**Digital Image Processing – Homework #5**

**Juan Silva**

**ECE 595**

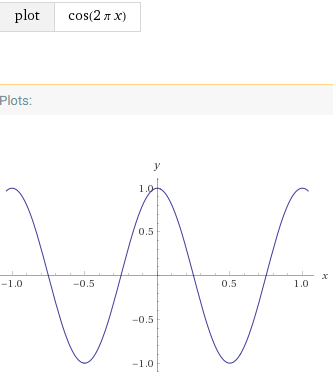
**October 2, 2019**

1. Consider a spatial mask that averages the four closest neighbors of a point (x, y), but excludes the point itself from the average.   
   (a) Find the equivalent filter H(u,v) in the frequency domain.   
   (b) Show that your result is a lowpass filter.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |



(b)



,



The function G(u,v) will peak at 1 when U = M/2 and dips at -1 when U = M.

1. The following Matlab instructions create a white rectangle in the center of black square.   
     
   f=zeros(30,30);   
   f(5:24,13:17)=1;   
   imshow(f,'InitialMagnification', 'fit')   
     
   Write a sequence of Matlab instructions to create the Fourier transform of the given figure. Use shift property to place the highest frequency at the center of the figure.

**hw5\_q2.m:**

clear, clc

format short, format compact

%Generate centered white rectangle

f = zeros(30,30);

f(5:24, 13:17) = 1;

imshow(f, 'InitialMagnification', 'fit');

%apply fast fourier transform

F = fft2(f);

F\_abs = abs(F);

%Display Fourier-transformed image

figure

imshow(F\_abs,[],'InitialMagnification', 'fit');

%Apply padding to filter

F = fft2(f, 256, 256);

F\_abs = abs(F);

%Display edges

figure

imshow(F\_abs, []);

%shift towards center

F\_shift = fftshift(F);

F\_abs = abs(F\_shift);

figure

imshow(F\_abs,[]);

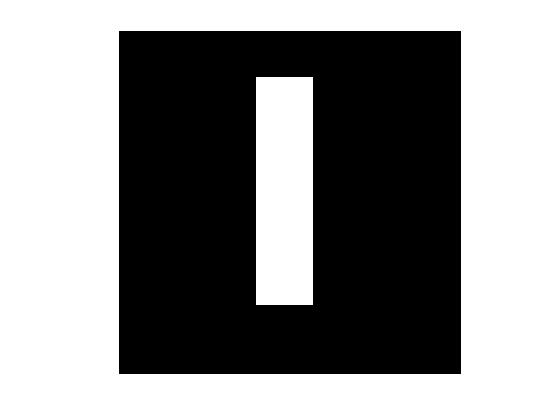
%reduce contrasting using log function

F\_cntr = log(1 + F\_abs);

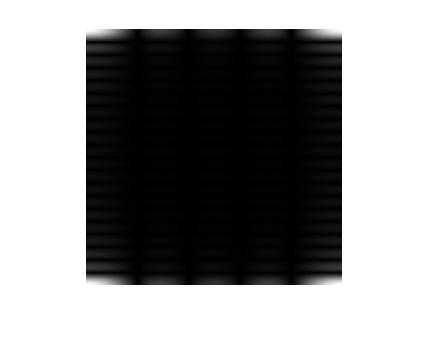
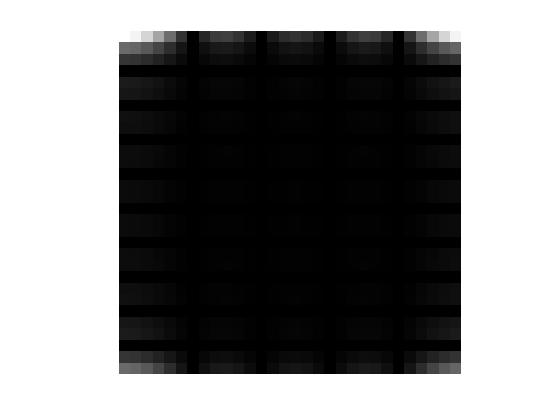
figure

imshow(F\_cntr,[]);

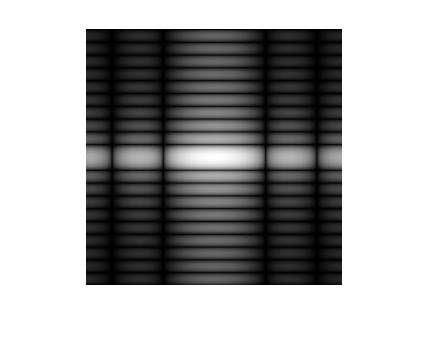
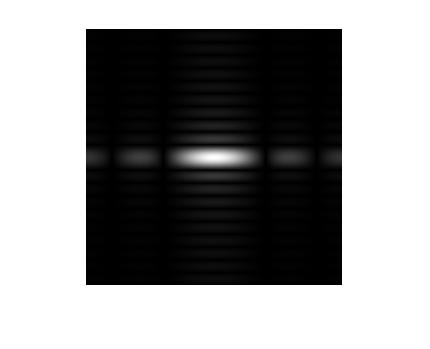
**Results**



Generated White Square



Fast Fourier Transform Padding Applied



Shift to Center Contrast Reduced

1. The file “noisy.png” from the course website is an image with a certain amount of periodic noise. The noise looks like a pattern of close lines running diagonally from bottom left to top right. You can verify this by zooming in the image. Using the method of notch filtering, remove this noise. Show:   
   a. The Matlab code   
   b. The mask you used   
   c. The resulting image.

clear, clc

format compact, format short

%Open image

I = imread('noisy.png');

imshow(I, []);

%Apply fourier transform

Itemp = fft2(double(I));

%Apply shifting to center fourier

Ishift = fftshift(Itemp);

I = abs(Ishift);

%Display noise

figure

imshow(log(I), []);

%Find the peaks via inspection

Y = [125, 175; 125, 425; 400, 100; 400, 380];

%upper left, upper right, lower left, lower right

%Create a mask of zeros where

M = ones(size(I,1), size(I,2));

size(I); %image is 512x512

N = 15;

z = zeros(N,N); %

%Apply a mask

for i = 1:length(Y)

M(Y(i,2) - floor(N/2) : Y(i,2) + floor(N/2), Y(i,1) - ...

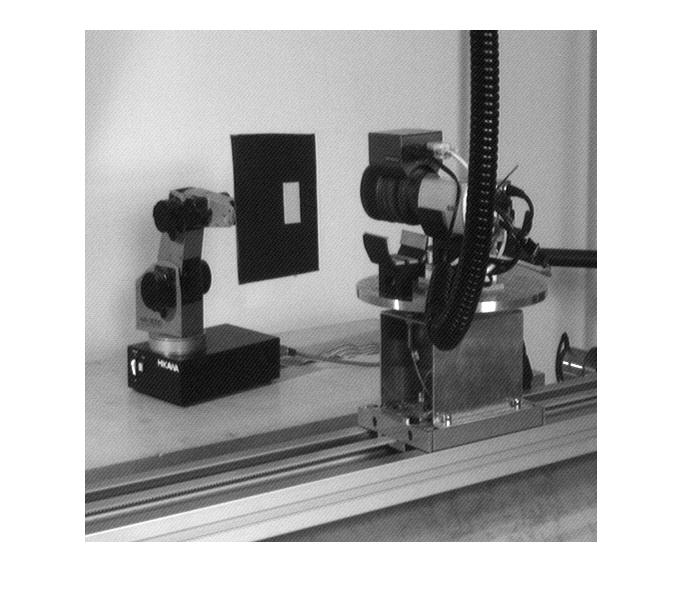
floor(N/2): Y(i,1) + floor(N/2)) = z(:,:);

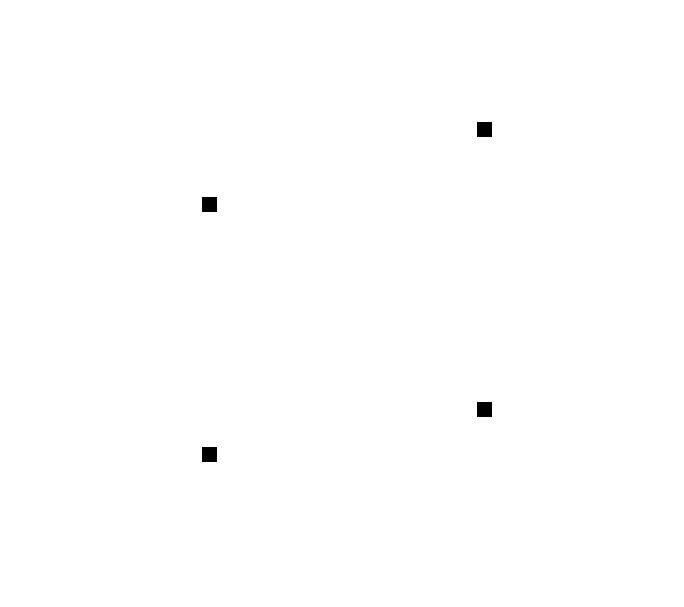
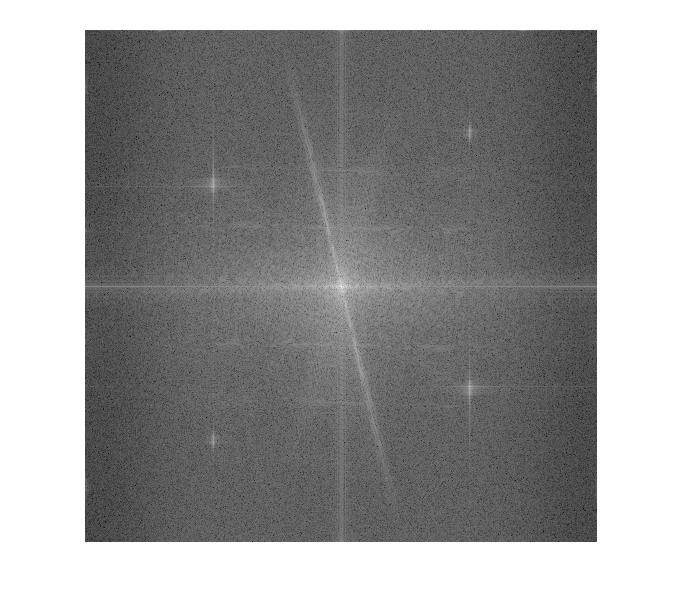
end

figure

imshow(M, []);

**Results**

  
Original Noisy Image



Fourier Transform Identified Peaks