Assignment 5 Question 6 Report

Sai Gaurav Anugole 170070008
Titas Chakraborty 170070019
Jayesh Choudhary 170070038

September 2019

1 Shift Detection with Normal Noise free Images

First of all, we generate the images I and J.

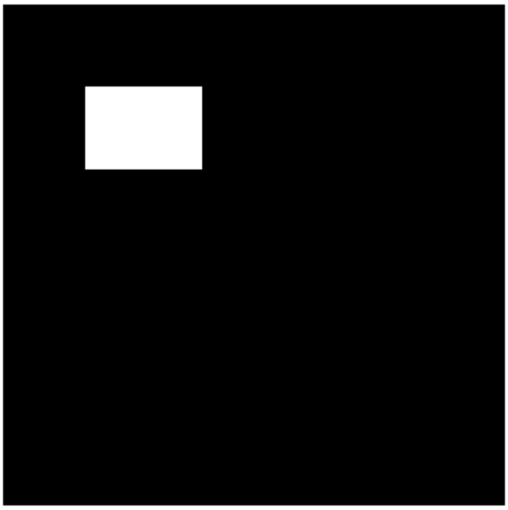

Figure 1: Image I

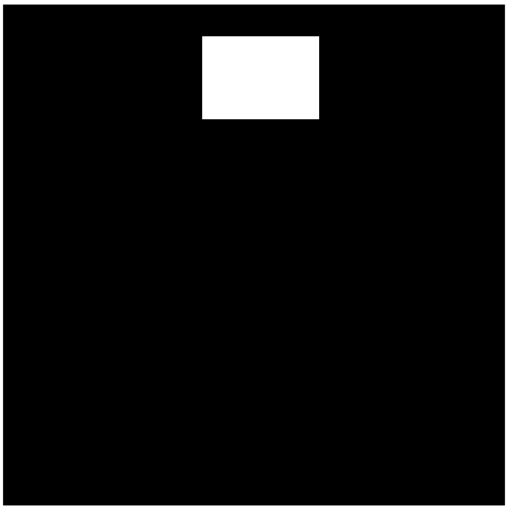

Figure 2: Image J

The shift is $t_x = -30$ and $t_y = 70$. Here for the image, X axis is directed from top to bottom while Y axis is directed from left to right.

- Here we compute the cross spectral density of the two images I and J.
- Then we invert the cross spectral density. The point with highest magnitude gives us the shift between the two images.
- An important thing to note is that in the formula for cross spectral density, the denominator terms became zero whenever $F1$ or $F2$ became 0 resulting in a NaN output. To avoid this, the denominator had a very small constant of 10^{-40} added to it.

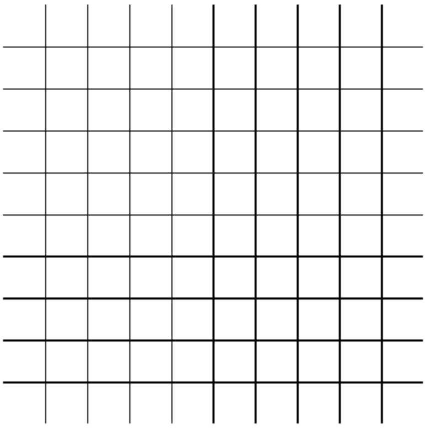

Figure 3: Cross Spectral Density

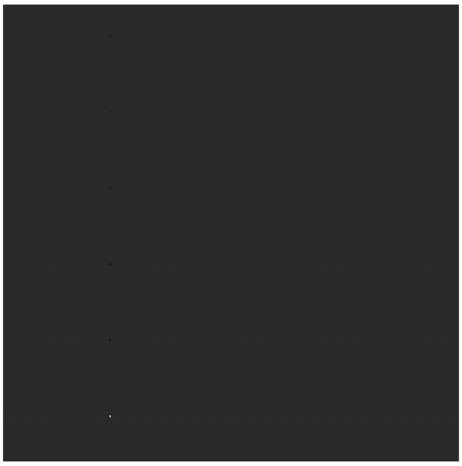

Figure 4: Inverse Fourier Transform of Cross Spectral Density

- We see the dot in the inverse Fourier Transform of the Cross Spectral Density at position (270,70) taking matrix indexing from 0.
- Since the image is of size 300 x 300, a displacement of 270 actually corresponds to a displacement of -30.
- Hence our method works extremely well.
- For the cross spectral density, ideally it should be 1 everywhere as the absolute value of $e^{j\Theta}$ is 1.
- However, the problem is that both $F1$ and $F2$ become 0 at intervals of 30.
- At that point, the cross spectral density cannot be calculated as it becomes a zero by zero form.

- To avoid that in our calculations, we added the denominator by 10^{-40} as mentioned before.
- Hence at every point where $F1$ and $F2$ become 0, the cross spectral density becomes 0 as the numerator becomes 0 while the denominator does not, leading to a grid like pattern being formed as $F1$ and $F2$ become 0 after intervals of 30.

2 Shift Detection with Noisy Images

- The steps followed are exactly the same as before expect for the fact that we do not need to use the factor of 10^{-40} in the denominator as in the previous case as $F1$ and $F2$ do not become 0 anywhere.

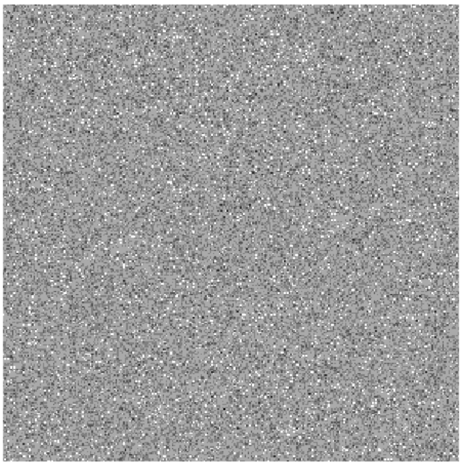

Figure 5: Cross Spectral Density

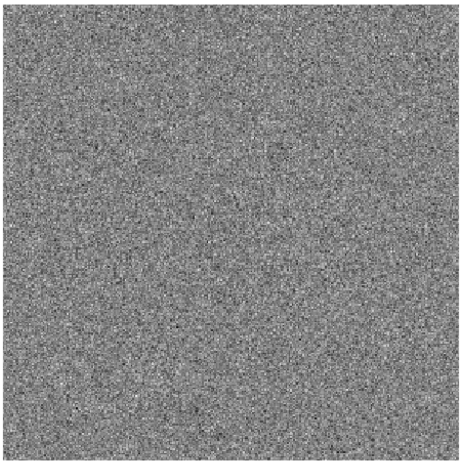

Figure 6: Inverse Fourier Transform of Cross Spectral Density

- The predictions are completely random and arbitrary in this case.
- There is no dot in the inverse Fourier transform of the cross spectral density. Instead the inverse Fourier transform is extremely noisy. So is the cross spectral density.
- Clearly this method is not usable even when there is a slight amount of noise in the images.

3 Time Complexity

- The time complexity of this procedure is $O(N^2 log(N))$. This is because of the computation of the Fourier and Inverse Fourier Transforms, which are both $O(N^2 log(N))$ operations. The computation of cross power spectral density is an $O(N^2)$ operation.
- On the other hand, pixel wise comparison for all shifts would be an $O(N^4)$ operation as the number of possible shifts are N^2 and pixel wise comparison for each shift is an $O(N^2)$ operation.
- Hence the time complexity of our procedure is much better than brute force pixel shifting.

4 Rotation detection in images

- In the paper, they have considered a general case of correction for rotation and translation together. Let us, for simplicity, consider correcting for rotation.
- Let us consider that $f_2(x, y)$ is a rotated version of $f_1(x, y)$ by angle θ_0.
- Hence f_2 will be equal to f_1 provided that one has rotated the axes by θ_0. Or f_1 will be equal to f_2 if one has rotated the axes by $-\theta_0$.
- So, $f_2(x, y) = f_1(xcos\theta_0 + ysin\theta_0, -xsin\theta_0 + ycos\theta_0)$
- If we take the Fourier Transform on both sides of the equation, we get that, $F_2(a, b) = F_1(acos\theta_0 + bsin\theta_0, -asin\theta_0 + bcos\theta_0)$
- If we convert the axes to polar co-ordinates, we get that, $F_1(r, \theta) = F_2(r, \theta - \theta_0)$
- This corresponds to a linear translation in polar co-ordinates. Hence we can find the value of θ_0 by finding out the cross spectral density and taking its inverse Fourier Transform.
- That is how we can correct for rotation in images.