Discrete Fourier Transform

Computer Vision

Mehrad Haghshenas

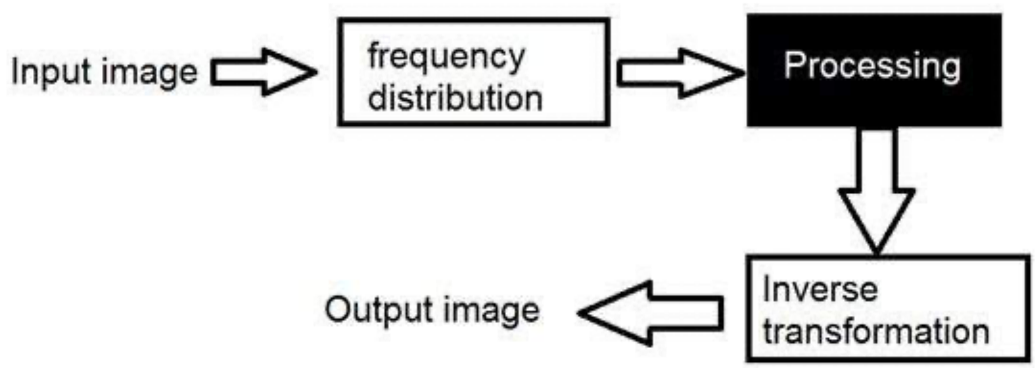
Images can be presented in two domains: the spatial domain and the frequency domain. This research studies and analyzes images in the frequency domain. In the frequency domain, we don't analyze signals based on time but with respect to frequency. Furthermore, we have the original image and intensity values in the spatial domain. Whereas in the frequency domain, we have the rate at which the pixel values change in the spatial domain.

There are two transformations to convert an image from the spatial domain to the frequency domain: 1- Discrete Cosine Transform. 2- Discrete Fourier Transform. In this project, we focus on the Discrete Fourier Transformation on grey-scale images. We will first mention the importance of the frequency domain. We will then note the mathematical background of the Fourier Series and Fourier Transformations and subsequently introduce the principles of Discrete Fourier Transformation. Accordingly, we aim to show how to receive the frequency domain of an image from the spatial domain of the picture. Ultimately, we will mention the applications and properties of the frequency domain.

1- Why the frequency Domain

The question arises that why we want to convert our images from the spatial domain to the frequency domain. Some concepts and techniques in image processing are easier to describe in the frequency domain compared to the spatial domain. Likewise, the frequency domain analysis provides insights into the initial image. We first Map our image using DFT; accordingly, changes in the frequency domain are called Fourier filter operations. The inverse DFT then maps the modified Fourier transform back into the modified image. This process allows us to enhance contrast (high frequencies), remove noises, and smooth images (low frequencies). Now that we know the purpose of the frequency domain, how do we map the image to this domain?

2- How to map images in the frequency domain



Images consist of pixels which are atomic elements of images. Each pixel has an intensity value, and images are initially represented as intensity values in the spatial domain; that is, as (x, y, u) which x and y indicate the pixel's location and u is the value. A signal can be mapped into the frequency domain using different kinds of transformation, such as Fourier Series, Fourier transformation, Laplace transform, Z transform. We will use DFT to map images from the spatial domain to the frequency domain in this research.

3- What is DFT?

We will first look at the mathematical background of DFT and then define it.

3.1 Fourier Series

Fourier transformation can be denoted as a kind of coordinate transformation. Fourier invented the Fourier Series in early 1800, which until 1870 was not popular amongst mathematicians and was not even translated to English. Accordingly, Fourier showed that each periodic function could be approximated as the sum of sines and cosines of increasing frequencies. Specifically, for any periodic function $f(x)$ with $T = 2\pi$ we have:

$$f(x) = \left(\frac{A0}{2}\right) + \sum_{x=1}^{\infty} A_k \cos(kx) + B_k \sin(kx)$$

$$\frac{A0}{2}\, is\ the\ DC, A_k\ and\ B_k\ are\ Fourier\ Series\ coeffiecients.$$

$$A_k = \left(\frac{1}{\pi}\right) \int_{-\pi}^{\pi} f(x) \cos(kx) dx$$

$$B_k = \left(\frac{1}{\pi}\right) \int_{-\pi}^{\pi} f(x) \sin(kx) dx$$

A_k and B_k are calculated by multiplying both sides of the initial equation by $cos\ (kx)$ and $sin\ (kx)$ respectively and calculating the integrals of both sides. We mentioned before that that Fourier transformation is like a coordinate transformation. In this case, the orthogonal vectors are like sine and cosine (in the cartesian system it was x-y) and the Fourier series is the sum of the projection in the cosine direction and sine direction. If $T = L$, then the equations above would slightly change as the following: inside the cosine and sine we would have $2\pi ki/l$ instead of kx, and for the coefficients, instead of $\frac{1}{\pi}$ we would have $\frac{2}{l}$.

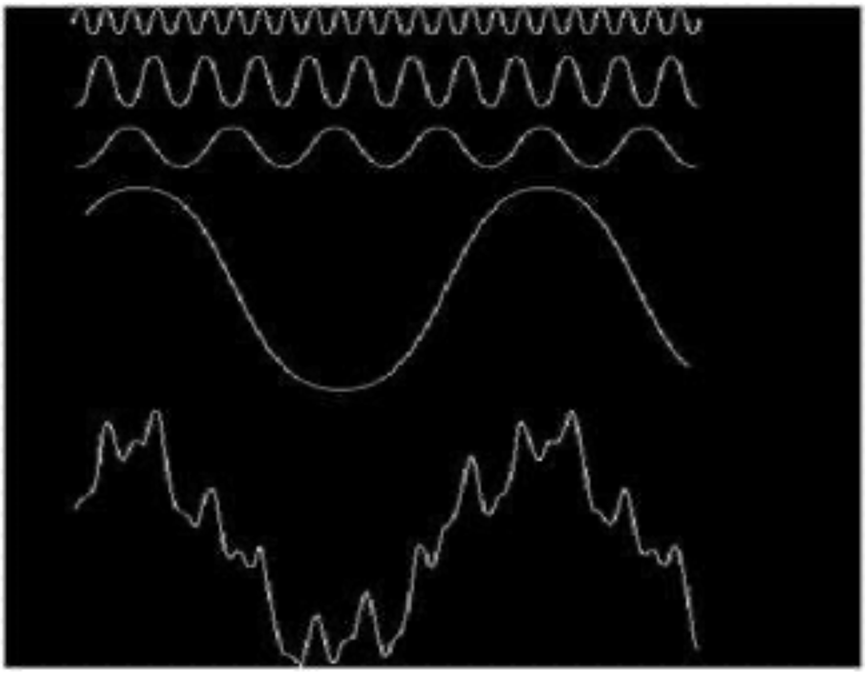
3.2 Fourier Transformation

It is very similar to the Fourier series, but instead of having periodic functions on a definite domain, $f(x)$ has no limitations. The Fourier transformation of $f(x)$ is:

$$F(f(x)) = \int_{-\infty}^{\infty} f(x)e^{-i\omega x}dx$$

Note that $exp(i\alpha) = e^{i\alpha} = cos\alpha + i \cdot sin\alpha.$

The reverse of the transformation would be as: $F^{-1}(f(\omega)) = (\frac{1}{2\pi}) \int_{-\infty}^{\infty} f(\omega)e^{i\omega x}d\omega$



The above image shows the idea of Fourier transformation; the last signal is the sum of the signals shaped as sines and cosines.

3.3 Discrete Fourier Transformation

In discrete Fourier transformation, we have a discrete sample of data. Take, for instance, n dots in the one-dimensional domain:

$$f0, f1, f2, f3, f4, \ldots, fn-1 \ then\ under\ DFT\ they\ will\ turn\ to\ F0, F1, F2, \ldots, Fn-1$$

$$Fn = \sum_{j=0}^{n-1} fn * \ e^{-\frac{i2\pi jk}{n}}, \quad w_{n=}e^{-\frac{i2\pi jk}{n}}$$

1D Fourier filtering is commonly used in audio processing and 2D Fourier filtering of images follows the same principles. Given that our focus in the following sections is on Discrete Transformation Fourier in the two-dimensional domain, we will give the latter equation, and the inverse in R^2.

$$
\begin{aligned}
G(m,n) &= \frac{1}{\sqrt{MN}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} g(u,v) \cdot e^{-\mathrm{i}2\pi \frac{mu}{M}} \cdot e^{-\mathrm{i}2\pi \frac{nv}{N}} \\
&= \frac{1}{\sqrt{MN}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} g(u,v) \cdot e^{-\mathrm{i}2\pi (\frac{mu}{M} + \frac{nv}{N})}
\end{aligned}
$$

$$
\begin{aligned}
g(u,v) &= \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} G(m,n) \cdot e^{\mathrm{i}2\pi \frac{mu}{M}} \cdot e^{\mathrm{i}2\pi \frac{nv}{N}} \\
&= \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} G(m,n) \cdot e^{\mathrm{i}2\pi (\frac{mu}{M} + \frac{nv}{N})}
\end{aligned}
$$

As we can see the equations above, are only a simple extension of the equation in R. Transform Equations of DFT vary, but the important concept is that the Discrete Fourier transform always maps an image from the spatial domain into the frequency domain. In order to apply a two-dimensional discrete Fourier transform to an intensity image, we can either directly implement the formula for 2D or simplify the formula by representing it as a sequence of 1D discrete Fourier transformations. That is, applying the Fourier transform to each row vector of the image and replacing the original value with the result value. Then applying it to each column vector of the image.

Some tips about the frequency domain and spatial domain: 1- both domains have the same dimension. Because the transformation is in place and each pixel is transformed to a unique u,v. 2- To receive the frequency domain from the spatial domain apply the first equation in section 3.3. We must apply this on each pixel that is all u and v; m and n are the corresponding location of the pixel in the frequency domain. M and N are the dimensions of the image. 3- Applying the latter equation will map the image from the frequency domain to the spatial domain. (Inverse of Fourier) 4- Usually, the Fourier image is shifted in such a way that the DC-value is displayed in the center of the image.

We can derive two images in the frequency domain, the magnitude image, and the phase image. The phase image does not yield much new information about the structure of the spatial domain image. However, we must keep the information given that reconstructing the image in the spatial domain requires pixel phases. Most of the information of the spatial domain can be seen in the magnitude image in the frequency domain.

Final Note: FFT (Fast Fourier Transformation) is a fast version of DFT. In DFT we must calculate two loops for all pixel values and thus the complexity operation would be $O(N^{\wedge}2)$. However, in FFT, we first add pixel values 0 such that the number of pixels will become a power of 2. If we divide the pixel values into even and odd indexed pixels (like the pixel values in which $x + y$ are odd and even) and repeat the process, then O($N\ log\ N$) will be the complexity which is much better. (For detailed information on the mathematical background of FFT the following link can be useful. However, most programming languages have a built-in FFT function: https://ocw.mit.edu/courses/mechanical-engineering/2-161-signal-processing-continuous-and-discrete-fall-2008/study-materials/fft.pdf). FFT is the heart of audio recording and image processing processes.

Now that we know how to map the image, we shall mention some of the applications and properties of the frequency domain.

4- Frequency Domain properties & Applications.

4.1 Properties

Shift in x: The spectrum for $\mathrm{f}(x - d)$ would be $e^{-i\omega d}\mathrm{F}(\omega)$.

Shift in frequency: If The spectrum for $e^{-i\omega_0 x}\mathrm{f}(x)$ would be $\mathrm{F}(\omega+\omega_0)$. Multiplying in the spatial domain by some real value affects only the amplitude of a sinusoid. While multiplying by some complex value c modifies the function's phase and amplitude (if |c| = 1 then amplitude does not change).

Convolution: Convoluting two functions $\mathrm{f}(x)$ and $\mathrm{g}(x)$ corresponds to multiplying the two functions in the frequency domain. Therefore, multiplying is much easier than calculating the convolution, making the frequency domain efficient in this regard of applying filters.

Rotation: If an image rotates by a degree α, the spectrum also rotates by the same angle.

4.2 Applications:

The Fourier Transform is applied in image filtering, image reconstruction, and image compression. (We talked about image reconstruction. Here we will talk about filtering and compression.)

High-frequency information in the frequency domain indicates an edge in the corresponding image in the spatial domain. Therefore, the frequency domain can be used for edge detection, which has the most crucial information of an image.

Low-frequency information in the frequency domain indicates a smoothing image in the spatial domain. Therefore, we can alter the frequency domain by filters to smooth the initial image.

Frequency filters are applied to an image in the frequency domain. The image is Fourier transformed and multiplied by the filter function and then re-transformed. (In the spatial domain, if we wanted to apply the filter, we had to convolute the filter and the image).

How is the frequency domain used for image compression? An image into the frequency domain is indicated by a complex number (with a real and imaginary component) per pixel. Converting an image to find the lowest amplitude frequencies and cancel them out is one way to compress images. Therefore, it'll take fewer data to describe the frequency content of an image than the picture's pixels.

Resources:

W. Burger and M. Burge, Principles of Digital Image Processing. Volume 2: Core Algorithms.

R. Klette, Concise Computer Vision – An Introduction into Theory and Algorithms, Springer Undergraduate Topics in Computer Science, London, 2014.

https://www.youtube.com/watch?v=jNC0jxb0OxE&list=PLMrJAkhIeNNT_Xh3Oy0Y4LTj0Oxo8GqsC&index=1

https://www.youtube.com/watch?v=MB6XGQWLV04&list=PLMrJAkhIeNNT_Xh3Oy0Y4LTj0Oxo8GqsC&index=3

https://www.youtube.com/watch?v=Ud9Xtxsi2HI&list=PLMrJAkhIeNNT_Xh3Oy0Y4LTj0Oxo8GqsC&index=4

https://www.youtube.com/watch?v=jVYs-GTqm5U&list=PLMrJAkhIeNNT_Xh3Oy0Y4LTj0Oxo8GqsC&index=11

https://www.youtube.com/watch?v=E8HeD-MUrjY&list=PLMrJAkhIeNNT_Xh3Oy0Y4LTj0Oxo8GqsC&index=18

https://blog.demofox.org/2020/11/04/frequency-domain-image-compression-and-filtering/#:~:text=When%20you%20transform%20an%20image,go%20into%20making%20the%20image.

https://www.tutorialspoint.com/dip/fourier_series_and_transform.htm

https://homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm