;
clc;
%% Reading an Image
img = imread('images/airplane.jpg');
% Converting image to grayscale
if size(img, 3) == 3
    img = rgb2gray(img);
end
%Resizing image to square
dim = size(img);
minDim = min(dim);
img = img(1:minDim, 1:minDim);
%Displaying the resized imaage
subplot(1,2,1);
imshow(img);
title('Original Image');
%% (a). Computing the FFT, Power Spectrum
fImg = fft2(img);   %Computing the FFT
fImg = fftshift(fImg); %Centering the FFT at origin
psdImg = abs(fImg).^2; %Computing magnitude of FFT
subplot(1,2,2);
imagesc(log(1+psdImg));
%Setting Scales and title for fourier plot
xticklabels = -minDim/2:64:minDim/2;
xticks = linspace(1, size(log(1+psdImg), 2), numel(
    xticklabels));
set(gca, 'XTick', xticks, 'XTickLabel', xticklabels)
yticklabels = -minDim/2:64:minDim/2;
yticks = linspace(1, size(log(1+psdImg), 1), numel(
    yticklabels));
set(gca, 'YTick', yticks, 'YTickLabel', flipud(
    yticklabels(:)))
set(gca,'dataAspectRatio',[1 1 1])
title ('Power Spectrum of Image');
%% (b). Rotational Average of the Power Spectrum
rotationalAverage(psdImg);
%% (d). Whitening Filter for natural images
%Initializing the filter
[xgrid, ygrid] = meshgrid(1:size(img,2), 1:size(img,1));
midx = floor(size(img,1)/2);
midy = floor(size(img,2)/2);
filter = sqrt((xgrid-midx).^2 + (ygrid-midy).^2);
filter = filter./max(filter);
```

```
% Applying Gaussian smoothening
sigma = 0.7;
smoothenedFilter = imgaussfilt(filter,sigma);
%Finding Rotational Average of smoothened filter
rotationalAverage(abs(smoothenedFilter).^2);
%Plotting the filter
figure(3);
mesh(filter);
%imagesc(filter);
colormap gray;
% plot(smoothenedFilter);
title('Smoothened Whitening Filter');
xlabel('Frequency');
ylabel('Frequency');
zlabel('Filter Output');
xticklabels = -minDim/2:128:minDim/2;
xticks = linspace(1, size(filter, 2), numel(xticklabels))
    ;
set(gca, 'XTick', xticks, 'XTickLabel', xticklabels)
yticklabels = -minDim/2:128:minDim/2;
yticks = linspace(1, size(filter, 1), numel(yticklabels))
    ;
set(gca, 'YTick', yticks, 'YTickLabel', flipud(
    yticklabels(:)))
%% (e). Filter in the Spatial Domain
spatialFilter = ifft2(ifftshift(smoothenedFilter));
figure(4);
%imagesc(fftshift(abs(spatialFilter)));
mesh(fftshift(abs(spatialFilter)));
colormap gray;
title('Smoothened Whitening Filter in Spatial Domain');
xlabel('X Axis');
ylabel('Y Axis');
zlabel('Filter Output');
xticklabels = -floor(minDim/2):128:floor(minDim/2);
xticks = linspace(1, size(filter, 2), numel(xticklabels))
    ;
set(gca, 'XTick', xticks, 'XTickLabel', xticklabels)
yticklabels = -floor(minDim/2):128:floor(minDim/2);
yticks = linspace(1, size(filter, 1), numel(yticklabels))
    ;
set(gca, 'YTick', yticks, 'YTickLabel', flipud(
    yticklabels(:)))
%% (f). Whitening filter on example image
output = fImg.*smoothenedFilter;
psd_output = abs(output).^2;
```

```
82  out_img = abs(ifft2(ifftshift(output)));
83  rotationalAverage(psd_output);
84  figure();
85  subplot(1,2,1);
86  imagesc(img);
87  colormap gray;
88  title('Original Image');
89  subplot(1,2,2);
90  imagesc(out_img);
91  colormap gray;
92  title('Filtered Image');
93  %% Functions
94  function rotationalAverage (img)
95      % Extracting the coordinates
96      midx = floor(size(img,1)/2);
97      midy = floor(size(img,2)/2);
98      error = 0.5;
99      meanValues = zeros(1, (midx+rem(midx,2)));
100     for r = 1:(midx+rem(midx,2))
101         [xgrid, ygrid] = meshgrid(1:size(img,2), 1:size(
                img,1));
102         equation = ((xgrid-midx).^2 + (ygrid-midy).^2);
103         mask =  (equation >= (r-error).^2 & equation <= (
                r+error).^2);
104         values = img(mask);
105         meanValues(r) = nanmean(values);
106     end
107     figure();
108     x = 1:(midx+rem(midx,2));
109     loglog(x, meanValues);
110     disp(meanValues);
111     title('Rotational Average of Power Spectrum');
112     xlabel('Frequency');
113     ylabel('Power Spectrum');
114     axis([xlim 10^0 10^15 ylim 10^0 10^15]);
115     p = polyfit(log10(1+(2*pi).*x) , log10(1+meanValues)
            ,1);
116     fprintf("Slope is: %f \n", p(1));
117 end
```