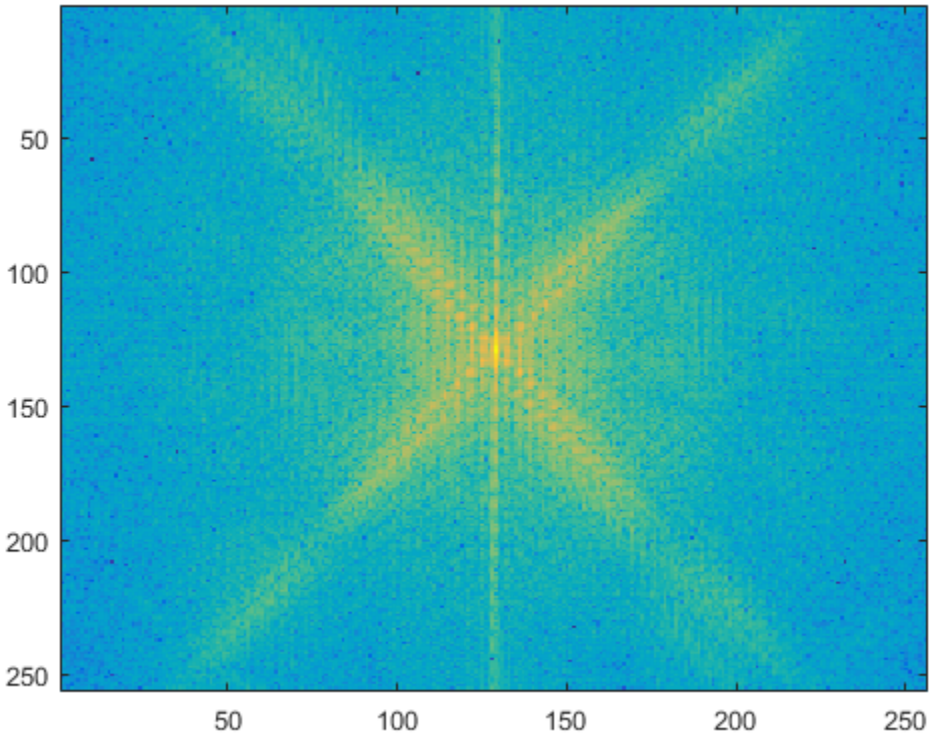
Table of Contents

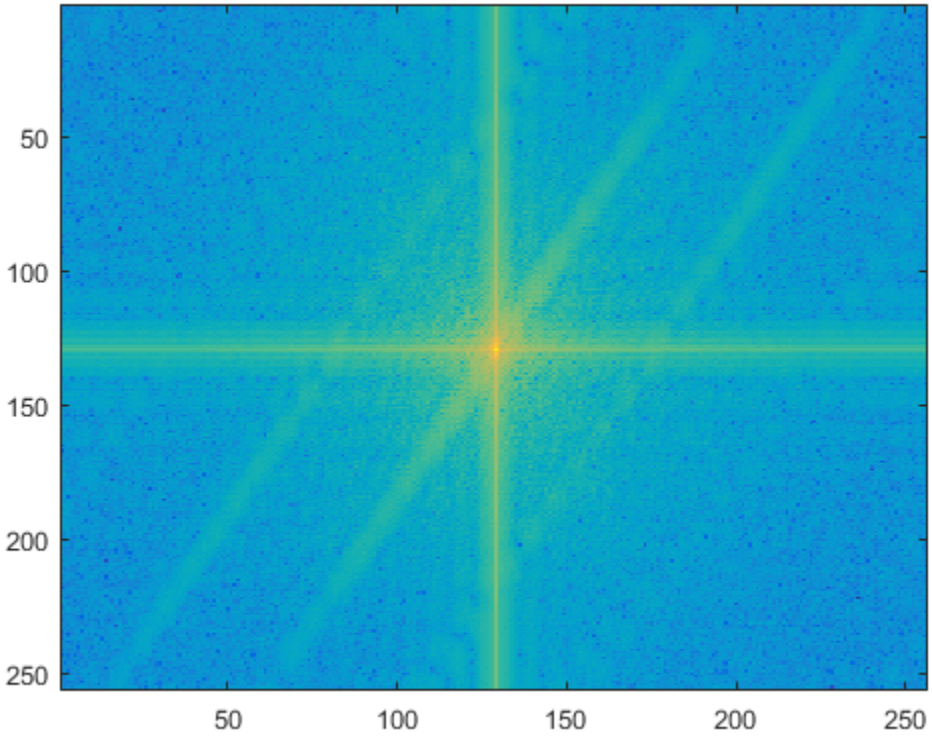

```
% Lab 3
%%Spatial Freq
fence = imread('fence.jpg');
fence_256 = imresize(fence, [256 256]);
sky = imread('sky.jpg');
sky_256 = imresize(sky, [256 256]);
figure;
imshow(fence_256)
fence_grayscale = rgb2gray(fence_256);
figure;
imagesc(log(abs(fftshift(fft2(fence_grayscale)))))
figure;
imshow(sky_256)
sky_grayscale = rgb2gray(sky_256);
figure;
imagesc(log(abs(fftshift(fft2(sky_grayscale)))))
% Applying a low pass filter to the bricks yields the following
lpf = fspecial('gaussian',5,1.7);
hpf = [0 0 0 0 0;
       0 0 0 0 0;
       0 0 1 0 0;
       0 0 0 0 0;
       0 0 0 0 0] - lpf;
fencelowfilter = imfilter(fence_grayscale, lpf, 'replicate');
fencehighfilter = imfilter(fence_grayscale, hpf, 'replicate');
figure;
imshow(uint8(fencelowfilter))
figure;
imagesc(log(abs(fftshift(fft2(fencelowfilter)))))
figure;
imagesc(angle(fftshift(fft2(fencelowfilter))))
figure;
imshow(uint8(fencehighfilter))
figure;
imagesc(log(abs(fftshift(fft2(fencehighfilter)))))
figure;
imagesc(angle(fftshift(fft2(fencehighfilter))))
%%low high energy fence
energylow = norm(fft2(fencelowfilter),2);
energyhigh = norm(fft2(fencehighfilter),2);
fprintf('Fence HighPassFilter Energy = %d, LowPassFilter Energy = %d
\n', energyhigh, energylow);
skylow = imfilter(sky_grayscale, lpf, 'replicate');
skyhigh = imfilter(sky_grayscale, hpf, 'replicate');
```

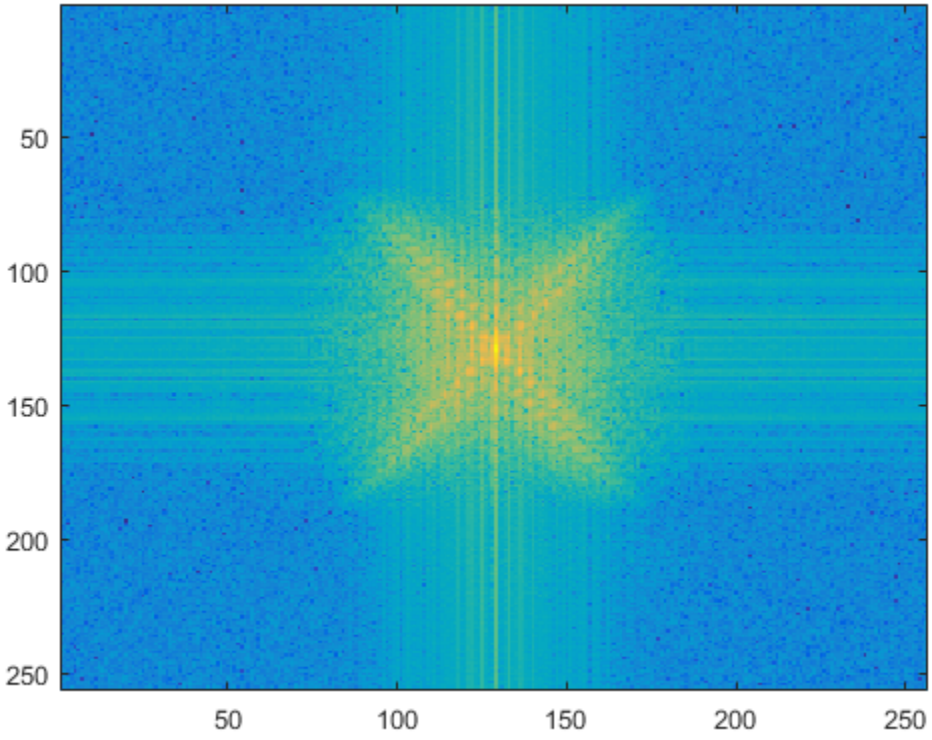
```
figure;
imshow(uint8(skylow))
figure;
imagesc(log(abs(fftshift(fft2(skylow)))))
figure;
imagesc(angle(fftshift(fft2(skylow))))
figure;
imshow(uint8(skyhigh))
figure;
imagesc(log(abs(fftshift(fft2(skyhigh)))))
figure;
imagesc(angle(fftshift(fft2(skyhigh))))
% low high energy sky
energy_lpf = norm(fft2(skylow),2);
energy_hpf = norm(fft2(skyhigh),2);
fprintf('Sky HighPassFilter Energy = %d, LowPassFilter Energy = %d\n',
 energy_hpf, energy_lpf);

% signals added to fft
sky_fft = fftshift(fft2(sky_grayscale));
figure;
imagesc(log(abs(sky_fft)));
amt_to_add = 10e5;
sky_fft(128+64,128+64) = amt_to_add;
sky_fft(128-64,128-64) = amt_to_add;
sky_fft(128-64,128+64) = amt_to_add;
sky_fft(128+64,128-64) = amt_to_add;
sky_fft(128,128+64) = amt_to_add;
sky_fft(128,128-64) = amt_to_add;
sky_fft(128+64,128) = amt_to_add;
sky_fft(128-64,128) = amt_to_add;
figure;
imagesc(log(abs(sky_fft)));
sky_fft_modified = ifftshift(sky_fft);
sky_gray_modfied = uint8(ifft2(sky_fft_modified));
figure;
imshow(sky_gray_modfied);
sky_fft(128+64,128+64) = 0;
sky_fft(128-64,128-64) = 0;
sky_fft(128-64,128+64) = 0;
sky_fft(128+64,128-64) = 0;
sky_fft(128,128+64) = 0;
sky_fft(128,128-64) = 0;
sky_fft(128+64,128) = 0;
sky_fft(128-64,128) = 0;
figure;
imagesc(log(abs(sky_fft)));
sky_fft_modified = ifftshift(sky_fft);
sky_gray_modfied = uint8(ifft2(sky_fft_modified));
figure;
imshow(sky_gray_modfied);
```

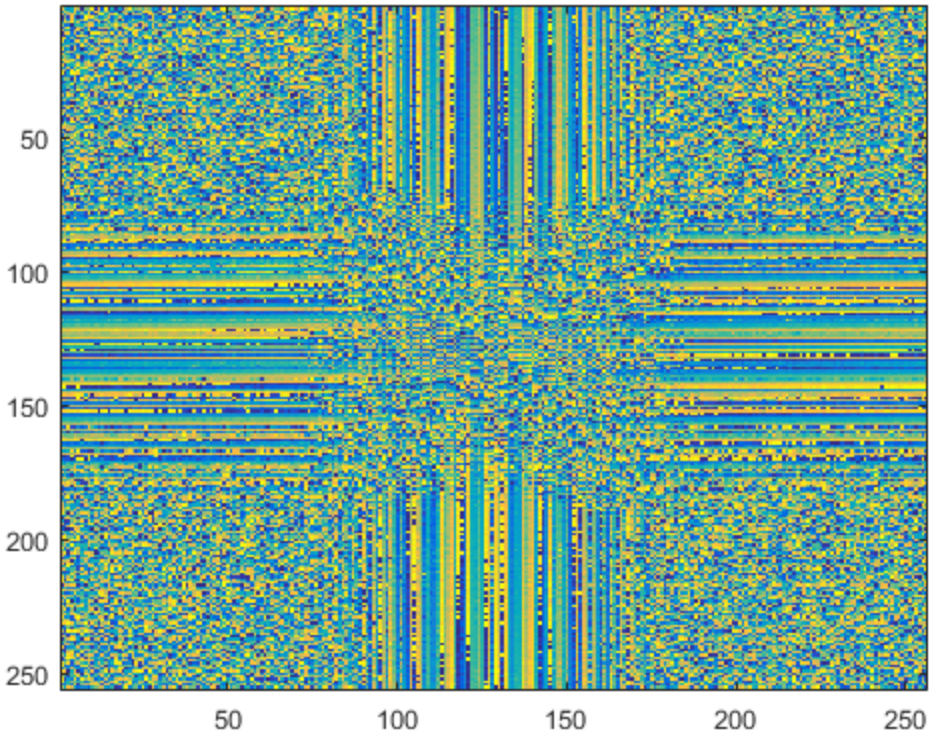
```
Fence HighPassFilter Energy = 2.126862e+05, LowPassFilter Energy =
 3.948228e+06
```

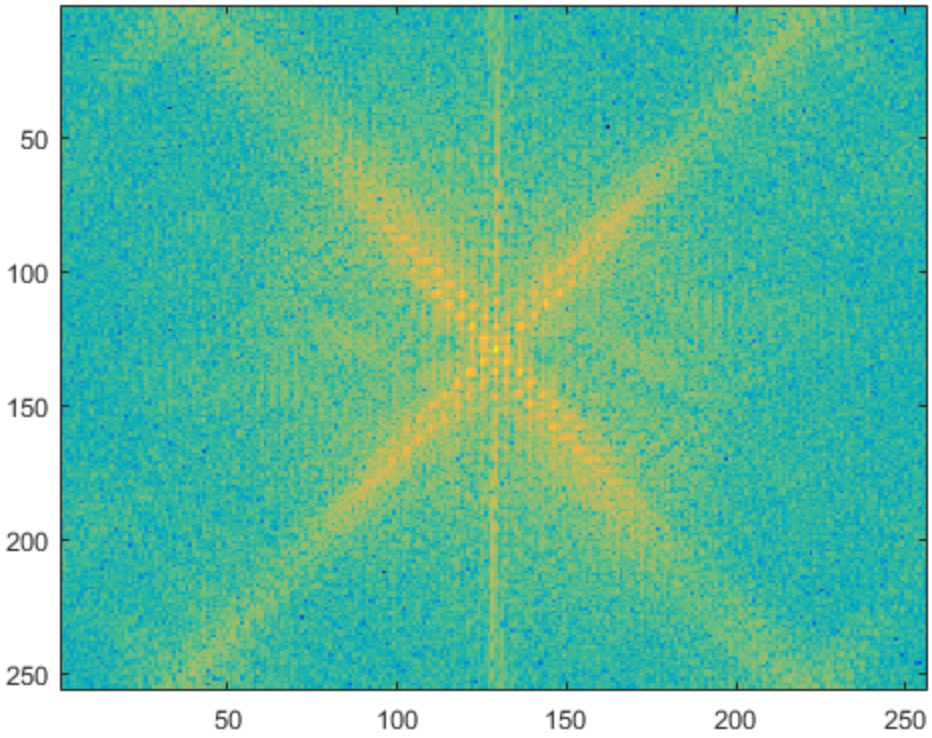
```
Sky HighPassFilter Energy = 3.813044e+04, LowPassFilter Energy =
 9.663096e+06
Warning: Displaying real part of complex input.
```

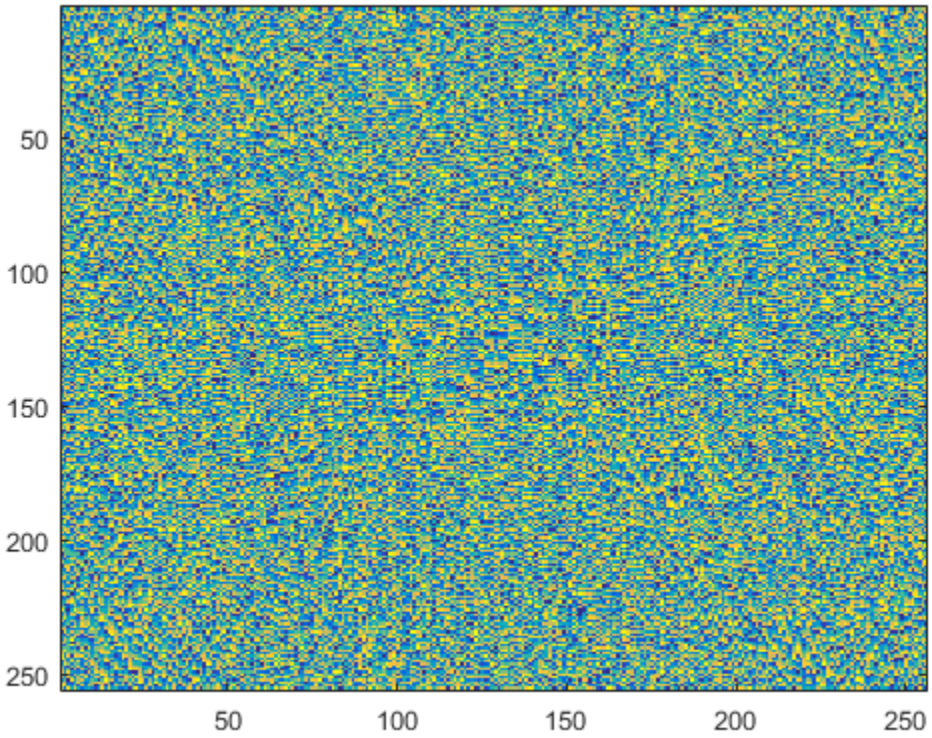


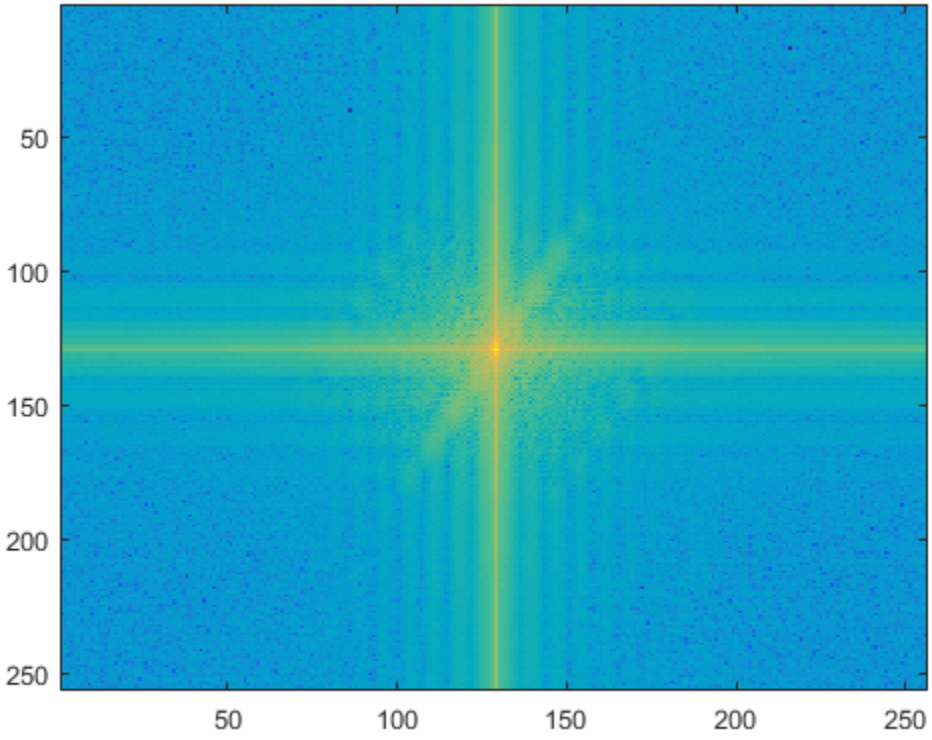

50
100
150
200
250
50
100
150
200
250

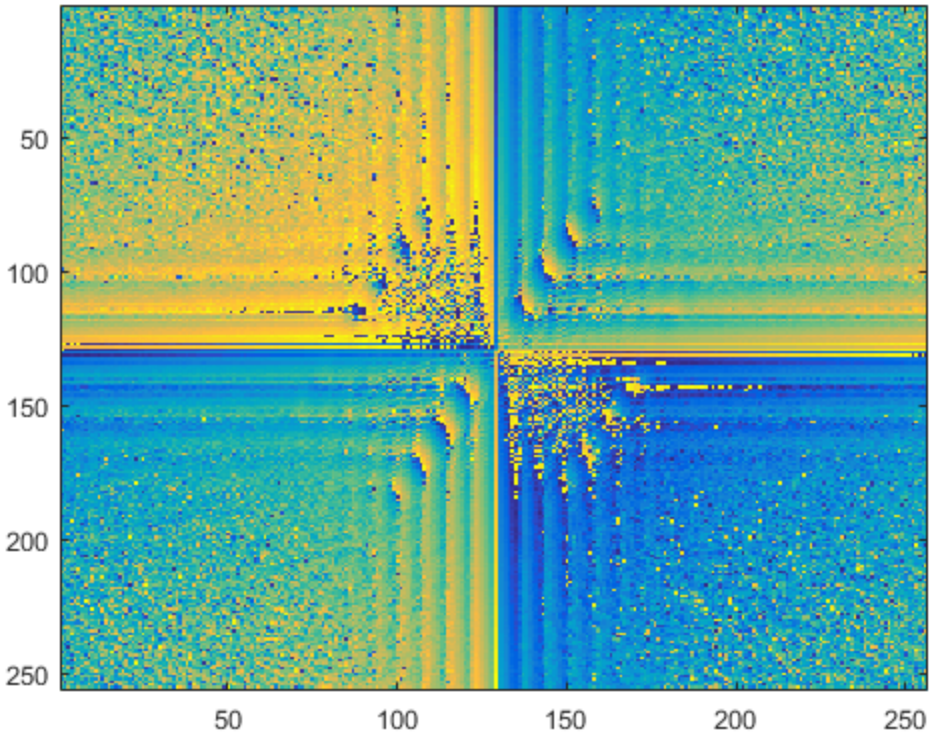

50
100
150
200
250
50
100
150
200
250

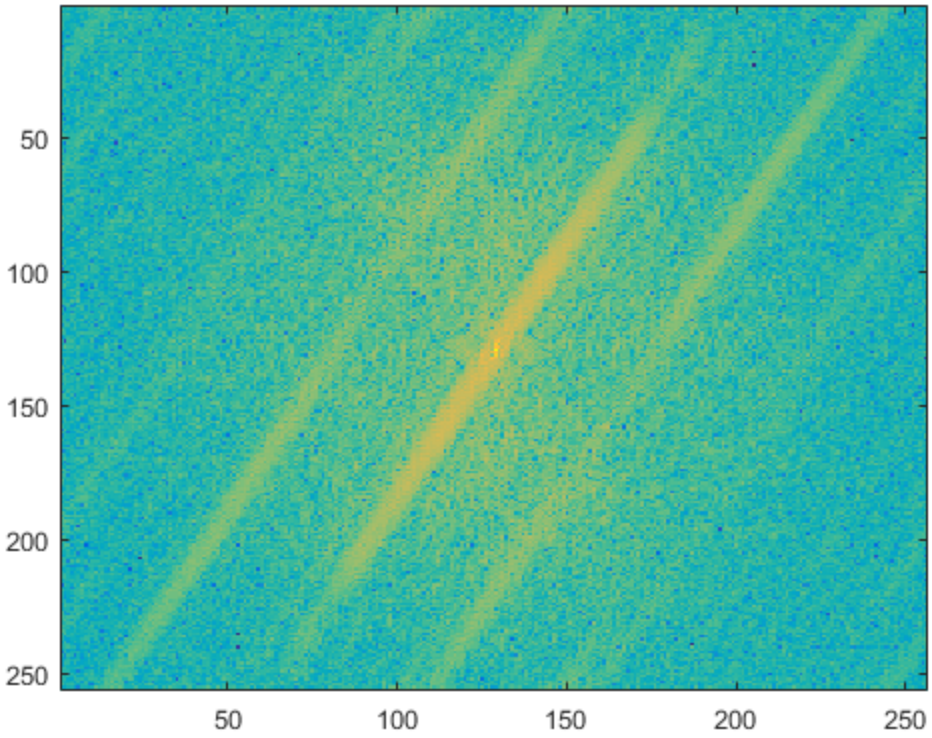

50
100
150
200
250
50
100
150
200
250

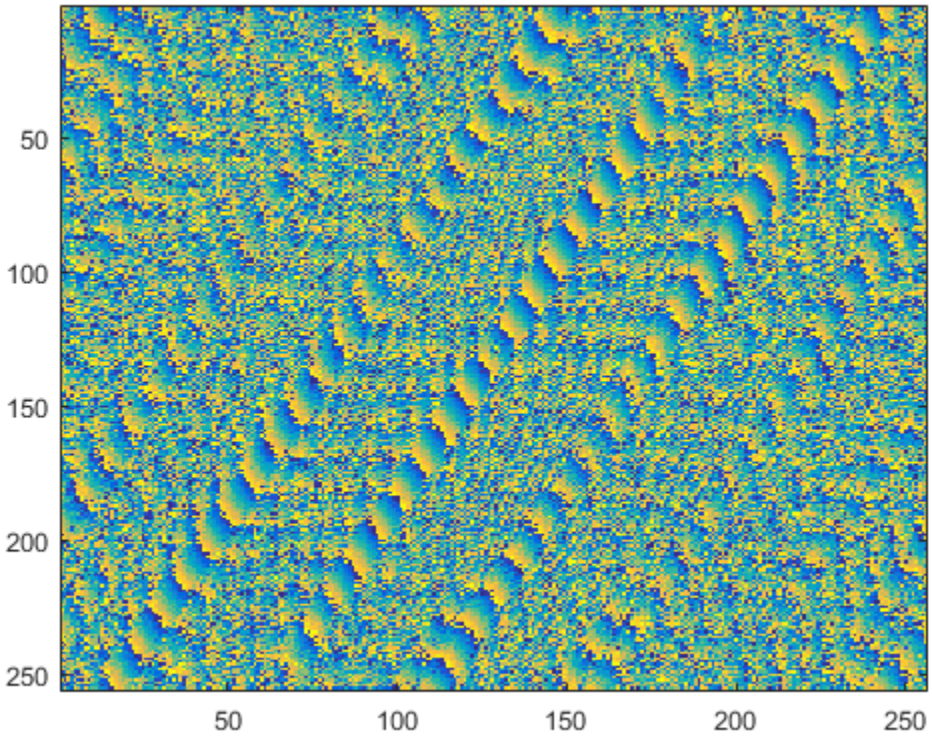

50
100
150
200
250
50
100
150
200
250

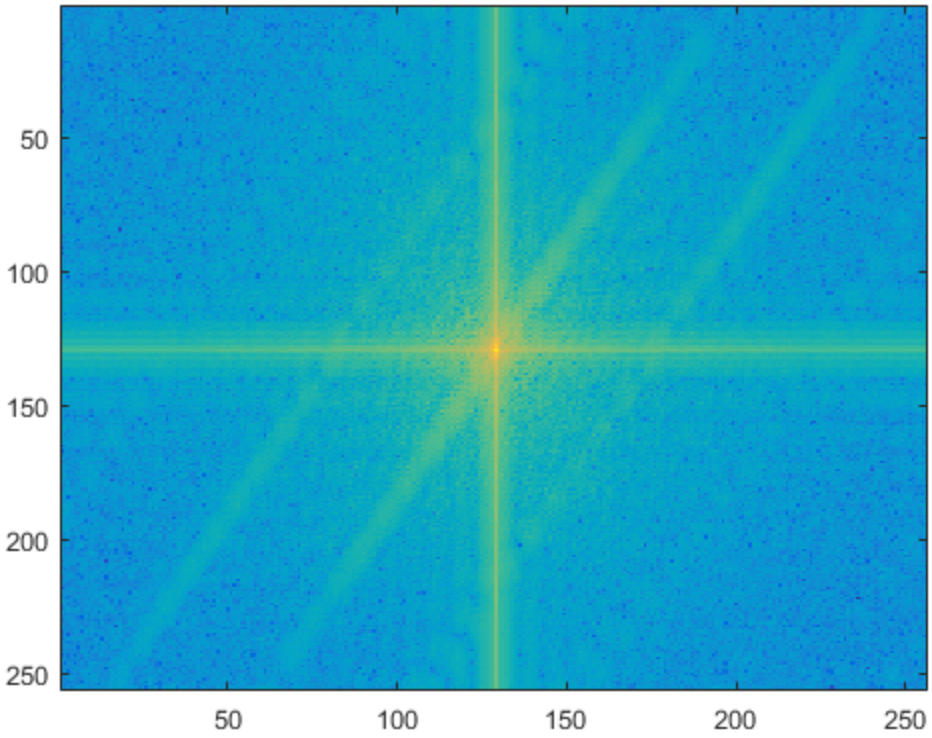

50
100
150
200
250
50
100
150
200
250

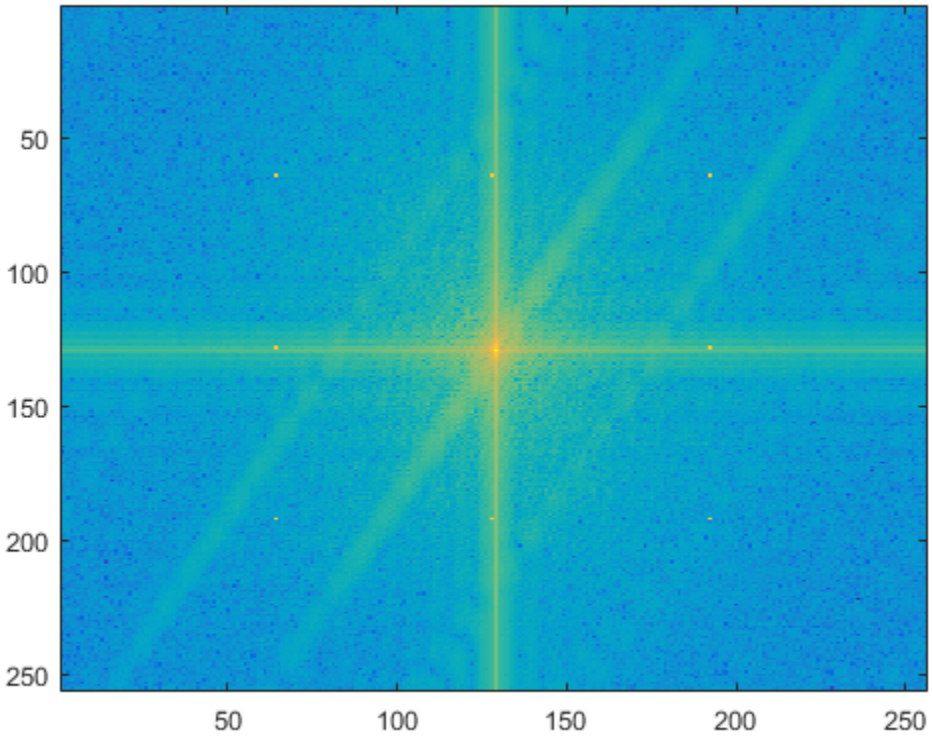

50
100
150
200
250
50
100
150
200
250

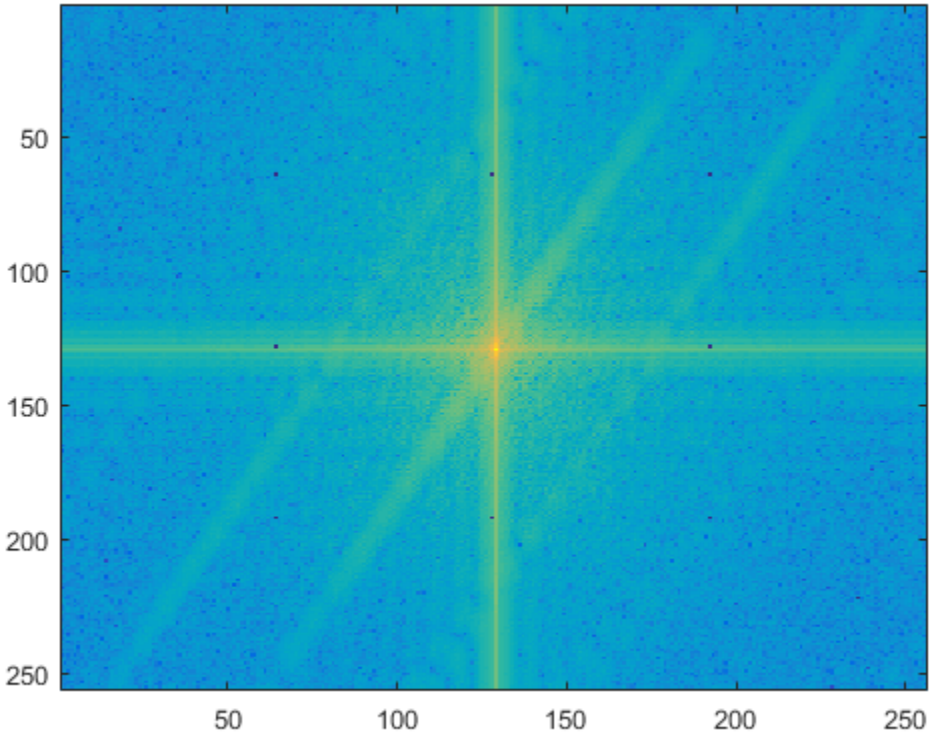

50
100
150
200
250
50
100
150
200
250


50
100
150
200
250
50
100
150
200
250


50
100
150
200
250
50
100
150
200
250


50
100
150
200
250
50
100
150
200
250


50
100
150
200
250
50
100
150
200
250

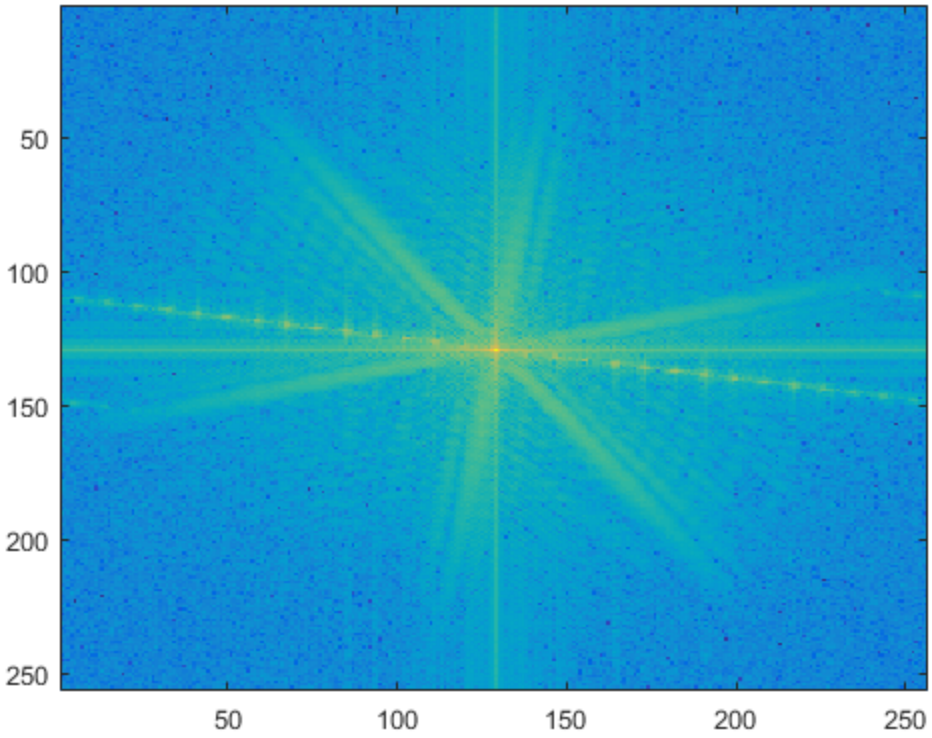

50
100
150
200
250
50
100
150
200
250

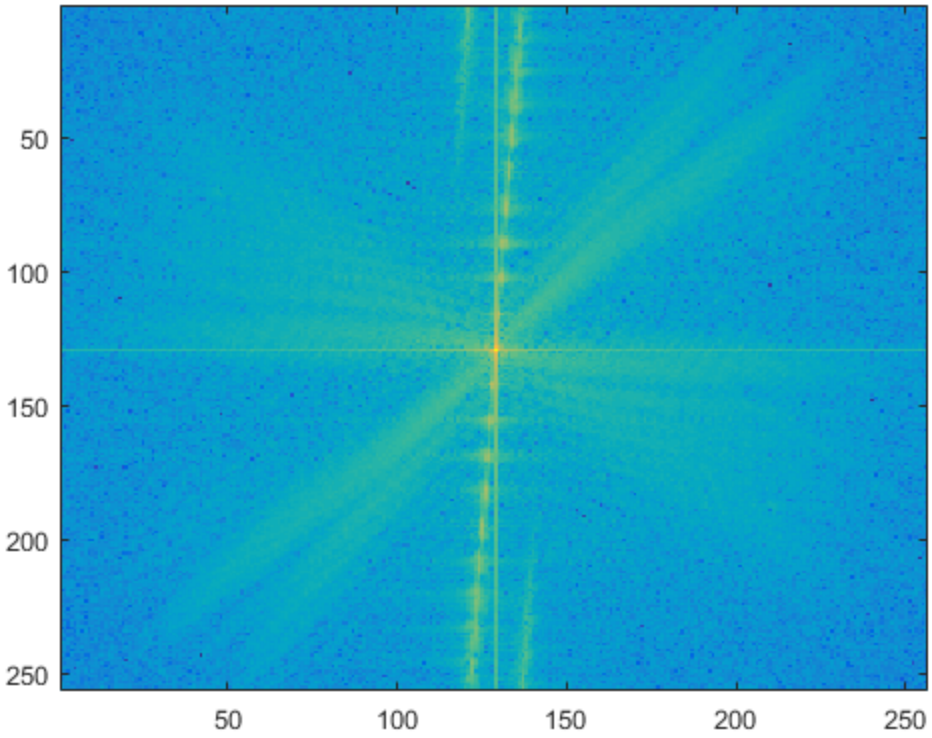
Spatial Frequencies 2

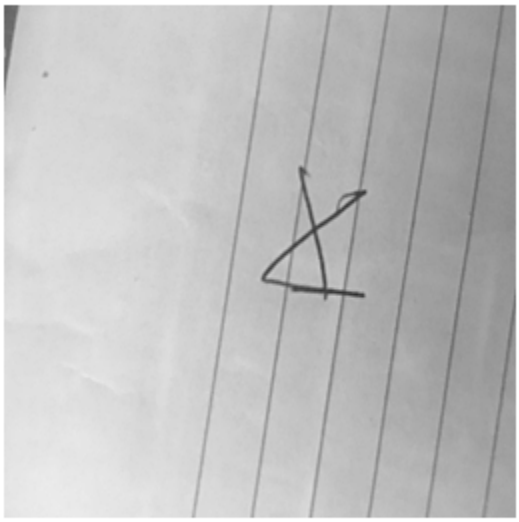
Load in the images

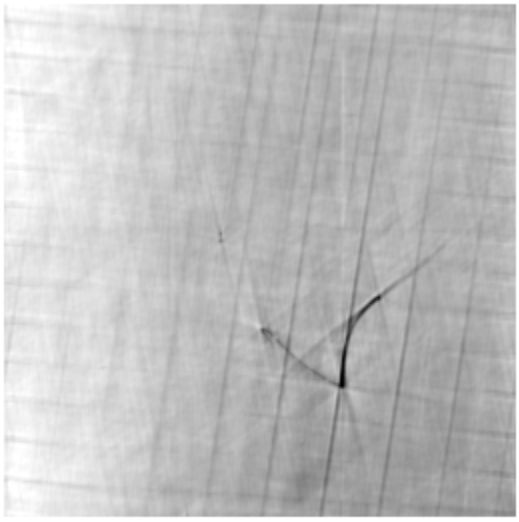
```
letterA = imread('letter_A.jpg');
letterB = imread('letter_B.jpg');

letterA_256 = imresize(letterA, [256 256]);
letterB_256 = imresize(letterB, [256 256]);
letterA_gray = rgb2gray(letterA_256);
letterA_gray = imrotate(letterA_gray, 90);
letterB_gray = rgb2gray(letterB_256);
letterA_fft = fft2(letterA_gray);
letterB_fft = fft2(letterB_gray);
figure;
imshow(letterA_gray);
figure;
imagesc(log(abs(fftshift(letterA_fft))));
figure;
imshow(letterB_gray);
figure;
imagesc(log(abs(fftshift(letterB_fft))));
unsharpA = imsharpen(letterA_gray);
unsharpB = imsharpen(letterB_gray);
figure;
imshow(unsharpA);
figure;
imshow(unsharpB);

% phase of images
```

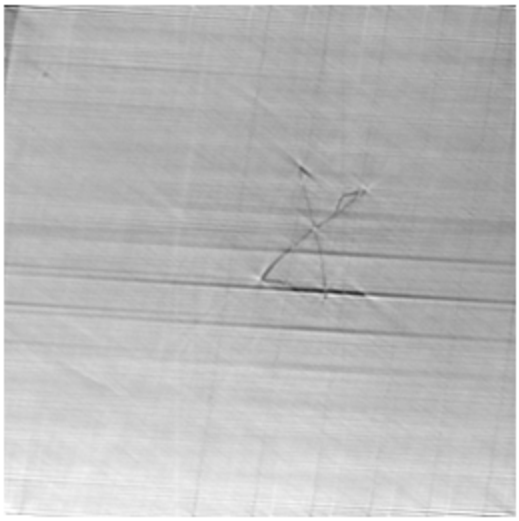
```
A_mag = abs(letterA_fft);
B_mag = abs(letterB_fft);
A_phase = angle(letterA_fft);
B_phase = angle(letterB_fft);
swapped1 = A_mag .* exp(1i*B_phase);
swapped2 = B_mag .* exp(1i*A_phase);
s1_img = uint8(ifft2(swapped1));
s2_img = uint8(ifft2(swapped2));
figure;
imshow(s1_img);
figure;
imshow(s2_img);
```


50
100
150
200
250
50
100
150
200
250


50
100
150
200
250
50
100
150
200
250







More Fun with Frequencies

```
anthony = imread('anthonyface.jpg');
imshow(anthony);
anthony = imcrop(anthony, [554 90 1500 2400]);
anthony = imresize(anthony, [256 256]);
imshow(anthony);
figure();
jacky = imread('jackyface.jpg');
imshow(jacky);
jacky = imcrop(jacky, [454 90 1500 2400]);
jacky = imresize(jacky, [256 256]);
imshow(jacky);
figure();

fixedPt = [82 118; 172 120];
movingPt = [90 128; 176 139];

tform = fitgeotrans(movingPt,fixedPt,'NonreflectiveSimilarity');
Jregistered = imwarp(anthony,tform,'Output',imref2d(size(anthony)));
figure;
imshowpair(anthony,Jregistered)
lpf = fspecial('gaussian',5,10);
hpf = [0 0 0 0 0;
       0 0 0 0 0;
       0 0 1 0 0;
       0 0 0 0 0;
       0 0 0 0 0] - lpf;

anthonylowfilt = imfilter(double(anthony), lpf, 'replicate');
```

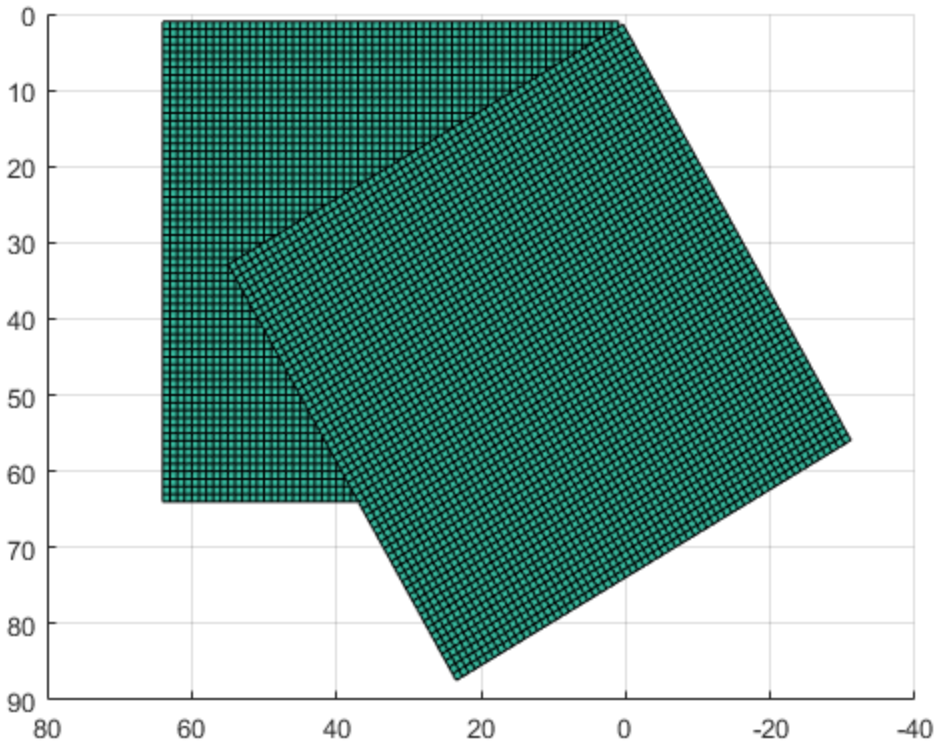
```
Jregisteredhighfilt = imfilter(double(Jregistered), hpf, 'replicate');
figure;
imshow(uint8(anthonylowfilt + 15*Jregisteredhighfilt));
```

Warning: Image is too big to fit on screen; displaying at 25%
Warning: Image is too big to fit on screen; displaying at 25%

Geometric Transforms

Rotation a mesh

```
theta=-30;
theta=deg2rad(theta);
R = [cos(theta) -sin(theta);
    sin(theta) cos(theta)];


[x,y] = meshgrid(1:64);
C = 0.*(x + y);

rot_x = zeros(64,64);
rot_y = zeros(64,64);
for i = 1:1:length(x)
    for j = 1:1:length(y)
        temp = R*[x(i,j);y(i,j)];
        rot_x(i,j) = temp(1);
        rot_y(i,j) = temp(2);
    end
end

figure;
surf(x,y,C);
hold on
surf(rot_x,rot_y,C);
hold off
view(-90,-90)
```



Now do it for the image

```
anthonygrayscale = rgb2gray(anthony);
[x,y] = meshgrid(1:256);
[u,v] = meshgrid(1:256);
rot_u = zeros(256,256);
rot_v = zeros(256,256);
for i = 1:1:length(u)
    for j = 1:1:length(v)
        temp = R*[u(i,j);v(i,j)];
        rot_u(i,j) = temp(1);
        rot_v(i,j) = temp(2);
    end
end

rot_anthony = interp2(u,v, im2double(anthonygrayscale), rot_u,
 rot_v, 'cubic');
figure;
imshow(rot_anthony);
```

Published with MATLAB® R2016b