CP467 - Assignment 2 - Useful Image Operations

(Fourier Demonstrated)

Spencer Bellerose

160795820

bell5820@mylaurier.ca

In class we talked about many different image operations, for my example I have chosen to display the Fourier Transformation. The Fourier Transformation decomposes a function into frequencies, this is useful for many things, especially recognizing characters. In other words, this transformation is moving things from the time domain to the frequency domain.

The next useful operation we talked about is the Wavelet transformation, which uses multiresolution analysis, which involves analyzing a signal into multiple different frequencies at different resolutions. Other applications of this type of transformation are in data and image compression.

The third operation I will mention is that of Histogram Equilization. Using this method, we can adjust constrast using the images histogram. This operation will provide us with a better quality image without risk of the loss of information.

The fourth operation I will mention is Grey Level Transformation. The goal of image enhancements, in this case grey level transformation, is to modify an image so that its utilization on a particular application is enhanced. This transformation is operated on individual pixel intensity values, and is therefore a data independent pixel transformation. This method uses 4 main approaches, these are: 1. Image Negatives. 2. Log Transformation. 3. Power Law Transformation. 4. and Piece-wise transformation.

Our final operation is Scaling, scaling is by far one of our most used operations in image processing. Scaling is used to resize an image, and is therefore required to perform lots of other operations, this has been shown with edge detection.

Here are the libraries and imports I will use to demonstrate Fourier

In [10]:

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
```

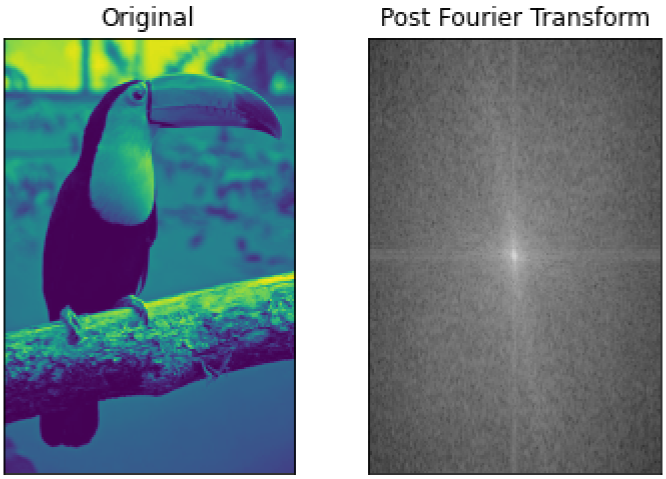
Below we have a python implementation of a Fourier Transformation (using OpenCV). As we can see the original image of the Toucan is represented in the frequency domain after the transformation.

In [9]:

```
img = cv2.imread('test1.png',0)
f = np.fft.fft2(img)
fshift = np.fft.fftshift(f)
fourier_t = 20*np.log(np.abs(fshift)) #Fourier Shift Using Numpy

#plots original image
plt.subplot(121),plt.imshow(img)
plt.title('Original'), plt.xticks([]), plt.yticks([])

#plots fourier transformed image
plt.subplot(122),plt.imshow(fourier_t, cmap = 'gray') #shown in greyscale
plt.title('Post Fourier Transform'), plt.xticks([]), plt.yticks([])
plt.show()
```



On the Left is the Original image, on the Right is the Post-Fourier Image. The Left is representative of the Time Based Domain, where as the "Post-Fourier Transform" image is representative of frequency based domain of the same image