Assignment 5 DFT: Question 6

We read the given paper *'An FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration'*. Equation 3 from the paper was implemented in Python 3 code to register 2 sets of images as described in the paper.

Implementation

The following function implements the algorithm.

```
def register_translation(a, b):
    assert a.shape == b.shape
    h, w = a.shape
    A = F.fft2(a)
    B = F.fft2(b)
    x_spec =  np.abs(F.ifft2((A * B.conjugate())/abs(A*B)))
    t_y, t_x = np.unravel_index(np.argmax(x_spec), (h, w))
    t_y = t_y-h if t_y > h//2 else t_y
    t_x = t_x-w if t_x > w//2 else t_x
    return t_x, t_y, x_spec
```

Registraion without noise

The results without noise was $t_x = -30, t_y = 70$ which is correct.

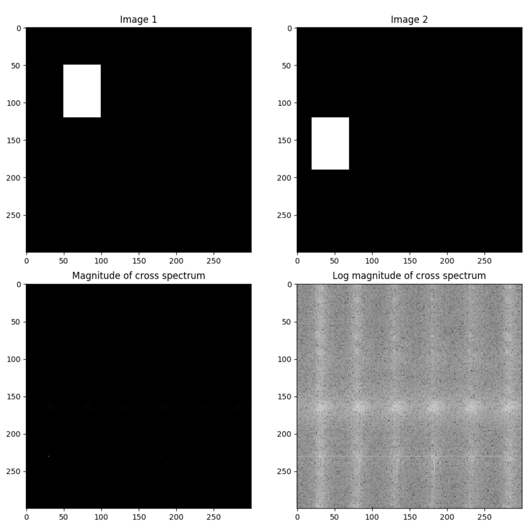

Figure 1: Result 1

Registraion with noise

The results obtained for the case with $N(0, 20)$ noise were $t_x = -30, t_y = 70$ which is correct.

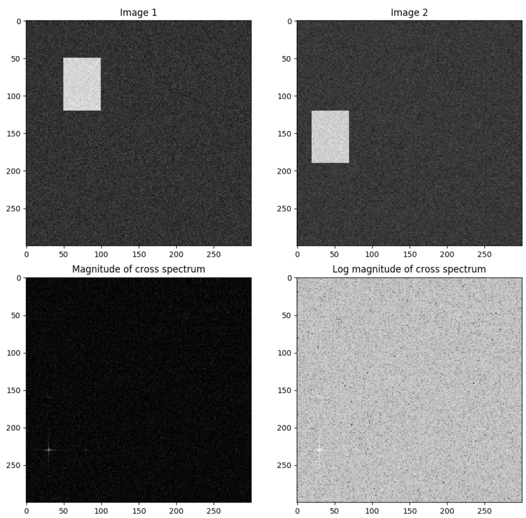

Figure 2: Result 2

Time Complexity

1D Discrete Time Fourier Transform on an array of length n takes $\mathcal{O}(n \log n)$ time with the FFT algorithm. For 2D FFT, we will first take DTFT row-wise and then column-wise (or vice versa), this will take $n \times \mathcal{O}(n \log n) + n \times \mathcal{O}(n \log n)$ time, which is ultimately $\mathcal{O}(n^2 \log n)$.

On the other hand, for brute-force pixel wise comparison, it would take $\mathcal{O}(n^4)$ time for comparison in each possible translation. There are n^2 possible translations. Therefore, the time taken for this would be $\mathcal{O}(n^4)$

Correcting Rotation

Assume image f_2 is a translated and rotated copy of f_1 with translation (x_0, y_0) and rotation θ_0.

$$f_2(x, y) = f_1(x \cos\theta_0 + y \sin\theta_0 - x_0, -x \sin\theta_0 + y \cos\theta_0 - y_0)$$

In frequency domain, the translation by (x_0, y_0) results in a phase change equivalent to multiplication with $e^{-j2\pi(\xi x_0 + \eta y_0)}$. But the magnitude is unchanged.

So, $M_1 = M_2$ holds where M_1, M_2 are magnitudes of F_1, F_2. Now, we change the coordinate system to polar coordinates.

$$M_1(\rho, \theta) = M_2(\rho, \theta - \theta_0)$$

Now, the problem has switched into finding the phase correlation which is a translation in θ space which is already solved. However, if there is a scale change between f_1 and f_2 too, then we switch to logarithmic scale where scaling reduces to simple translation. We start with

$$M_1(\rho, \theta) = M_2(\rho/a, \theta - \theta_0)$$

Then we convert to logarithmic scale to get

$$M_1(\log \rho, \theta) = M_2(\log \rho - \log a, \theta - \theta_0)$$
$$M_1(\xi, \theta) = M_2(\xi - d, \theta - \theta_0)$$

where

$$\xi = \log \rho, d = \log a$$

This has again reduced to an image registration with only transation, where the images is magnitudes of F_1 and F_2, which is a solved problem.