CS663 Assignment 5

Question 5

The paper “An FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration” gives us the following equation: for two images, with one being a translated version of the other, having Fourier transforms and , we have the cross-power spectrum

Where is the amount by which image 2 is translated in space relative to image 1, and is the complex conjugate of . This quantity is sometimes called the phase correlation.

Observe that the inverse Fourier transform of is simply an impulse at , so we can simply read off the value of the displacement from the inverse Fourier transform of the cross power spectrum. Naturally, the spectrum of an impulse is the constant function.

Now in the question, we are given the displacement as . Now the negative value can be problematic since our space variables (especially in MATLAB) are always positive.

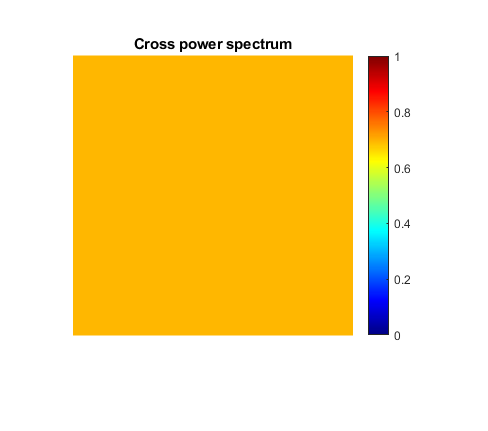
To solve this problem, we use the fact that the DFT is nothing but the Fourier series of a periodic repetition of the original signal. In other words, for a signal of length M, the M-point DFT itself will be periodic with fundamental period M. Extending this idea to images, we note that the displacements are equivalent for all integers *.*

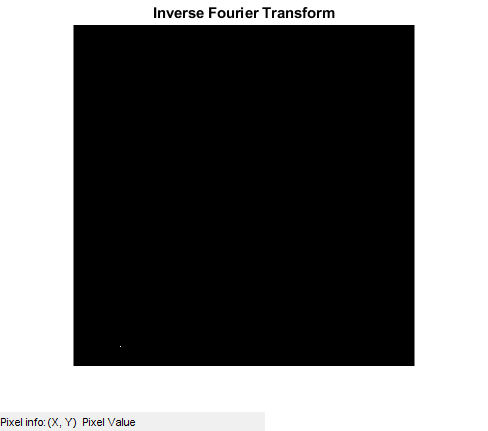
In MATLAB, to make use of this fact, provide as an argument to the function *fft2* a “size” argument such that the DFT computed follows the reasoning above.

Below is the code for the first case, for two images with white rectangles translated in the plane of the image relative to each other. We use *impixelinfo* to read off the displacement value from the inverse Fourier transform of the cross-power spectrum as shown. Note that we plot the log spectra.

# Without Noise

clear;  
clc;  
I = zeros(300);  
J = zeros(300);  
I(50:100,50:120) = 255;  
J(20:70,120:190) = 255;  
  
FI = fftshift(fft2(I,512,512));  
FJ = fftshift(fft2(J,512,512));  
  
FK = (conj(FI).\*FJ)./(abs(FI.\*FJ));  
lgFK = log(abs(FK)+1); figure(); imshow(lgFK); colormap('jet');title("Cross power spectrum"); colorbar;  
K = ifft2((ifftshift(FK)));  
figure; imshow(K/max(K(:)));title("Inverse Fourier Transform");  
impixelinfo;





The value read off comes out to be . Accounting for MATLAB’s 1-indexing, and the reasoning above, this matches the expected value: .

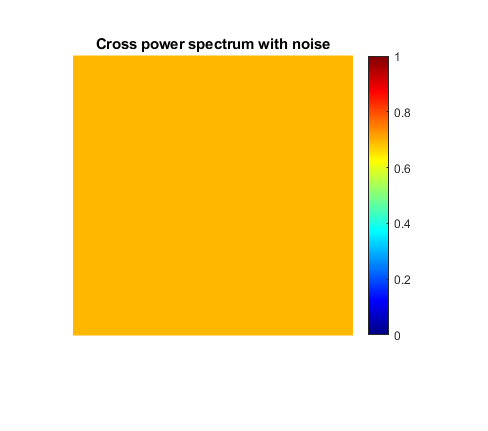
# With Noise

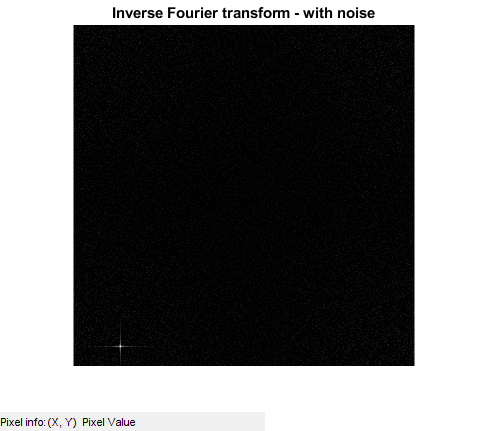
Now consider the case in which both images have IID gaussian noise of mean 0 and standard deviation 20. We expect results like the previous case.

The cross-power spectrum comes out to be a constant (approximately) like before.

Its inverse Fourier transform too looks like an impulse. The effect of the noise is evident – the signal is not a perfect impulse. Rather, it approximates one. However, we can still make out the peak to be at, giving us the net displacement between the two images as before.

In = I + random('norm',0,20,300,300);  
Jn = J + random('norm',0,20,300,300);  
  
FIn = fftshift(fft2(In,512,512));  
FJn = fftshift(fft2(Jn,512,512));  
  
FKn = (conj(FIn).\*FJn)./(abs(FIn.\*FJn));  
lgFKn = log(abs(FKn)+1); figure(); imshow(lgFKn); colormap('jet'); title("Cross power spectrum with noise");colorbar;  
Kn = ifft2((ifftshift(FKn)));  
figure; imshow(Kn/max(Kn(:)));title("Inverse Fourier transform - with noise");  
impixelinfo;





# Time Complexity

Let the images be of size . The operations required are computing forward and inverse Fourier transforms, all of time , and computing the cross-power spectrum, which requires time since it involves the Hadamard (pointwise) multiplication of the Fourier transform matrices.

Thus, the total time needed for this method is .  
Pixel-wise image comparison, on the other hand, will take, in the worst case, comparing every pixel of the first image with those of the second, giving a time complexity of .

# Correcting for Rotation

Assume we have two images – image 1 and image 2, with image 2 being a rotated version of image 1. If is the angle of rotation, and , are the Fourier transforms of the images, then by the Rotation theorem,

Thus, the spectrum of image 2 is itself a rotated version of the spectrum of image 1. Converting this to polar coordinates,

Now we can make use of Duality: consider and to be two new images in the coordinates and . is nothing but a translated version of in these coordinates. Thus, we can use our original method of phase correlation to find the displacement in the transform domain.

[*Published with MATLAB® R2020a*](https://www.mathworks.com/products/matlab)