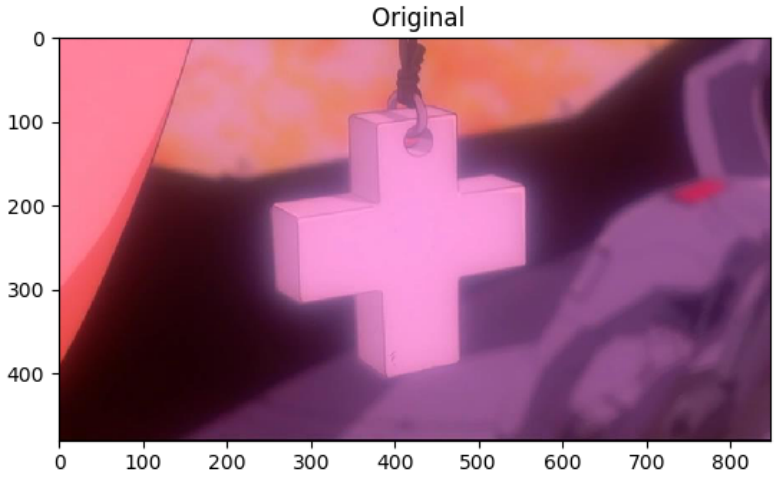
Fourier Transformation and application of filter with Python in Image processing

Lihao Wang

2018 Dec

1 Introduction

This project is intend to use Fourier Transform as a basic theory to perform comprasion with not just only Gray scale images import but also RGB colored images.
With the knowledge from the lecture, the labs, the computational simulation techniques and the method can be found outside the classroom, another different way to approach the final goal is developed and used, which would be detailed explained and described in this report. As a sample, I picked an image from my own collection as a sample usage.



2 Physics Background

Discrete 2D Fourier Transform [1]

In lab5, we were introduced Discrete Fourier Transform(DFT) and Fast Fourier Transform(FFT). However, in image processing, we will need a 2-D Fourier Transform, which is:

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux/M + vy/N)} \tag{1}$$

$$f(x,y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F(u,y) e^{j2\pi(ux+vy)} \tag{2}$$

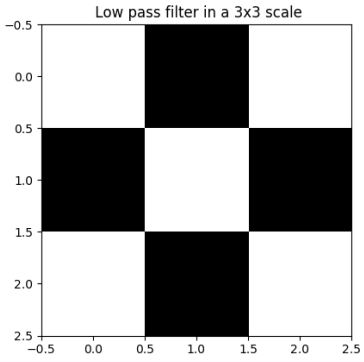
Such that the image can be described as in Time Domain & Frequency Domain, where can analysis the Spectral density.

Filter

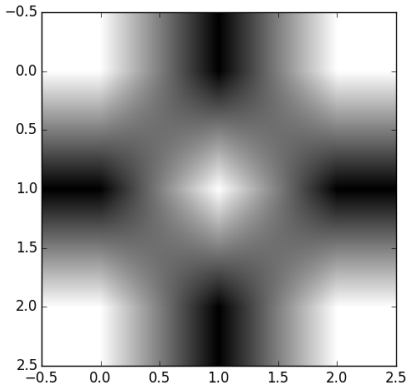
As normally described by 'Noise Filter' and 'Wave Filter', for Image processing, the common used 'Wave Filter' is mostly picked as Low-pass filter and High-pass filter in this simulation.

Low-pass A low-pass filter is an electronic filter only passes signal lower than a certain chosen frequency cut-off. For signal above the cut-off, attenuation would be applied on to the part of signal.

High-pass A high-pass filter is an electronic filter only passes signal higher than a certain chosen frequency cut-off.For signal below the cut-off, attenuation would be applied on to the part of signal.



When the filter is tranformed into Frequency Domain, it looks like below:



Band-pass A Band-pass filter is a comnbination of Low-pass filter and high-pass filter where Low-pass filter's cut-off is higher than the cut-off from High-pass.In this case, the Band-pass filter only passes signal in the range of both cut-off, which is the method to filter out the frequency by need.

3 Computational Background

Filter building with Python [4]

Building a filter in Python is quite easy compare to building a actual one physical.For example, a 3×3 low-pass filter can be easily defined as a 3×3 matrix as

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

for a 3×3 2-D signal wave to present filtering operation.

Filtering requrirement

As in this project we intend to use Fourier Transform as the basic support from Python environment,as the numpy package's fft2 function marks the input as a wave with just one period of Fourier Transform, which can be also treated as equivalent to 1D Fourier Transform tow times on both x-axis and y-axis.Such that, the output from fft2 would center the high frequency signal and place all low frequency signal on the edge.Obviously, in most kind images can reach, low frequency signal dominates. As a result like that, to proceed into the filter, the dominated frequency signal must be centerd in the frequency domain.

RGB images and process

Compare to gray scale diagram, in Python environment, RGB diagrams performs in a 5-D matrix,which have extra R G B axis on the z row of the matrix.
However, using a different way of importing the same RGB image can produce another totally different result matrix.

Method \ Order	z=0	z=1	z=2
cv2.imread	B	G	R
PIL.Image.open	R	G	B
matplotlib.image.imread	R	G	B
scipy.misc.imageio.imread	R	G	B
skimage.io.imread	R	G	B

(3)

Clearly, we can see except opencv package, the other mostly used method all saves with the same order RGB while opencv goes with BGR.

Method	Data type
cv2.imread	uint8 matrix
PIL.Image.open	Original format
matplotlib.image.imread	uint8 array
scipy.misc.imageio.imread	uint8 matrix
skimage.io.imread	uint8 array

(4)

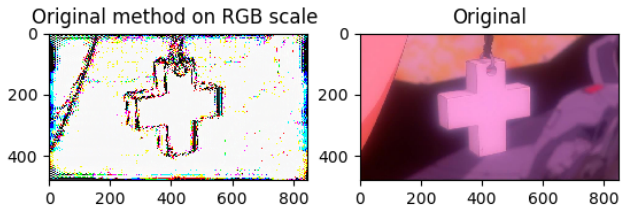
At the same time, for the container created by each method, PIL standed out and made some difference from all other four, which requires more actions and operations while actual programming.

4 Actual programming

Original method similar to lectured

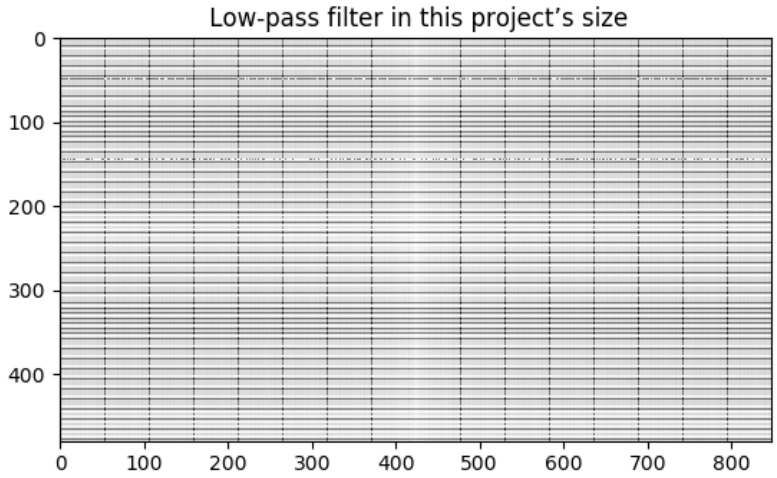
By lecture with PHY407, the most commonly used is to perform Fourier Transform of the target data set: either an array or a certain matrix, than disgard the low certain percentage of the coefficient results(For example, set 95% of the 1D result coefficient array to 0).However, in a 2D RGB input, as RGB constructed with three layers, which means the sampled operation should apply to all three layers.
Rivetingly,due to all three layers are from the same source, although each state in frequency domain performs relatively similar with each other, with the instructed method, as we disgarding the values, the operation itself slightly influencing the feature and eigenstate in frequency domain and make all three smoothened into more different.

As a result of above situation, when all three layers are decentrilzed and transform back into Time Domain, the stacked up final compressed image output still keeps identifiable details,shapes and lines, but totally lost the right color.

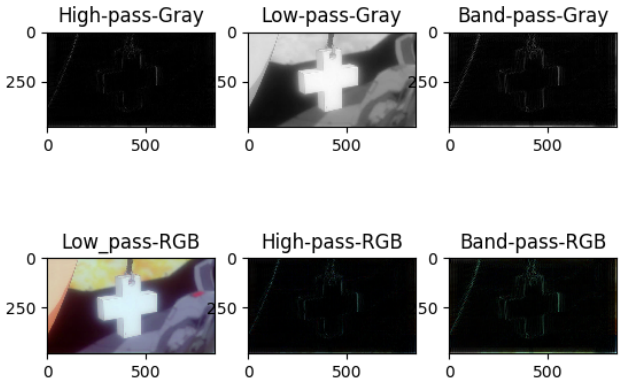


The Filter method comes up and ... [2]

As searching more resourse and material,some other methods like equalizer, it can be easily found not only in industrial but also in labs, filter are mostly common-used method to treat a given wave. Such that, if all images can be transformed into frquency domain in the form as wave. To approach that, Fourier Transform is the best method that have ever learned about that.
So as a starting approach from the method to a script, the different loading way gives a different input. After several trials,I kept with using the matplotlib which gives uint8 matrix [(4)has more details of each method].[3]
Similar to Lab 5, we start with a fast Fourier transform with the inputed matrix, and for certain filter usage, also perform the phase shift to pass through the filter and tranform back to Time domain.[2]However, as a complex output, Pyton does not have the ability to plot a complex matrix. Such that, we take the absolute value of the result.
As what has been taught among the semaster, we should be able to say we have finished the job. Whereas we still need to adjest of these digits to make it easier for python to plot. The most reason of this step is the filters, for example, the Low pass filter looks like:



By defining all processes as a function for efficiency, and apply all RGB layers has been treated, stack back causes a lots of problem:using float32 and float64 format, output looks OK but rather than compressing the image, if the filter window is about 1/4 of original image size, it looks like being sharpened. If the window drop down to 1/10 of the original image, the output is a total black image. Same kind of problems when using tuple which even lose right color when plotting. As a result, the best way is to convert to int with in $0 \sim 255$ range.
After all above , with all layer stacked and gray scale diagram are shown as below:



5 Conclusion

By this project, we learned the usage of filter with in Frequency domain and clearly, High pass keep the lines and main features of the diagram at the same time Low pass keep the most color with blur/compress the lines and the features of diagram. Although Band pass have the feature of both filter, it's extremely depend on the window(the range of the cutoff).
Inversely, we also have the Laplacian operator to sharpen the image. Which is an operator in Time domain to operat as a high pass filter or an inverse high pass filter depending on how it is set up.

6 After the project report

Thanks for reading this report to the end(I hope someone did), thanks again for such a great course we can have this year, I know there's still several filter out there, for example, Guassian Filter, Mean Filter, Median Filter and so on, but I choose to not proceed too far this time. As the most target has already completed among the progress.However, I found that finding out these filter and developing , testing them is really something enjoyable. Just as a suggestion, probably that is a good assignment for next year student as some of the student might be from engineer or studied/studying signal analysis, that's really helpful.Thanks again for making such a great course. orz

References

[1] Heinrich,Fourier Transform Insight
https://zhuanlan.zhihu.com/p/19763358

[2] AInewworld,Filter use in Frequency and Fourier Transform
https://blog.csdn.net/on2way/article/details/46981825

[3] Signal operation with Python
http://bigsec.net/b52/scipydoc/frequency_process.html

[4] Frequency Domain Process with Python
https://www.kancloud.cn/wizardforcel/hyry-studio-scipy/129098