Assignment 5: Discrete Fourier Transform

Tezan Sahu [170100035] & Siddharth Saha [170100025]

Due Date: 03/11/2019

Q6: FFT-Based Image Registration

1. Implementation on Images [Not Noisy]

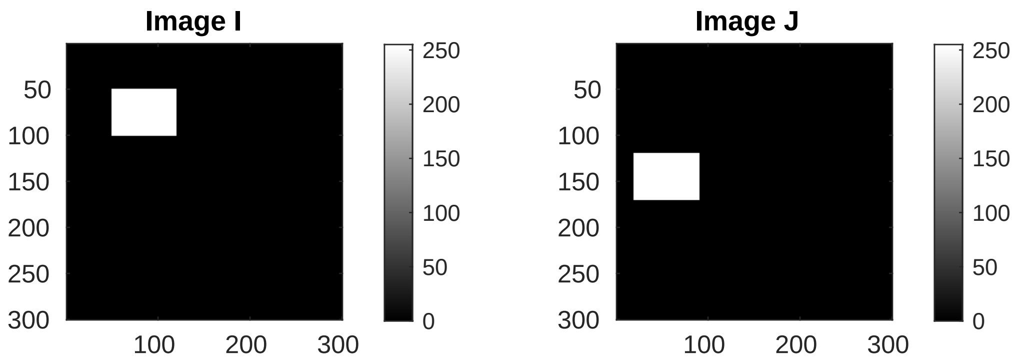

Fig 1: Images I & J [J is obtained by translation of rectangle in I by (t_x = -30, t_y = 70)]

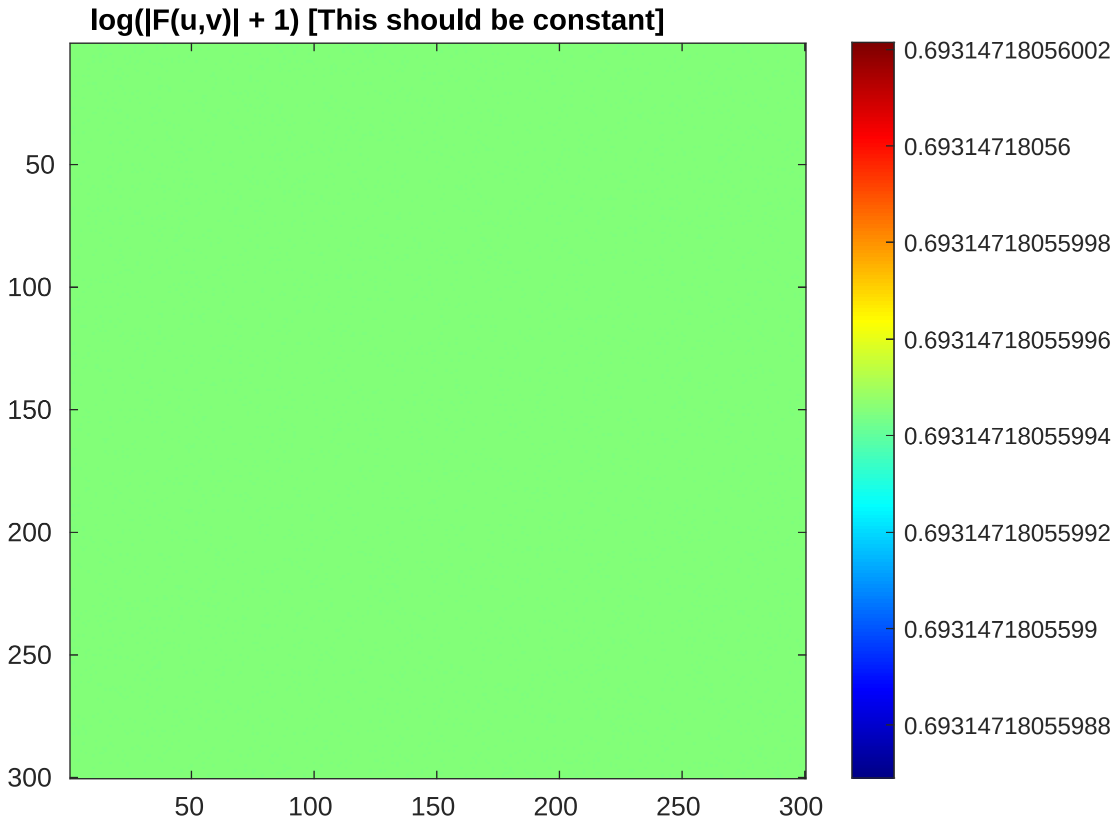

Fig 2: Logarithm of the Fourier magnitude of the cross-power spectrum

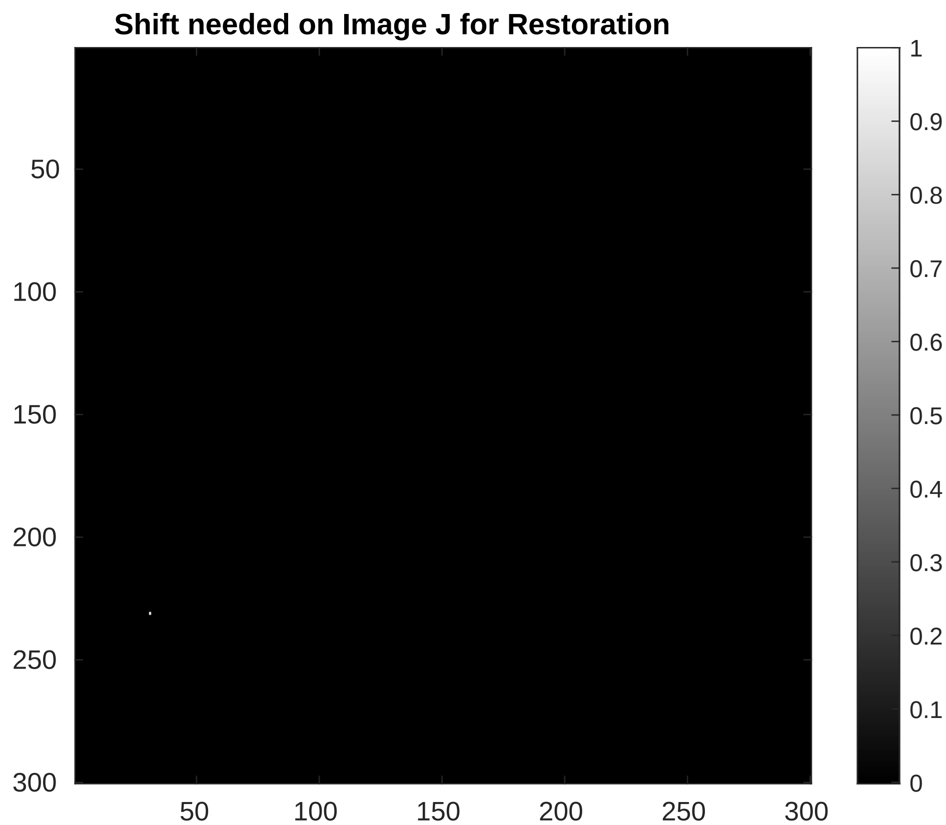

Fig 3: Shift needed on Image J for Restoration

2. Implementation on Images [Noisy]

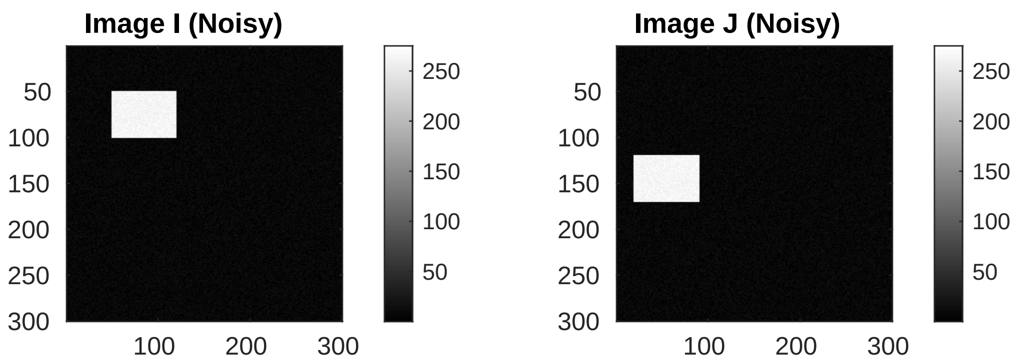

Fig 4: Noisy variants of images I & J [J is obtained by translation of rectangle in I by (t_x = -30, t_y = 70)]

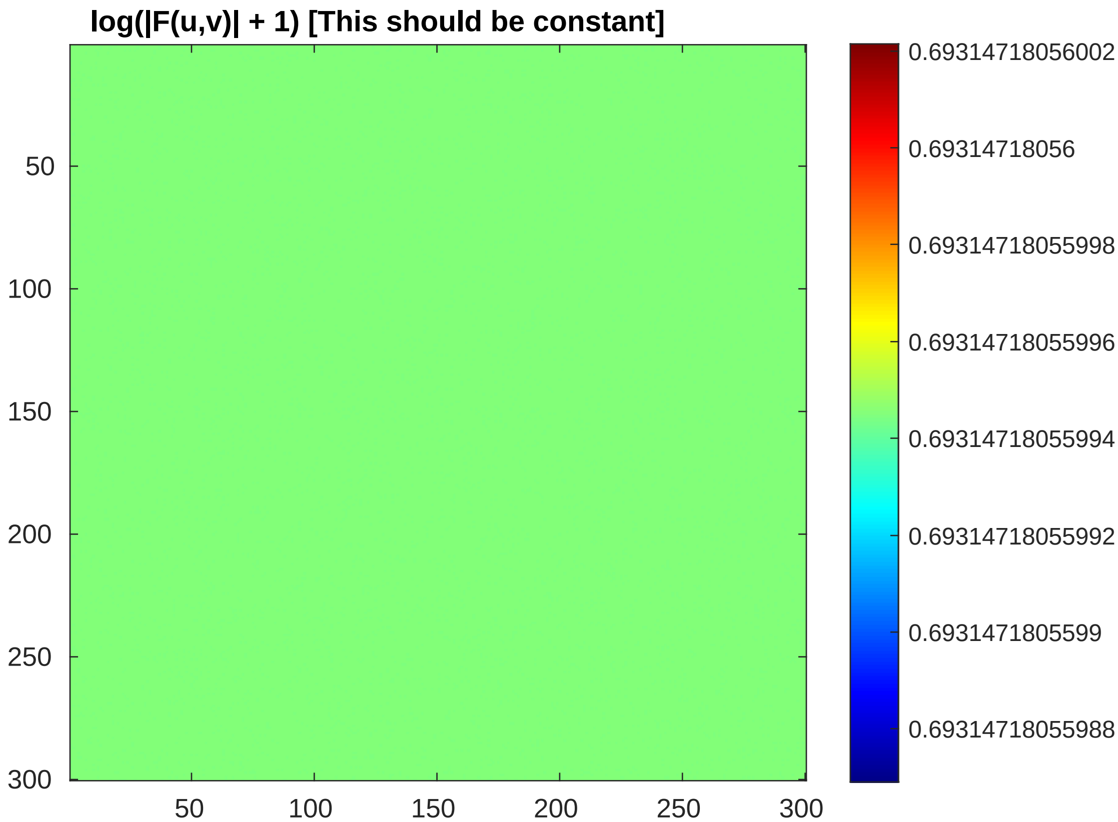

Fig 5: Logarithm of the Fourier magnitude of the cross-power spectrum

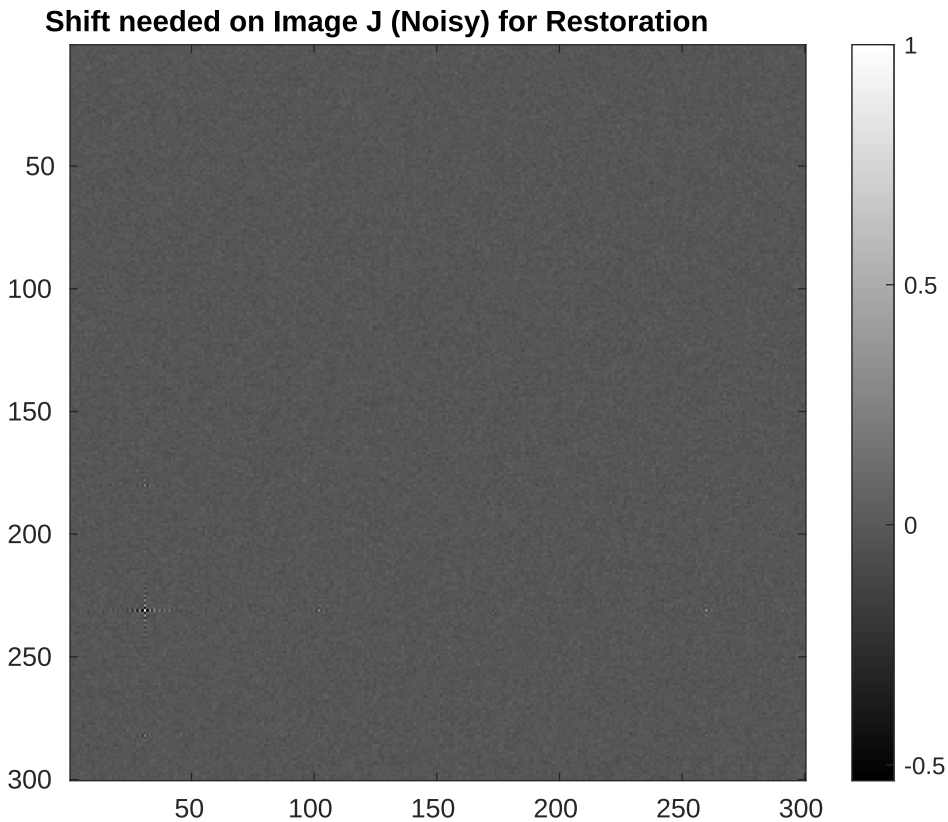

Fig 6: Shift needed on Noisy Image J for Restoration

Verification of result produced using Cross-Power Spectrum:

- Figure 3 shows a spike at (31, 231), which could be interpreted as as (31, -71) on applying a wrap-around on the image of size 300 * 300 while translation. This clearly is the translation to restore Image J back to Image I since the initial translation applied was (-30, 70). Note that the extra '1' is due to the MATLAB indexing.
- Similar to the previous case, figure 6 shows a spike at (31, 231), which could be interpreted as as (31, -71) on applying a wrap-around on the image of size 300 * 300 while translation. But this time, due to the noise present, the spike is not a clean spike, but surrounded by other frequencies of non-zero magnitude. Moreover, the relative magnitude of the spike w.r.t. surrounding region is not as high as the previous case compared
- We also see that the plots of logarithm of the Fourier magnitudes is a constant of value = log(2) because the result of the cross-power spectrum is a compex number of unit magnitude always.

Analysis of Time Complexities:

- For an Image of size N * N, using the cross-power spectrum to predict translation required for restoration involves the calculation of Fourier transforms using FFT [each being of O(N log(N))] followed by a conjugation [O(N)] & vectorized pointwise multiplication & division [O(1)]. Thus, the overall time complexity is O(N log N).
- If we use pixel-wise image comparison for an N * N image, the time complexity of predicting the translation would be O(N^2)

Approach for Correcting Rotation between Images:

[Here in the analysis, we consider correction if pure rotation of the image & ignore any translation or scaling]

If f2(x,y) is a rotated version of f1(x, y) [with a rotation of θo], doing a Fourier Transform in the cartesian coordinates would yield $F2(u, v) = F1(u\cos(\theta o) + v\sin(\theta o), -u\sin(\theta o) + v\cos(\theta o))$. Clearly, their magnitudes are the same. So, we can use the same concept of cross-power spectrum as before by converting the rotation by θo into a translation. This can be achieved by converting the images into polar coordinates & taking their Fourier Transform:

$f2(r, \theta) = f1(r, \theta-\theta o)$

$F2(m, n) = \exp(-2\pi j(n.\theta o)) * F1(m, n)$

Thus, cross-power spectrum of F1(m, n) & F2(m, n) would yield $\exp(2\pi j(n.\theta o))$, using which we can calculate the rotation.

Any translation in x & y would lead to a change in r by ro, such that the cross power spectrum would yield $\exp(2\pi j(m.ro + n.\theta o))$. Hence, displacement & rotation can be figured out. The exact (x, y) translations can be figured out using the original cross-power spectrum in the cartesian coordinates.