Fourier Transforms and Images

Nathan Conger

Introduction

In the prior paper, it looked at the seemingly unrelated connection between frequencies, such as light and sound waves, and breaking RSA encryptions through Fourier transforms. The equation allowed us to be able to breakdown complex waves into its constituent components. The natural application of this equation is with sound and light. The colors that we see all around are from light waves ricocheting off an object and hitting our eye. In computer science, we've managed to capture color scene by representing the color in 2D pixel space. How can we use the Fourier Transform to analyze and maybe even modify this pixel space?

Extending the Fourier Transform

To make things simpler, we'll first look at greyscale images (i.e. an M x N matrix A where $0 \leq A_{i,j} \leq 1$ where 0 represents a black square and 1 represents a white square). We need a way to be able to map the image space into a corresponding Fourier space. Luckily, the Fourier transform is easily extendable into 2D. Since pixel space is discrete, we can use that form of the transform $F: (u, v) \rightarrow \mathbb{C}$:

$$F(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-i2\pi(\frac{u}{M}x+\frac{v}{N}y)}$$

where $f(x, y)$ represents an entry in the image space A. An image can be represented as a sum of bases, where each base is represented by a series of alternating white and black bars across the pixel space. The length of each gap represents the frequency, whereas the angling of the bars represents the base. The original image is just a weighted sum of all of these bases. Thus, our output is just the average sum of all of these bases at each entry of the image matrix A. The Fourier transform is invertible with inverse function:

$$f(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{i2\pi(\frac{u}{M}x+\frac{v}{N}y)}$$

The Fourier Space

The Fourier space is an identically sized M x N matrix of complex numbers, which cannot be easily visualized. Thus, we represent it in two different forms: Magnitude $|F(u,v)|$ and Phase Angle $\varphi(u,v)$. The most commonly used representation of the Fourier space is the Magnitude $|F(u,v)|$ form. The magnitude represents how much each basis contributes to the brightness of the whole image. The bases are represented by an angle and a frequency, with higher frequencies the further away from the center of the matrix. However, most of the image encoding are represented in the higher frequencies. Additionally, we can centralize the higher frequencies by shifting the Fourier space such that *F(0,0)* (the average of every point in the image space) is in the center of the grid. Fortunately, since our basis is repeatable in both x and y directions, our Fourier space is only one chunk in a periodic space. Thus, we can translate our output such that our high frequencies lie in the center.

The Magnitude space is much more easily analyzed than the Phase Angle space. The seemingly random assortment of values encodes where to place each contribution found measured in Magnitude. Calculating the Magnitude and Phase of two different images of identical size and rebuilding each with the other's Phase shows the brightness of both images are distorted, but their images are swapped.

Properties of 2D Fourier

The 2D Fourier transform has specific properties that will help in modifying an image.

- Rotation preserved: Rotating an image by some angle θ_0 is preserved after transformation
$$x = r\cos\theta\,,\ \ y = r\sin\theta\,, u = \omega\cos\varphi\,, v = \omega\sin\varphi$$
$$f(r, \theta + \theta_0) \leftrightarrow F(\omega, \varphi + \ \theta_0)$$
- Convolution Theorem: An image that is convolved in some form (through blurring like with a long exposure *image*, or atmosphere, etc) can be fixed by multiplication in the fourier space. $F(f \otimes\ h) = F(f) * F(h)$ where

  $(f \otimes\ h)(x, y) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(x, y)h(x - m, y - n)$ is the convolution operation.

Gaussian Filtering

As we learned, the pipeline of modifying an image is to first apply the Fourier transform *F(u,v)*, centralize *F(0,0)*, apply some modification function *h(x,y)*, undo the centralization, and then apply the inverse Fourier *f(x,y)*. There are many functions that can be applied to modify the image, but these are a few basic filters; Ideal, Butterworth, and Gaussian filtering. Each of the three filters have low pass (removing lower frequency) and high pass (removing higher frequencies) forms. We'll look only at the Gaussian filter.

The low pass Gaussian filter removes any noise that may be present in an image. The formula that does this is

$$H(u, v) = e^{-D^2(u,v)/2D_0^2}$$

where $D^2(u, v)$ is the square of the distance between point (u, v) and the center of the image and D_0 is some positive constant. This filters out the lower frequencies from the image which cause some of the noise. Letting D_0 be small causes the image to blur and remove more noise.

The high pass Gaussian filter sharpens the edges. It cuts out the high frequencies which hold most of the image brightness information and shows only the edges when inverting the transform. The equation that does this is

$$H(u, v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

The high pass is useful in sharpening an image. An image's sharpness relates to the intensity of its edges and the Gaussian high pass finds them. Applying some sort of intensifying function along the found edges sharpens the image.

Image Convolution

One of the Fourier properties has to do with convolution. A convolution describes how the shape of one function is affected by another. Convolution usually occurs when taking pictures like image blurring from long camera exposure and the atmosphere in the photographed area. Convolution is usually done using some kernel, a square matrix smaller than the image, and applying it in the image space for each pixel. This process is rather complicated, but can be easily done as a result of the Convolution Theorem (In the Properties Section). This allows us to be able to represent convolution as multiplication in Fourier space, which is much easier to do than regular convolution. This makes using the Fourier space the easiest way to deconvolve an image and restore it into its original form.

Applications and More Research

As high filtered images show the edges of objects, it can be used in machine learning for object identification. The shape of defined edges tends to be preserved in the Fourier space. Since each letter has distinct shape, and the semi preservation of edges, the transform of letters can be represented matched like fingerprint. The high filter can also be used in the medical field in creating clearer pictures from medical imaging, making it easier for doctors and nurses to detect any abnormalities a patient may have.

The pixel space of a color image is more slightly more complex than greyscale. Each pixel in a colored image uses RGB color values unlike the singular brightness value for greyscale images. One method to applying color images is to separate the image into R, G, and B image components and apply the Fourier transformation to each component individually. There is new research where an image doesn't have to be separated by representing pixel color as a quaternion. Then using an extended quaternion Fourier transform, the whole color image can be transformed into its respective Fourier space. There are limitations of using quaternions such as losing the property of commutativity.

References

Chen, Craig. "Digital Image Processing Using Fourier Transform in Python." *Medium*, Medium, 17 Feb. 2020, medium.com/@hicraigchen/digital-image-processing-using-fourier-transform-in-python-bcb49424fd82.

Dogra, Ayush, and Parvinder Bhalla. "Image Sharpening By Gaussian And Butterworth High Pass Filter." *Biomedical and Pharmacology Journal*, vol. 7, no. 2, 2014, pp. 707–713., doi:10.13005/bpj/545.

Fisher, Robert, et al. "Fourier Transform." *Image Transforms - Fourier Transform*, Hypermedia Image Processing Reference, 2003, homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm.

Spatial Frequency Domain, www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic1.htm.

Wang, Yao. "2-D Fourier Transforms." Polytechnic University.