CS 663 - Fundamentals of Digital Image Processing
Assignment 05

Gagan Jain - 180100043
Hitesh Kandala - 180070023

November 16, 2020

1 Implementation Results

1.1 On images without noise

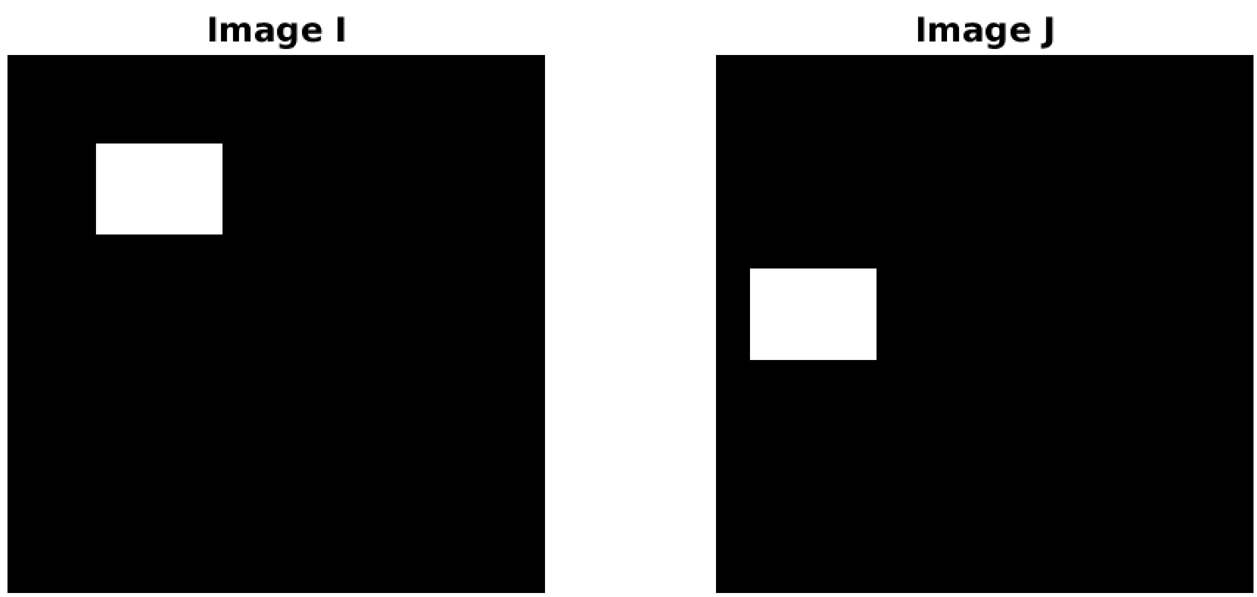

Figure 1: Image J is obtained by translation of rectangle in image I by $(t_x = -30, t_y = 70)$

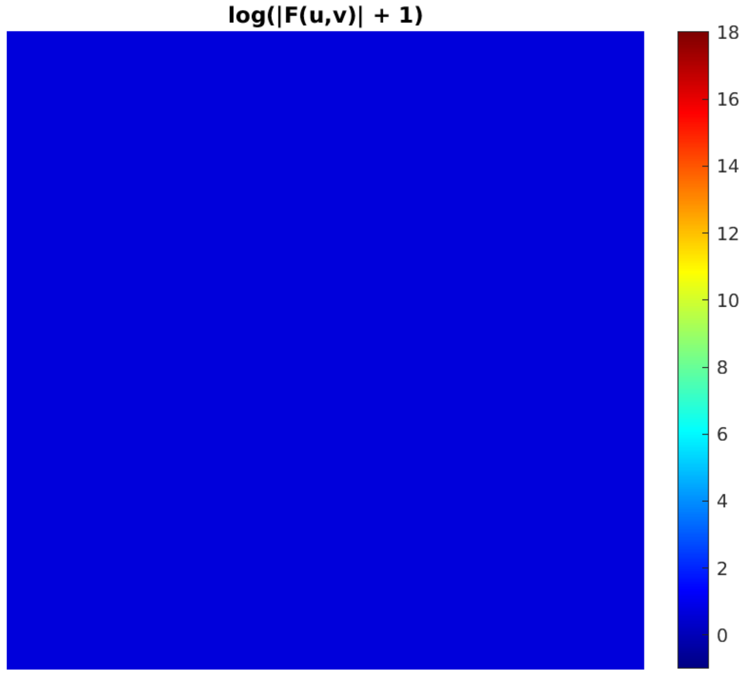

Figure 2: Logarithm of the Fourier magnitude of the cross-power spectrum (This is constant)

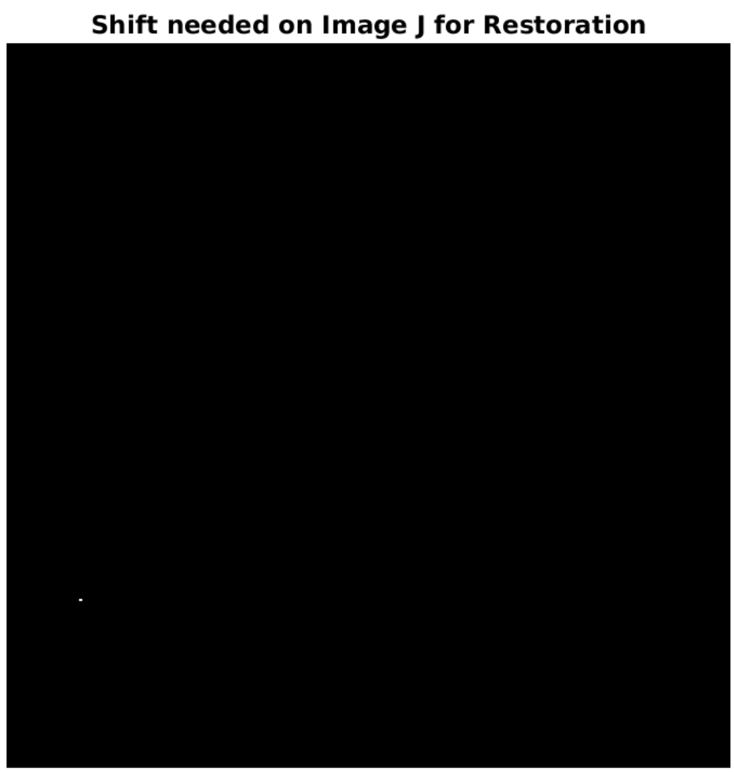

Figure 3: Shift needed on Image J for Restoration

1.2 On images with noise

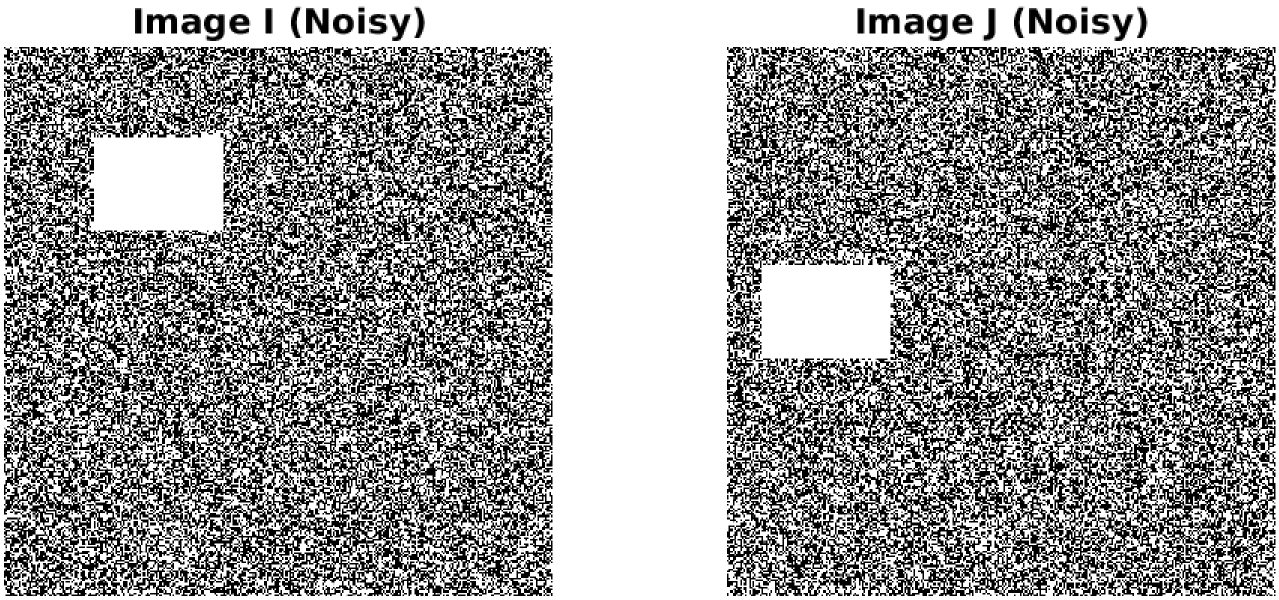

Figure 4: Images I and J with gaussian noise of zero mean and variance=20

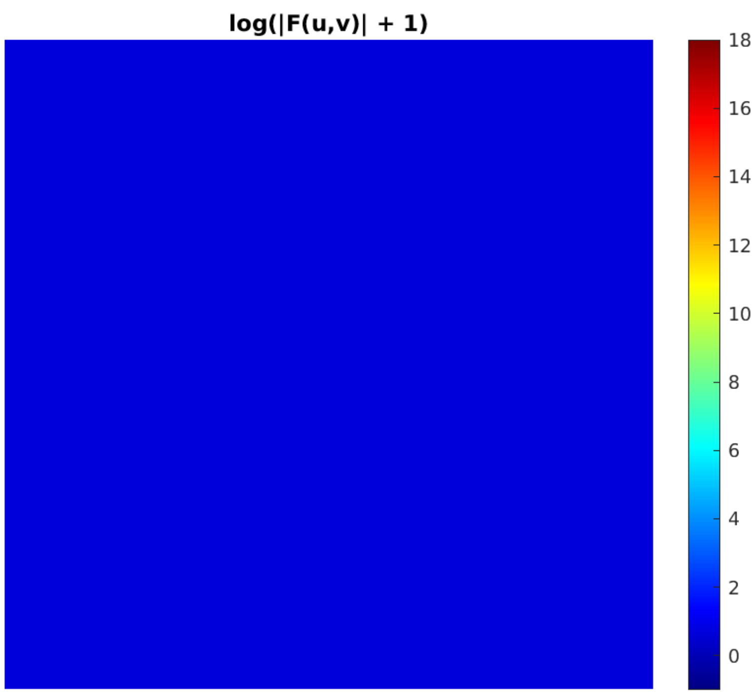

Figure 5: Logarithm of the Fourier magnitude of the cross-power spectrum (This is constant)

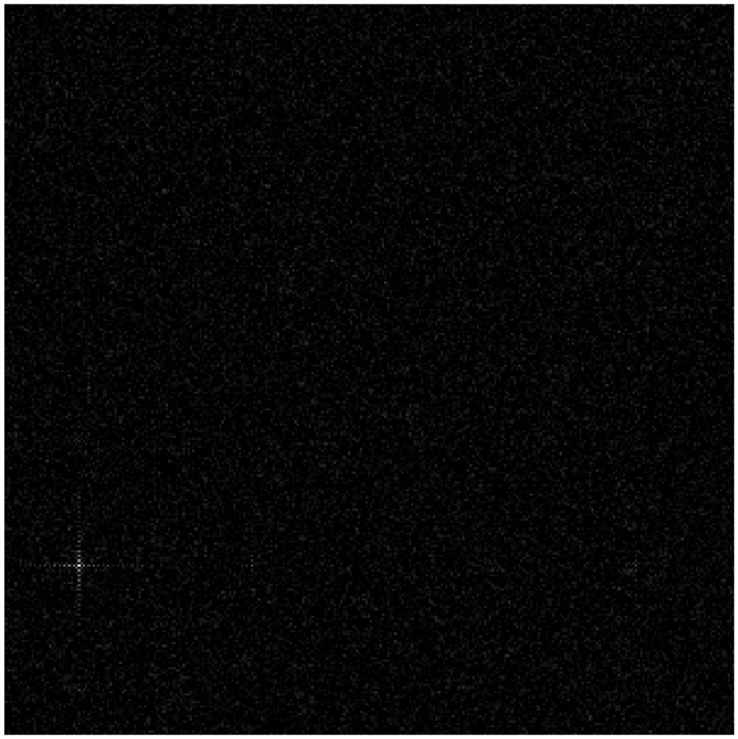

Figure 6: Shift needed on Noisy Image J for Restoration

2 Verification of result produced

1. Figure 3 shows a spike at (31.231), which could be interpreted 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).
2. 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.
3. 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.

3 Time Complexity Analysis

1. For an Image of size $N \times N$, 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, overall time complexity is O(N log N).
2. If we use pixel-wise image comparison for an $N \times N$ image, the time complexity of predicting the translation would be $O(N^2)$.

4 Approach for Correcting Rotation between Images

Note: Here in the analysis, we consider correction of pure rotation of the image and ignore any translation or scaling.

If $f_2(x, y)$ is a rotated version of $f_1(x, y)$ [with a rotation of θ_0], doing a Fourier Transform in the cartesian coordinates would yield $F_2(u, v) = F_1(ucos(\theta_0) + vsin(\theta_0), -usin(\theta_0) + vcos(\theta_0))$. Clearly, their magnitudes are the same. So, we can use the same concept of cross-power spectrum as before by converting the rotation by θ_0 into a translation. This can be achieved by converting the images into polar coordinates and taking their Fourier Transform:

$$f_2(r, \theta) = f_1(r, \theta - \theta_0) \tag{1}$$

$$F_2(m, n) = \exp -2\pi j(n\theta_0) * F_1(m, n) \tag{2}$$

Thus, cross-power spectrum of $F_1(m, n)$ and $F_2(m, n)$ would yield $exp(2\pi j(n\theta_0))$, using which we can calculate the rotation.

Any translation in x and y would lead to a change in r by r_o, such that the cross power spectrum would yield $exp(2\pi j(m.r_0 + n\theta_0))$. Hence, displacement and rotation can be figured out. The exact (x, y) translations can be figured out using the original cross-power spectrum in the cartesian coordinates.