Unit -3

Image enhancement in frequency domain

****

Knocking Back on:

Image Enhancement in Spatial Domain and Frequency Domain

Image enhancement is required in many different digital domains, but sometimes these technicalities are covered up by powerful editing software and other tools that have become an everyday part of the business. But learning how the many different image enhancement techniques work within the different digital domains will serve as a tremendous benefit to you and your teams.
Here this is going to dive deep into the image enhancement techniques used in the spatial and frequency domains. Without getting too technical, let's take a closer look at these two areas of interest.

Difference Between Enhancements in the Spatial Domain and the Frequency Domain

The biggest difference is that when we're dealing with images in the spatial domain, we're essentially dealing with the images as they are. Their pixel values might change, for example, but only concerning the settings and the scene. However, in the frequency domain, the rate in which the pixels are changing values is one of the focal points. Here

Spatial Domain:
Input -> Image Processing -> Output

Frequency Domain:
Frequency input -> Distribution -> Image Processing -> Inverse Transformation -> Frequency Output

Without getting too technical in these areas, we'll shift our focus onto the image enhancement side of this. So, let's look at how image enhancement occurs in the spatial and frequency domains.

Image Enhancement in the Spatial Domain

The spatial domain is used to define the actual spatial coordinates of pixels within an image, so when we use this term in the image enhancement business, we're talking about things like equalization, smoothing, and sharpening. Here are a few examples:

Histogram Equalization

This is a common image enhancement technique that looks to improve on the overall appearance of an image. Imagine that you have a picture that's a bit too dark, quite common in image enhancement. Well, the fact that it's too dark tells us that its spatial alignment is on the lower side of the greyscale. So, stretching out those grey levels would create an image that's clearer because it creates a more uniform distribution.
The same is true of an image that is too bright. Its balance might be tilted to a specific brighter color, so equalizing the darker areas of the image to match would make it clearer.

Image Smoothing

The primary purpose of image smoothing is to help alleviate the symptoms that cameras can cause camera noise, as well as spurious and missing pixel values. There are several different types of smoothing, so we'll focus on a process known as neighborhood averaging.
A camera can sometimes cause the edges of an image to be blurred, the result of higher band frequencies in the image being diminished because of lighting or other environmental conditions. So, experts will use a professional tactic known as median filtering to set the grey levels of those areas to the same pixel values in the same *neighborhood* (close to) those of those pixels.

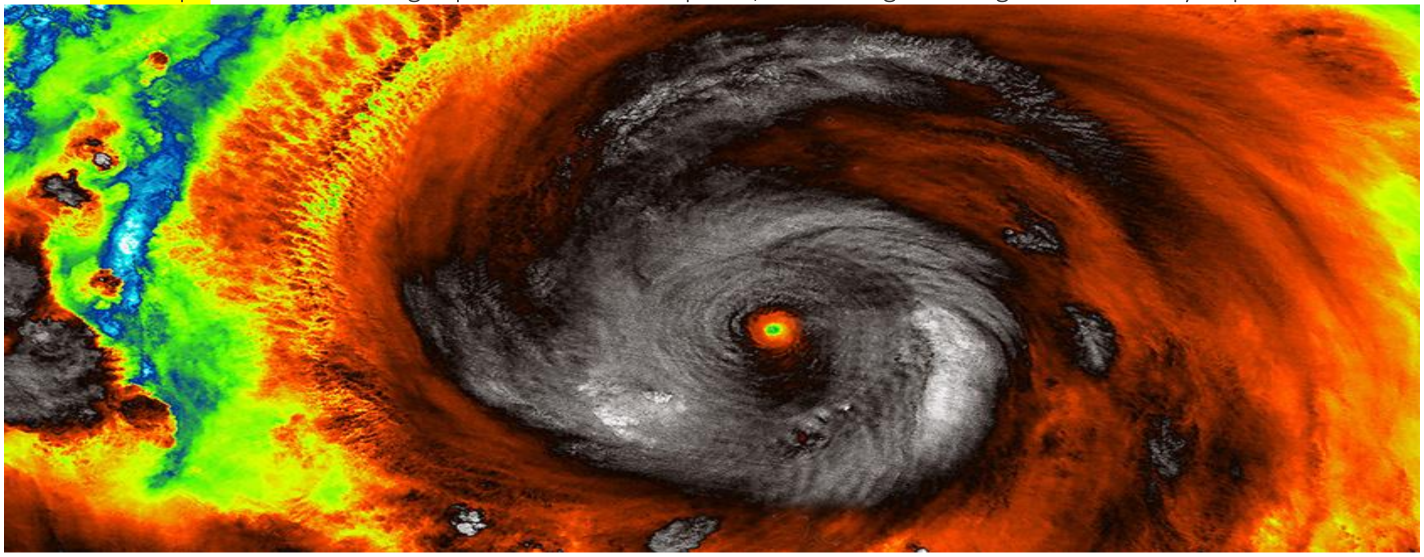
Image Enhancement in the Frequency Domain

Images edited in this way will be transferred into the frequency domain, where it's possible to work on the spectrum itself. Image enhancement occurs in the Fourier transform of the image and, from an editing perspective, deals with blurring, sharpening, contrast, and the distribution of greys. However, the result is that the pixel values are also going to intensify as the transformation occurs.

The most common image types that require this type of editing are satellite and medical images. Let's look at a few examples.

Low-Pass Filtering (Blurring) Image Enhancement

The edges and transitions in an image's greyscale values contribute significantly to the content of an image found in the higher frequencies. In layman's terms, low-pass filtering will make an image blurrier. There are several reasons why you might need to have a blurrier image. First of all, even the highest quality cameras in the world leave behind a certain amount of noise. So, blurring will help eliminate it.
Different pixels will be affected by this noise in different ways since they all generate it differently. So, using a low-pass filter can eliminate a lot of noise while only having a minimal impact on the image. For instance, an image taken from a telescope will have its image spread over a lot of pixels, so blurring the image would mostly impact the noise.



High-Pass Filtering (Sharpening) Image Enhancement

A high-pass filter is used to sharpen an image and is another common form of frequency-domain image enhancement. This method focuses on the finer details in the image and does the exact opposite of a low-pass filter.

Unfortunately, that also means that it enhances the noise as well, the price that's extracted to sharped up the content.

That tells us that this image enhancement method should only be used if the image contains minimal noise, otherwise sharpening it will make it much worse. It would look grainy and unnatural, so be careful that you don't overdo it.

Enhancing a Normal Image to an Amazing One

The truth is that image enhancements can turn an average photo into an amazing one, but if done improperly, it could end up ruining the image. That's why it's important to bring in professionals to help get this done correctly. At Smart Photo Editors, we have dedicated ourselves to building teams that come equipped with knowledge and experience with image enhancement in every industry, so we are able to handle image enhancement in the spatial and frequency domains.

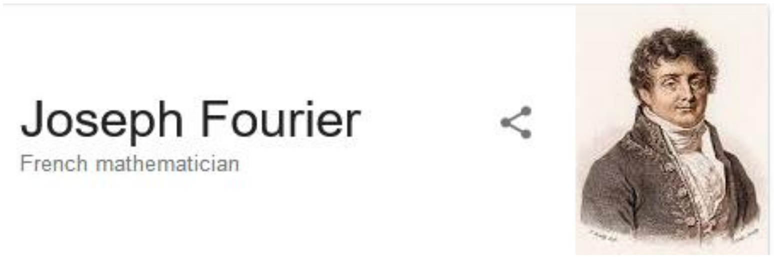
So, if you have a photo that needs to be sharpened but you don't want to risk it being grainy, or if you need some of its edges smoothed out, then let us transform your average photo into a breathtaking masterpiece

.

Image enhancements are done in two domain

1. Spatial or time domain
2. Frequency or Fourier domain

Now we are processing **signals** (images) in ***frequency domain***. Since this **Fourier series** and Frequency domain is purely mathematics, so we will try to minimize that math's part and focus more on its use in DIP.



Frequency domain analysis:

Till now, all the domains in which we have analyzed a signal, we analyze it with respect to **time**. But in frequency domain **we don't** analyze signal with respect to time, but with respect of **frequency.**

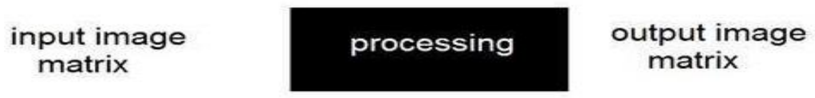
Difference between spatial domain and frequency domain

In **spatial domain**, we deal with images as it is. The value of the pixels of the image change with respect to scene.

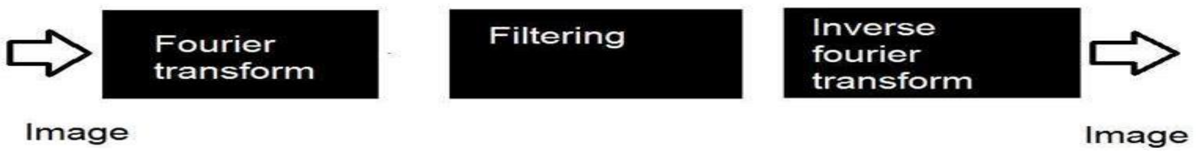
Whereas in **frequency domain**, we deal with the **rate** at which the pixel values are changing in spatial domain.

For simplicity, Let's put it this way.

Spatial domain, In simple spatial domain, we directly deal with the image matrix.



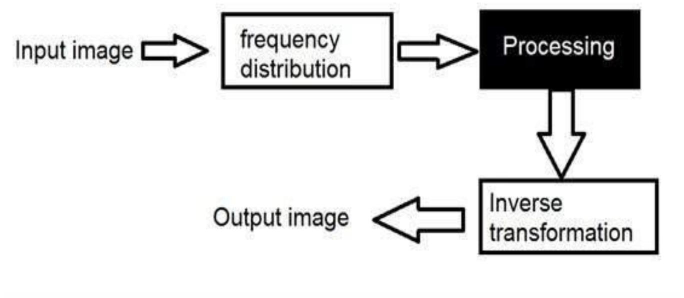
Frequency domain, we deal an image like this.



Frequency Domain

We first transform the image to its frequency distribution.

Then our black box system performs whatever processing it has to performed, and the output of the black box in this case is not an image, but a transformation.



After performing inverse transformation, it is converted into an image which is then viewed in spatial domain.

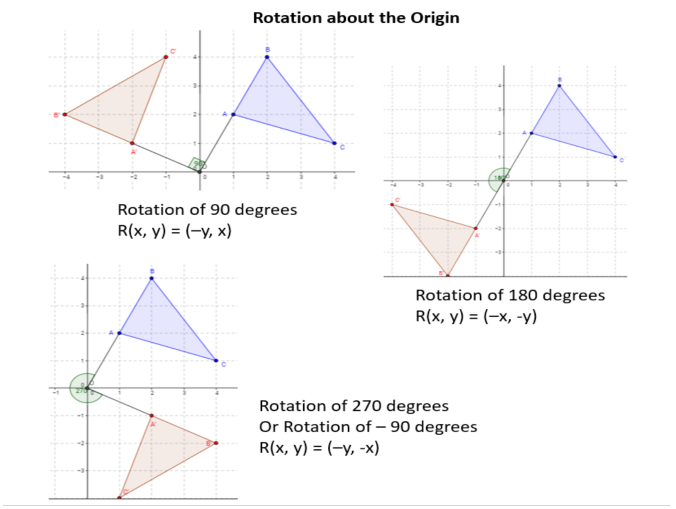
Here we have used the word transformation. **What does it actually mean?**

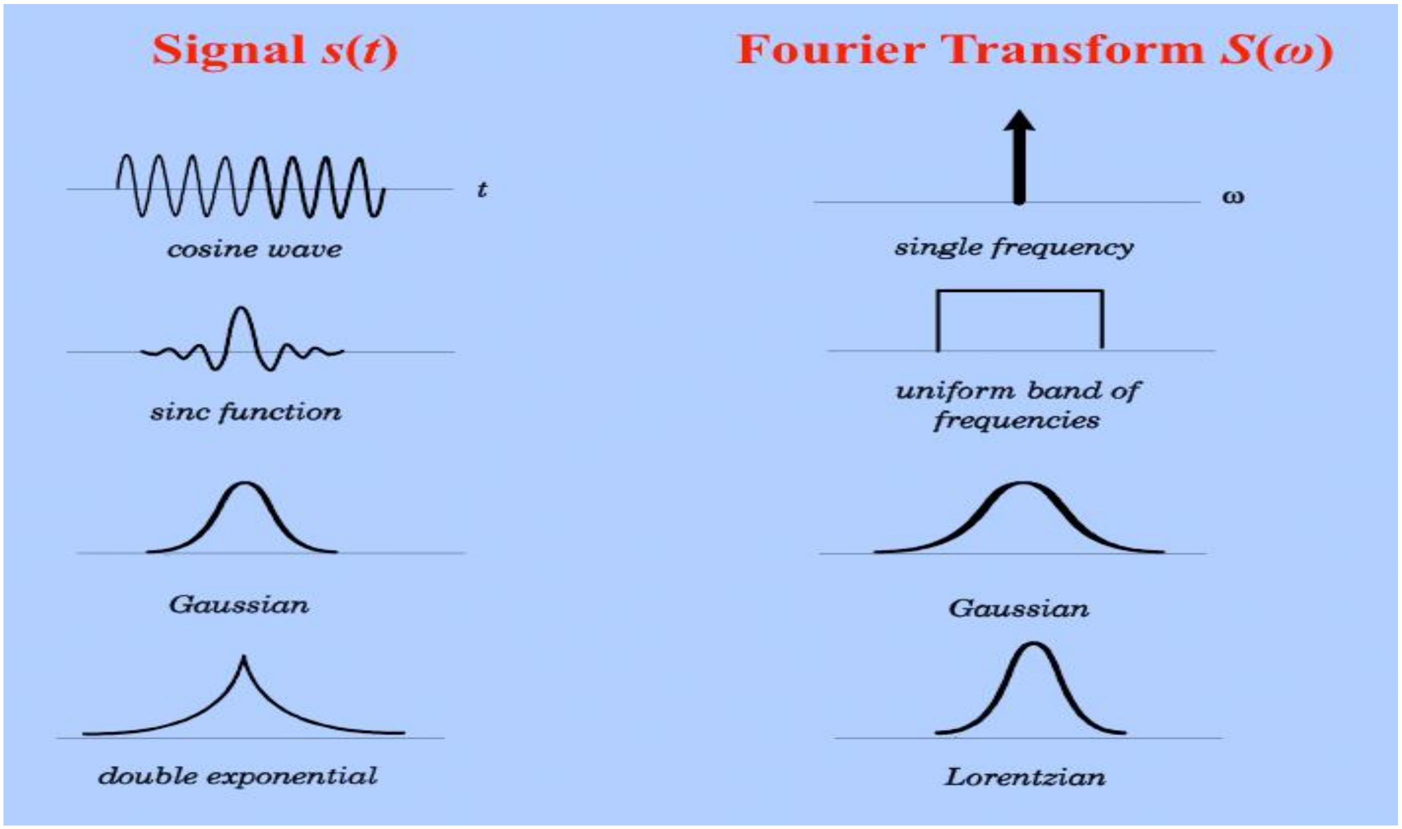
Transformation

A signal can be converted from **time domain into frequency domain** using mathematical operators called transforms/ transformation.

There are many kind of transformation that does this. Some of them are given below.

- **Fourier Series**
- **Fourier transformation**
- **Laplace transform**
- Z transform





Laplace Transform Table

$$\mathcal{L}\left(\frac{df(t)}{dt}\right) = \int_{0^-}^{\infty} \frac{df(t)}{dt} e^{-st} dt$$

$$\int_a^b u \cdot dv = u \cdot v\Big|_a^b - \int_a^b v \cdot du$$

$$du = -s \cdot e^{-st} dt \qquad u = e^{-st}$$

$$v = f(t) \qquad dv = \frac{df(t)}{dt} dt$$

$$\int_{0^-}^{\infty} \frac{df(t)}{dt} e^{-st} dt = \left[e^{-st} \cdot f(t)\right]_{0^-}^{\infty} - \int_{0^-}^{\infty} f(t) \cdot (-s) \cdot e^{-st} dt$$

$$= \left[e^{-st} \cdot f(\infty) - e^{-0^- t} \cdot f(0^-)\right] + s\int_{0^-}^{\infty} f(t) \cdot e^{-st} dt$$

$$\int e^{-px} \sin kx \, dx = -\frac{1}{p} e^{-px} \sin kx - \int -\frac{1}{p} e^{-px} \cdot k \cos kx \, dx$$

$$= -\frac{1}{p} e^{-px} \sin kx + \frac{k}{p} \int e^{-px} \cos kx \, dx$$

$$= -\frac{1}{p} e^{-px} \sin kx +$$

$$\frac{k}{p}\left[-\frac{1}{p} e^{-px} \cos kx + \int \frac{1}{p} e^{-px} \cdot (-k \sin kx) \, dx\right]$$

$$= -\frac{1}{p} e^{-px} \sin kx - \frac{k}{p^2} e^{-px} \cos kx$$

$$-\frac{k^2}{p^2} \int e^{-px} \sin kx \, dx$$

Electrical 4 U

$$X(z) = \sum_{n=0}^{\infty} x[n] z^{-n}$$

Out of all these, we will thoroughly discuss Fourier series and Fourier transformation in our next DIP context.

Frequency components

Any image in spatial domain can be represented in a frequency domain. But what do this frequency actually mean.

We will divide frequency components into two major components.

- Low frequency components

  Low frequency components in an image correspond to **smooth** regions.

- High frequency components

  High frequency components correspond to **edges** in an image.

Lowpass filter (smoothing)

A low pass filter is used to pass low-frequency signals. The strength of the signal is reduced and frequencies which are passed is higher than the cut-off frequency. The amount of strength reduced for each frequency depends on the design of the filter. Smoothing is low pass operation in the frequency domain.

Following are some lowpass filters:

Original image

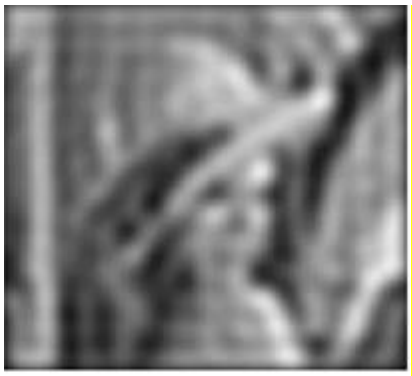
1. Ideal Lowpass Filters

The ideal lowpass filter is used to cut off all the high-frequency components of Fourier transformation.

Below is the transfer function of an ideal lowpass filter.

$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \le D0 \\ 0 & \text{if } D(u,v) > D0 \end{cases}$$

$$D(u,v) = \left[\left(u - \frac{M}{2}\right)^2 + \left(v - \frac{N}{2}\right)^2\right]^{\frac{1}{2}}$$

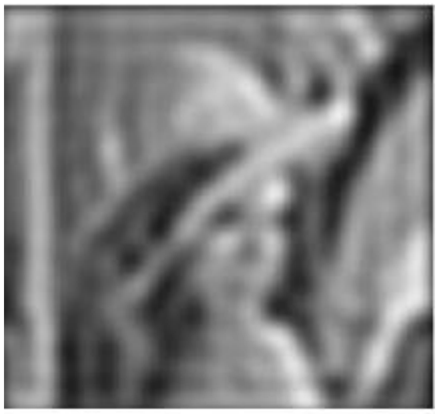


ideal lowpass

2. Butterworth Lowpass Filters

Butterworth Lowpass Filter is used to remove high-frequency noise with very minimal loss of signal components.

$$H(u,v) = \frac{1}{1+\left[\frac{D(u,v)}{D0}\right]^{2n}}$$

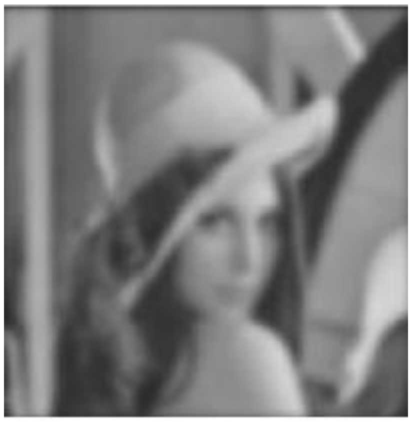


Butterworth Lowpass

3. Gaussian Lowpass Filters

The transfer function of Gaussian Lowpass filters is shown below:

$$H(u,v) = e^{-D^2(u,v)/2D_0^2} \quad (8)$$



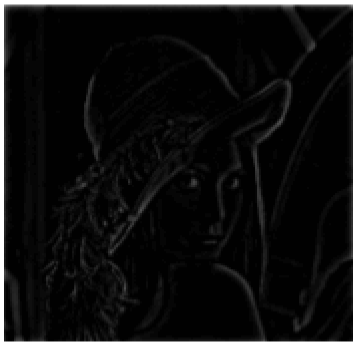
Gaussian Lowpass

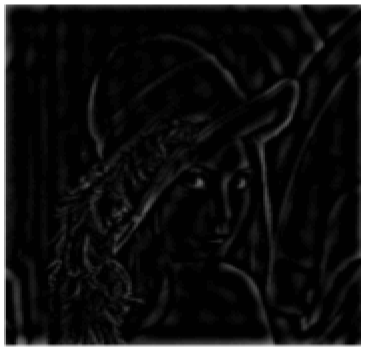
High pass filters (sharpening)

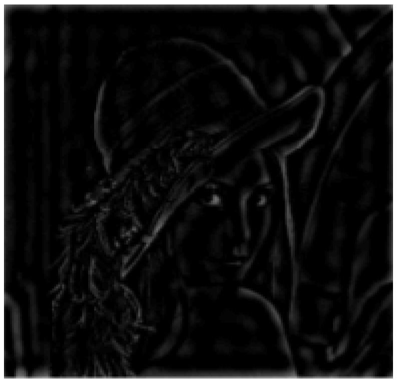
A highpass filter is used for passing high frequencies but the strength of the frequency is lower as compared to cut off frequency. Sharpening is a highpass operation in the frequency domain.

As lowpass filter, it also has standard forms such as Ideal highpass filter, Butterworth highpass filter, Gaussian highpass filter.

Original image


Gaussian highpass

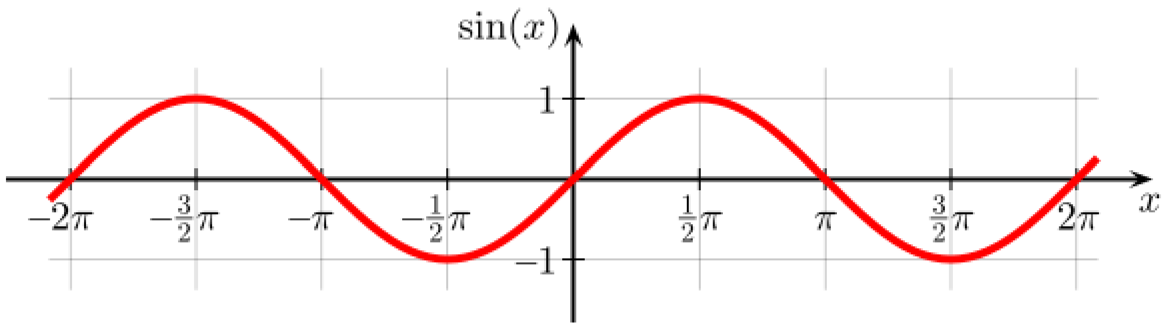

Butterworth highpass


Ideal highpass

Fourier

Fourier was a mathematician in 1822. He gives Fourier series and Fourier transform to convert a **signal** into frequency domain.

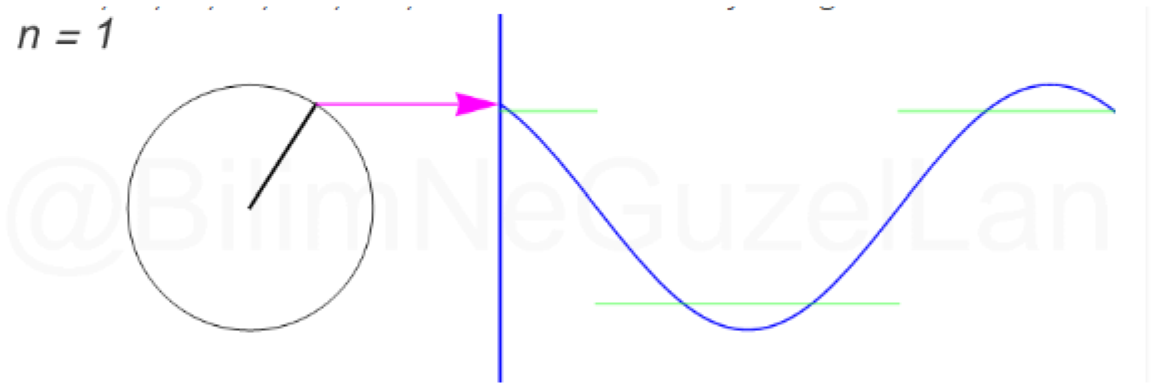
Before going Fourier, let us Analyzing with signals:

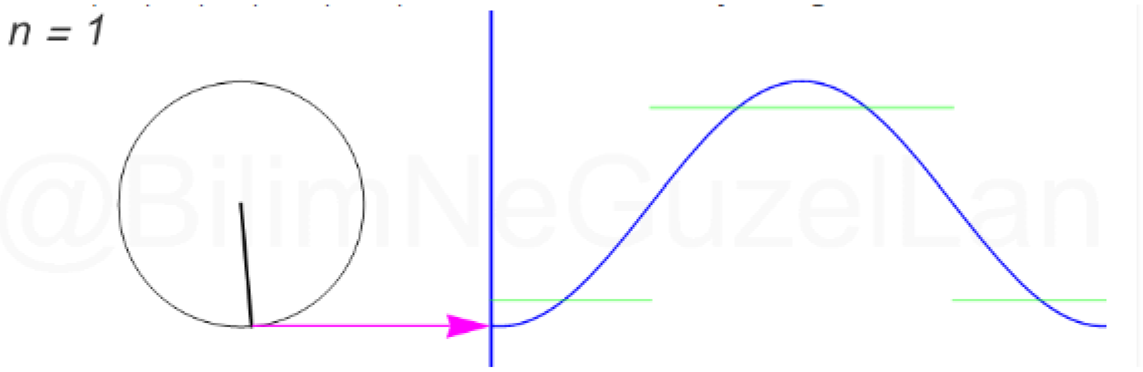


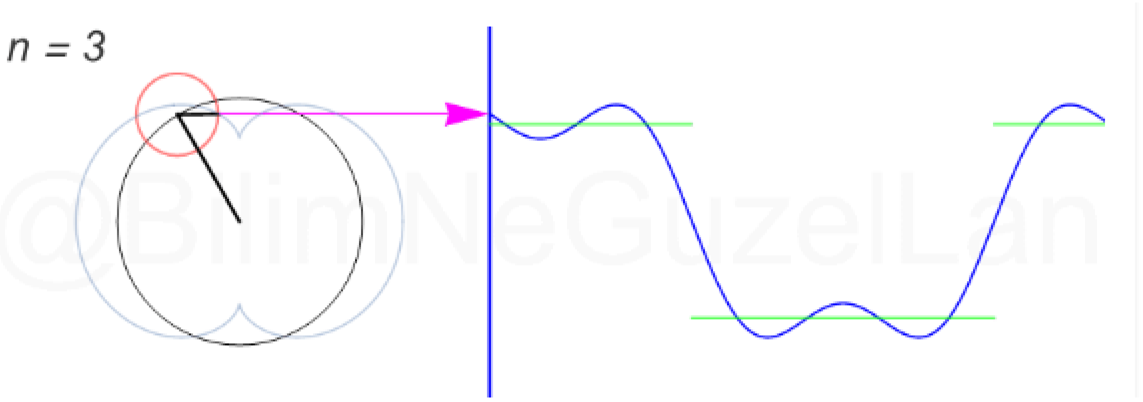
- Consider one of the most common waveforms, the sinusoid.
- A general form of a sinusoidal wave is y(x,t)=Asin(kx−ωt+ϕ)
- where A is the amplitude of the wave,
- ωω is the wave's angular frequency,
- k is the wavenumber,
- and ϕϕ is the phase of the sine wave given in radians.
- This waveform gives the displacement position ("y") of a particle in a medium from its equilibrium as a function of both position "x" and time "t".

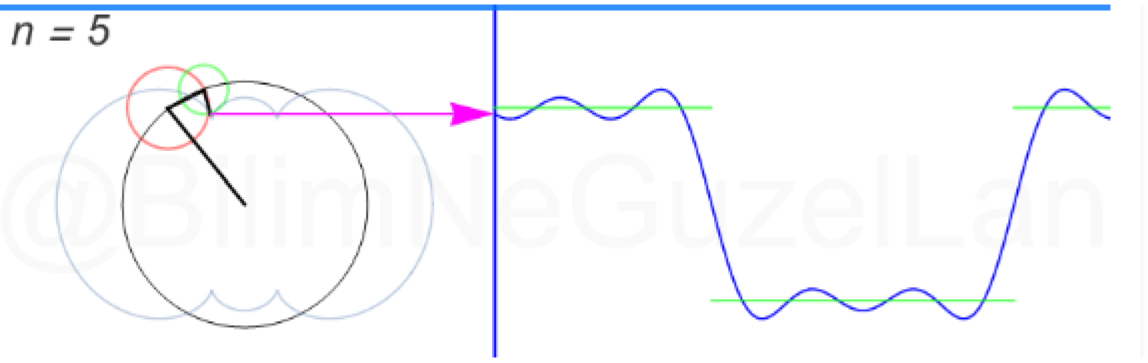
Source: https://bilimneguzellan.net/fuyye-serisi/

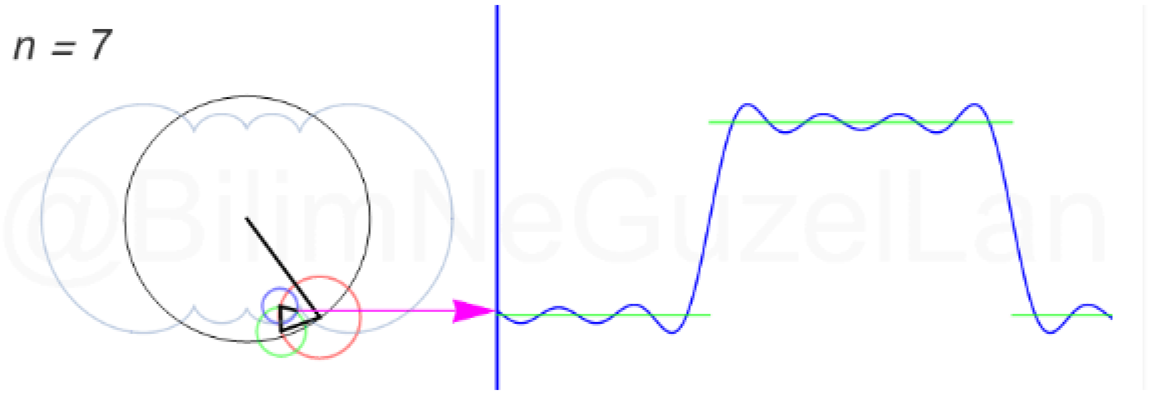
https://www.youtube.com/watch?v=ds0cmAV-Yek

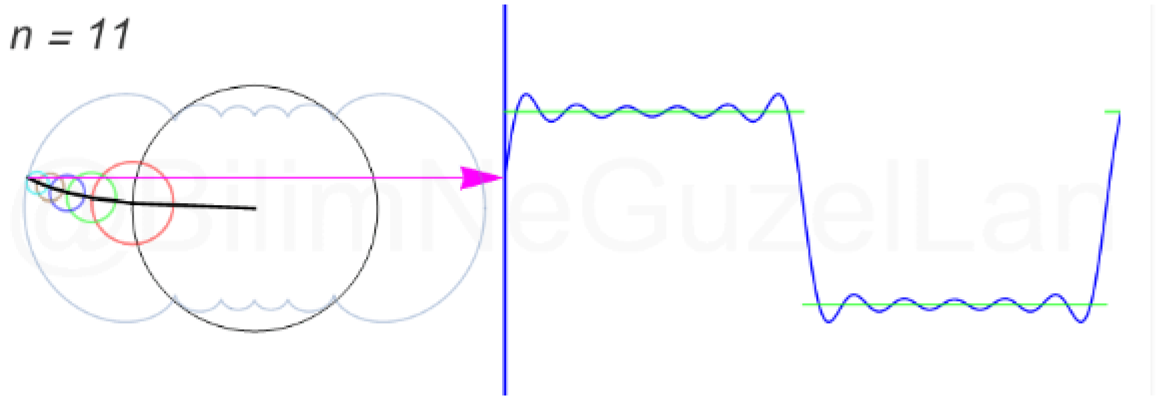


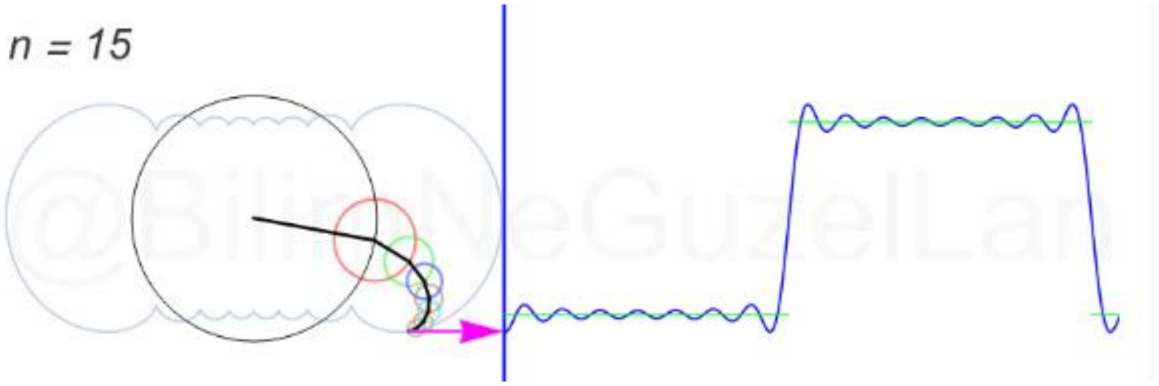

n = 1

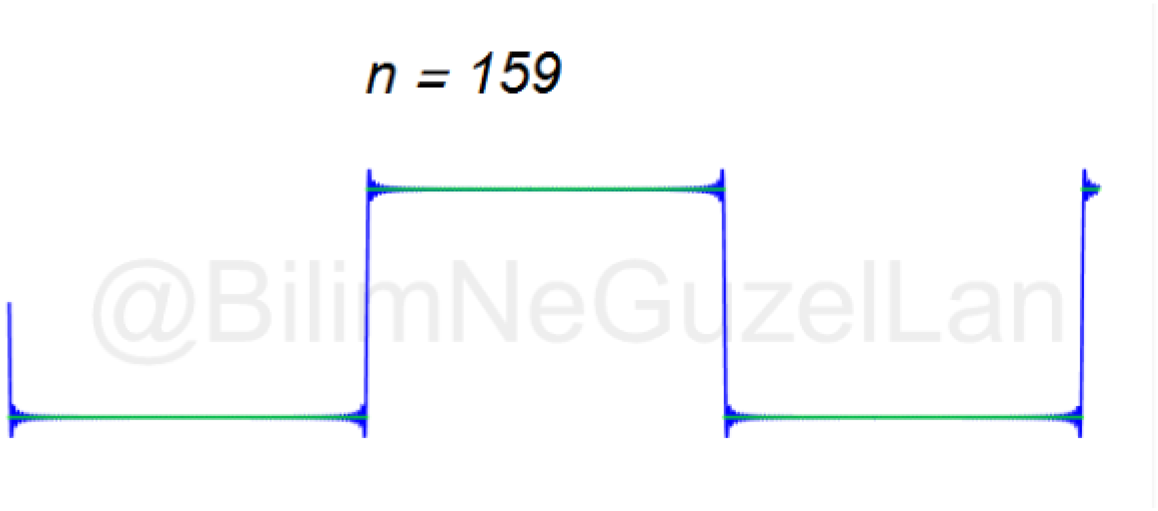

n = 3

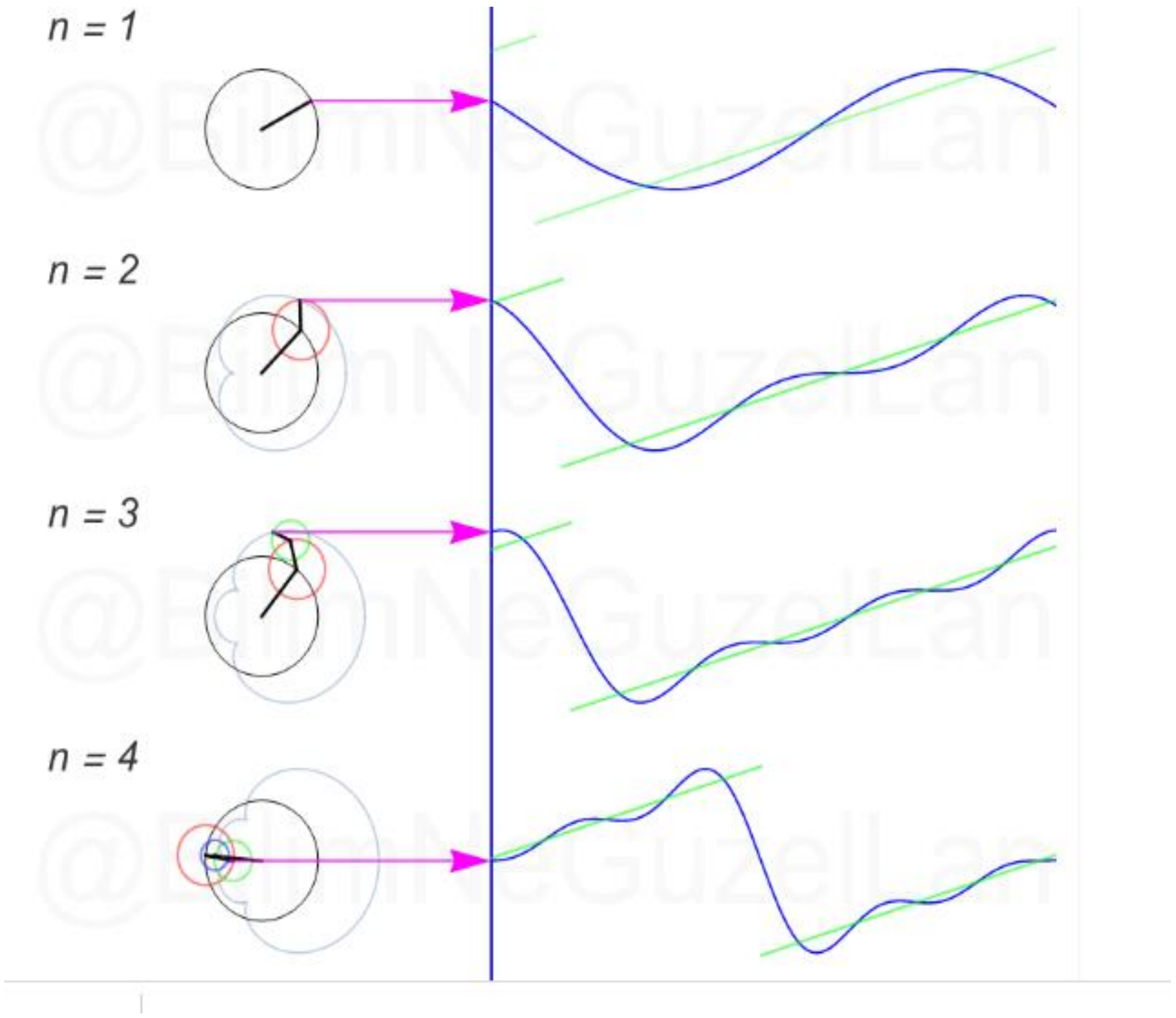

n = 5


n = 7

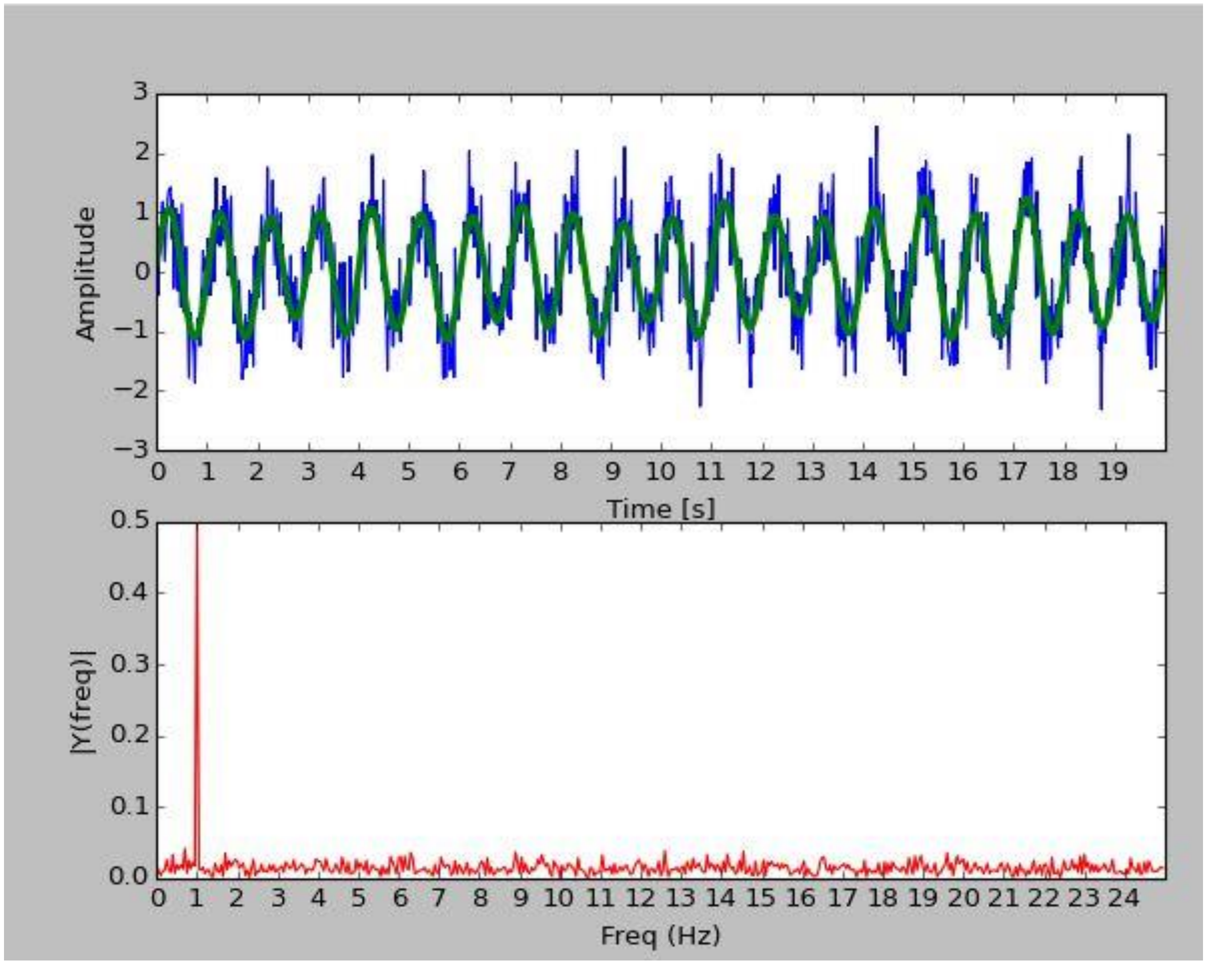

n = 11


n = 15

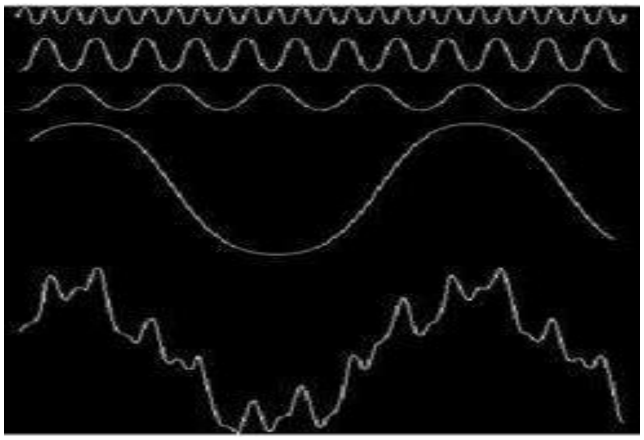

n = 159



In the image below, you take the Fourier transform of the sound above and it tells you that there is more than one frequency (1 Hz) wave.
It has very good uses in image processing and audio processing.



Fourier Series



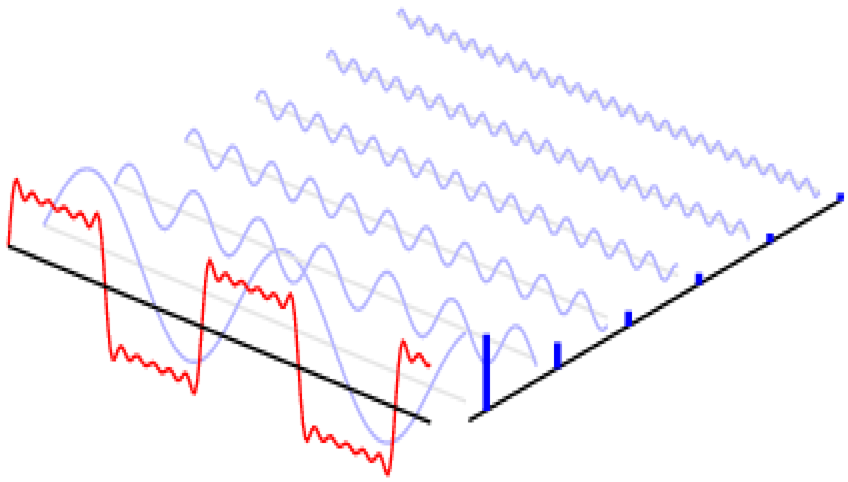
Fourier series simply states that, "periodic signals can be represented into sum of sines and cosines when multiplied with a certain weight".

It further states that periodic signals can be broken down into further signals with the following properties.

- The signals are sine and cosine
- The signals are harmonics of each other
- It can be pictorially viewed.
- In the above signal, the last signal is actually the sum of all the above signals. This was the idea of the Fourier.

How it is calculated?

Since as we have seen in the frequency domain, that in order to process an image in frequency domain, we need to first convert it using into **frequency domain** and we have to take **inverse** of the output to convert it back into **spatial domain**. That's why both Fourier series and Fourier transform has **two formulas**. One for **conversion** and one converting it back to the **spatial domain**. J



Fourier series

The Fourier series can be denoted by this formula.

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)}\, dx\, d$$

The inverse can be calculated by this formula.

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j2\pi(ux+vy)}\, du\, dv.$$

Fourier transform

The Fourier transform simply states that that the non-periodic signals whose area under the curve is finite can also be represented into integrals of the sines and cosines after being multiplied by a certain weight.

The Fourier transform has many wide applications that include, **image compression (e.g JPEG compression), filtering and image analysis.**

Difference between Fourier series and transform:

Although both Fourier series and Fourier transform are given by Fourier, but the difference between them is

- Fourier series is applied on **periodic signals**
- and Fourier transform is applied for **non-periodic signals.**

Which one is applied on images:

Now the question is that which one is applied on the images, the Fourier series or the Fourier transform?

Well, the answer to this question lies in the fact that what images are. Images are non – periodic.

And since the images are non-periodic, so Fourier transform is used to convert them into frequency domain.

Discrete Fourier transform (DFT):

Since we are dealing with images, and in fact digital images, so for digital images we will be working on discrete Fourier transform

Consider the above Fourier term of a sinusoid. It includes three things.

- Spatial Frequency
- Magnitude
- Phase

The *spatial* frequency directly relates with the *brightness of the image*.

The *magnitude* of the sinusoid directly relates with the *contrast*.

Contrast is the difference *between maximum and minimum pixel intensity*. **Phase** contains the color information.

For 1 D Fourier Transform:

$$F(k)=\sum_{x=0}^{N-1} f(x)\, e^{-\frac{j2\pi kx}{N}}$$

Where K= 0,1,2,.....N-1

And its inverse 1 D Fourier Transform is

$$f(x)=\sum_{x=0}^{N-1} F(k)\, e^{\frac{j2\pi kx}{N}}$$

Where K= 0,1,2,.....N-1

The formula for **2-dimensional discrete Fourier transform** is given above.

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

The discrete Fourier transform is actually the sampled Fourier transform, so it contains some samples that denotes an image. In the above formula **f(x,y) denotes the image**, and **F(u,v) denotes the discrete Fourier transform.**

The formula for 2-dimensional inverse discrete Fourier transform is given

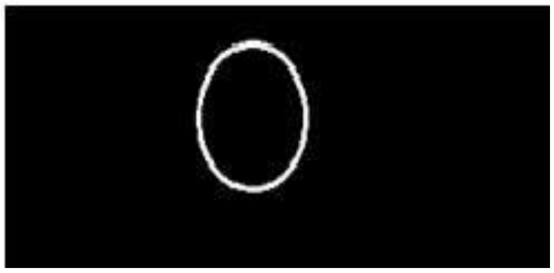
$$f(x,y)=\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F(u,v)\, e^{j2\pi(\frac{ux}{M}+\frac{vy}{N})}$$

The inverse discrete Fourier transform converts the Fourier transform back to the image.

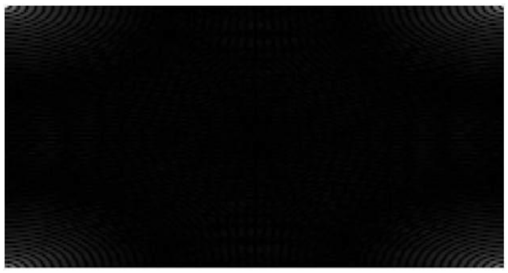
Consider this signal

Now we will see an image, whose we will calculate FFT(Fast Fourier Transform) magnitude spectrum and then shifted FFT magnitude spectrum and then we will take Log of that shifted spectrum.

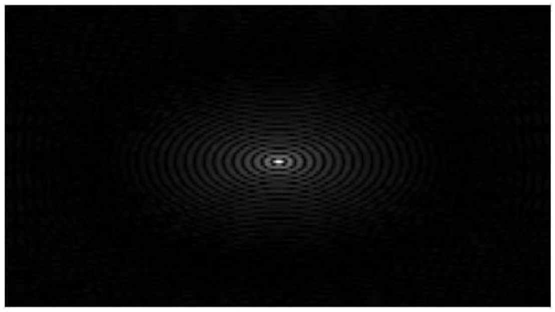
(a) Original Image



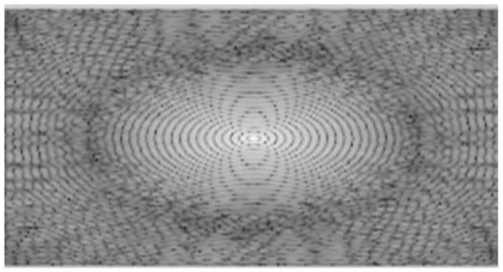
(b) The Fourier transform magnitude spectrum



(c) The Shifted Fourier transform



(d) The Shifted Magnitude Spectrum



DFT Properties

- Translation
- Rotation
- Separability
- Distributive
- property
- Scaling property
- Convolution
- Correlation
- Periodicity

****

DFT of $f(x) = \{0, 1, 2, 1\}$ (10)

Sol^n

DFT ; $F(k) = \sum_{x=0}^{N-1} f(x) \cdot e^{-j2\pi kx/N}$

where, $k = 0, 1, 2, \ldots (N-1)$

Here,

$N = 4$, $k = 0, 1, 2, 3$

we have to find, $F(0), F(1), F(2), F(3)$; $F(k) = ?$

for $k = 0$,

$F(0) = \sum_{x=0}^{3} f(x) \cdot e^{-j2\pi \cdot 0 \cdot x/4}$

$= f(0) \cdot e^{-0} + f(1) \cdot e^{-j2\pi \cdot 0 \cdot 1/4} + f(2) \cdot e^{-j2\pi \cdot 0 \cdot 2/4} + f(3)\, e^{-j2\pi \cdot 0 \cdot 3/4}$

$= f(0) \cdot e^{-0} + f(1) \cdot e^{-0} + f(2) \cdot e^{-0} + f(3) \cdot e^{-0}$

$= [0 \times 0 + 1 \times 1 + 2 \times 1 + 1 \times 1]$ $\quad [\because e^{-0} = 1]$

$= 0 + 1 + 2 + 1$

$= 4$

$[e^{j\theta} = \cos\theta + j\sin\theta]$

for $k = 1$

$F(1) = \sum_{x=0}^{3} f(x)\, e^{-j\cancel{2}\pi \cdot 1 \cdot x/\cancel{4}_2} = \sum_{x=0}^{3} f(x)\, e^{-j\pi x/2}$

$= f(0) \cdot e^{-j\pi \cdot 0/2} + f(1) \cdot e^{-j\pi \cdot 1/2} + f(2)\, e^{-j\pi 2/2} + f(3)\, e^{-j\pi 3/2}$

$= f(0) \cdot e^{-0} + f(1) \cdot e^{-j\pi/2} + f(2)\, e^{-j\pi} + f(3)\, e^{-j\pi 3/2}$

$= 0 \times 1 + 1\left(\cos\frac{\pi}{2} + j\sin\frac{\pi}{2}\right) + 2 \cdot [\cos\pi - j\sin\pi] + 1\left[\cos\frac{3\pi}{2} - j\sin\frac{3\pi}{2}\right]$

$= 0 + [0 - j] + 2[-1 - j \cdot 0] + [0 + j \cdot 1]$

$= 0 - \cancel{j} - 2 + 0\ (+\cancel{j})$

$= -2$

for $k=2$

$F(2) = \sum_{x=0}^{3} f(x) \cdot e^{-j2\pi 2x/4} = \sum_{x=0}^{3} f(x)\, e^{-j\pi x}$

$= f(0) \cdot e^{-0} + f(1) e^{-j\pi} + f(2)\, e^{-j\pi 2} + f(3) \cdot e^{-j\pi 3}$

$= 0 \cdot 1 + 1 \{ \cos\pi - j\sin\pi \} + 2 \{ \cos 2\pi - j \sin 2\pi \} + 1 \{ \cos 3\pi - \sin 3\pi \}$

$= 0 + 1(-1 + j \cdot 0) + 2(1 + j \cdot 0) + 1(-1 + 0 \cdot j)$

$= 0 + (-1) + 2 + (-1)$

$= -1 + 2 - 1$

$= 0$

For $k=3$

$F(3) = \sum_{x=0}^{3} f(x)\, e^{-j2\pi 3x/4} = \sum_{x=0}^{3} f(x)\, e^{-j3\pi x/2}$

$= f(0) e^{-0} + f(1) e^{-j3\pi/2} + f(2) e^{-j3\pi} + f(3) e^{-j9\pi/2}$

$= 0 \cdot 1 + 1 [\cos \frac{3\pi}{2} - j \sin \frac{3\pi}{2}] + 2 [\cos 3\pi - j \sin 3\pi] + 1 [\cos \frac{9\pi}{2} - j \sin \frac{9\pi}{2}$

$= 0 + 1 [0 + j] + 2[-1 + j \cdot 0] + 1 [0 - j \cdot 1]$

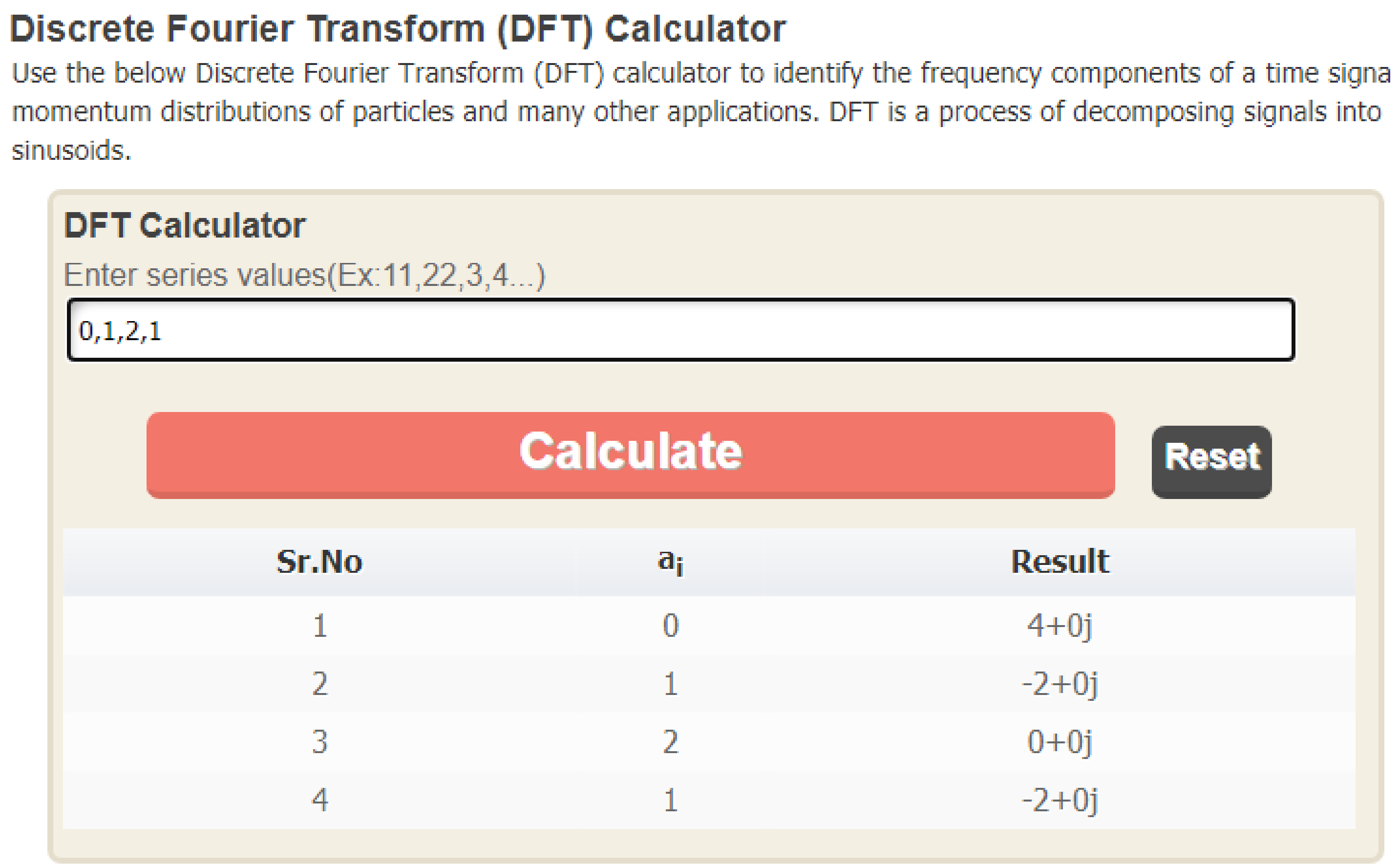
$= 0 + \cancel{j} \oplus -2 \cancel{-j}$

$= -2$

$\therefore f(k) = \{ 4, -2, 0, -2 \}$

$\xleftarrow{}$ DFT

https://www.easycalculation.com/engineering/mechanical/discrete-fourier-transform.php?



Discrete Fourier Transform (DFT) Calculator

Use the below Discrete Fourier Transform (DFT) calculator to identify the frequency components of a time signa momentum distributions of particles and many other applications. DFT is a process of decomposing signals into sinusoids.

DFT Calculator

Enter series values(Ex:11,22,3,4...)

0,1,2,1

Sr.No	a_i	Result
1	0	4+0j
2	1	-2+0j
3	2	0+0j
4	1	-2+0j

Fast Fourier Transform (FFT)

Due to number of complex multiplications and additions to compute the one-dimensional discrete Fourier transform (1D DFT) of a function *f(x,y), Fast Fourier Transform* (FFT) develop to ***reduce the computational complexity***

There are many implementations of the FFT but present a general idea of the **divide-and-conquer** approach toward implementation of the FFT

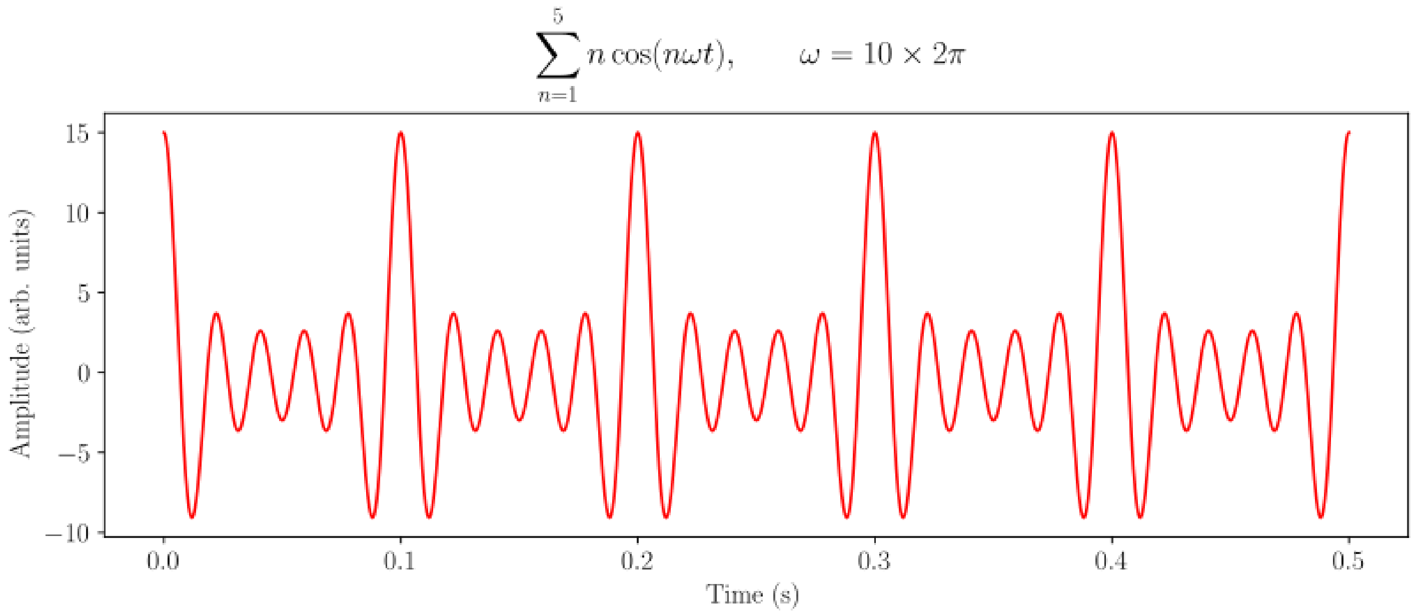
one-dimensional DFT of a one-dimensional signal *f(x)* is

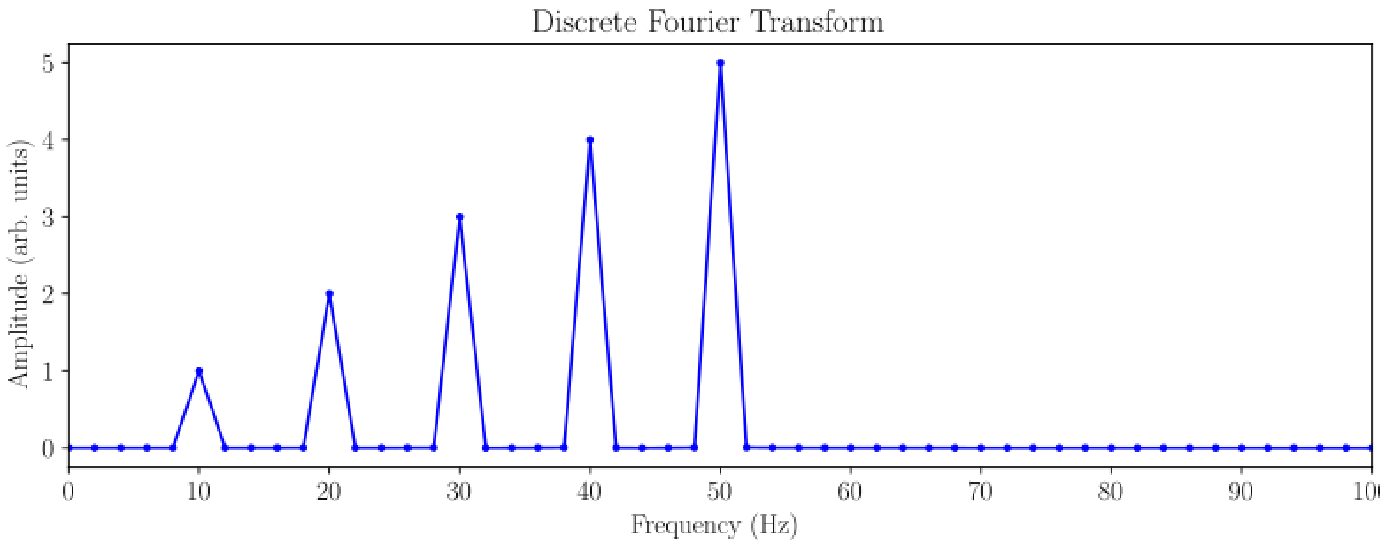
$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x)\, k_N^{ux}$$

where

$$k_{N} = e^{-j2\pi/N}$$

- A fast Fourier transform (FFT) is an algorithm that computes the discrete Fourier transform (DFT) of a sequence, or its inverse (IDFT).
- Fourier analysis converts a signal from its original domain (often time or space) to a representation in the frequency domain and vice versa.
- The DFT is obtained by decomposing a sequence of values into components of different frequencies.
- This operation is useful in many fields, but computing it directly from the definition is often too slow to be practical.
- An FFT rapidly computes such transformations by factorizing the DFT matrix into a product of sparse (mostly zero) factors. As a result, it manages to reduce the **complexity of computing** the DFT from **O (N^2)** which arises if one simply applies the definition of DFT, to **O (N log N)** , where N is the data size.
- The **difference in speed can be enormous**, especially for long data sets where N may be in the thousands or millions.
- In the presence of round-off error, many FFT algorithms are much more accurate than evaluating the DFT definition directly or indirectly.
- There are many different FFT algorithms based on a wide range of published theories, from simple complex-number arithmetic to group theory and number theory.





Applications

- The FFT is used in digital recording, sampling, additive synthesis and pitch correction software.
- The FFT's importance derives from the fact that it has made working in the frequency domain equally computationally feasible as working in the temporal or spatial domain. Some of the important applications of the FFT include.
- Fast large-integer and polynomial multiplication.
- Efficient matrix-vector multiplication for Toeplitz, circulant and other structured matrices.
- Filtering algorithms (see overlap-add and overlap-save methods)
- Fast algorithms for discrete cosine or sine transforms (e.g. fast DCT used for JPEG and MPEG/MP3 encoding and decoding)
- Fast Chebyshev approximation.
- Solving difference equations
- Computation of isotopic distributions.

****

Discrete Cosine Transform

Discrete Cosine Transform (DCT) is the basis for many image and video compression algorithms, especially **the baseline JPEG and MPEG** standards for **compression** of **still and video** images respectively.

- A discrete cosine transform (DCT) expresses a finite sequence of data points in terms of a **sum of cosine functions oscillating at different frequencies.**
- The DCT, first proposed by Nasir Ahmed in 1972, is a widely used transformation technique in signal processing and data compression.
- It is used in most **digital media, including digital images (such as JPEG and HEIF**, where small high-frequency components can be discarded), **digital video** (such as MPEG and H.26x), digital audio (such as Dolby Digital, MP3 and AAC), digital television (such as SDTV, HDTV and VOD), digital radio (such as AAC+ and DAB+), and speech coding (such as AAC-LD, Siren and Opus).
- DCTs are also important to numerous other applications in science and engineering, such as digital signal processing, telecommunication devices, **reducing network bandwidth usage,** and spectral methods for the numerical solution of partial differential equations.

The **one-dimensional** *forward discrete Cosine transform* (1D FDCT) of *N* samples is formulated by

$$F(u) = \sqrt{\frac{2}{N} C(u) \sum_{x=0}^{N-1} f(x)\, Cos\, (\frac{(2x+1)\pi u}{N})}$$

Where u=0,1,2,....N-1

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} \; for\ u = 0 \\ 1 \;\; otherwise \end{cases}$$

The two-dimensional 2DCT can be computed using the one-dimensional **DCT horizontally and then vertically** across the signal because DCT is a separable function.

$$F(u,v) = \frac{2}{\sqrt{MN}} C(u)C(v) \sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y) \cos\left[\frac{\pi(2x+1)u}{2N}\right] \cos\left[\frac{\pi(2y+1)}{2M}\right]$$

The *two-dimensional forward discrete Cosine transform* (2D FDCT) of a block of *M* x *N* samples of a two-dimensional signal *F (x ,* y) is formulated as

for u = 0 , 1 , . . . , *N* - 1 and

v= 0,1, . . . , *M*- 1, where

The above expression for 2D FDCT is clearly a separable function

$$F(u,v) = \sqrt{\frac{2}{M}} C(v) \sum_{y=0}^{M-1} \left\{ \sqrt{\frac{2}{N}} C(u) \sum_{x=0}^{N-1} f(x,y) \cos\left[\frac{\pi(2x+1)u}{2N}\right] \right\} \cos\left[\frac{\pi(2y+1)}{2M}\right]$$

And The *two-dimensional inverse discrete Cosine transform* (2D IDCT) of *F(u,* v) is formulated as

$$f(x,y) = \frac{2}{\sqrt{MN}} \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} C(u)C(v)F(u,v) \cos\left[\frac{\pi(2x+1)u}{2N}\right] \cos\left[\frac{\pi(2y+1)}{2M}\right]$$

for x= 0,1,. .. , *N* - 1 and y = 0,1,. . . , *M* – 1

****

WALSH-HADAMARD TRANSFORM (WHT)

- The Hadamard transform (also known as the Walsh–Hadamard transform, Hadamard–Rademacher–Walsh transform, Walsh transform, or Walsh–Fourier transform) is an **example of a generalized class of Fourier transforms.**
- It performs an orthogonal, symmetric, involutive, linear operation on 2m real numbers (or complex, or hypercomplex numbers, although the Hadamard matrices themselves are purely real).
- The Hadamard transform can be regarded as being built out of **size-2 discrete Fourier transforms (DFTs**), and is in fact equivalent to a multidimensional DFT of size 2 × 2 × ⋯ × 2 × 2.[2]
- It decomposes an arbitrary input vector into a superposition of Walsh functions.
- The discrete Walsh Hadamard transform of a function *f (z)* is denoted by

$$W(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) g(x,u)$$

where,

$$g(x,u) = \frac{1}{N}\left[\prod_{i=0}^{n-1} (-1) b_i(x) b_{n-1-i}(u)\right]$$

where $n = \log_2 N$ and $b_i(z)$ is the ith bit in binary representation of z

- The WHT kernel $g(x, u)$ is a symmetric matrix having a set of N orthogonal rows and columns
- The symmetric WHT kernel for N = 1, 2, 4 and 8 are shown

$$H_1 = [1]$$

$$H_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} H_1 & H_1 \\ H_1 & -H_1 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$H_4 = \frac{1}{\sqrt{2}}\begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} = \frac{1}{\sqrt{4}}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$H8 = \frac{1}{\sqrt{2}}\begin{bmatrix} H_4 & H_4 \\ H_4 & -H_4 \end{bmatrix} = \frac{1}{\sqrt{8}}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

Hence the recursive relation to generate a Walsh-Hadamard Transform kernel can be represented as

$$H_N = \frac{1}{\sqrt{2}}\begin{bmatrix} H_{\frac{N}{2}} & H_{\frac{N}{2}} \\ H_{\frac{N}{2}} & -H_{\frac{N}{2}} \end{bmatrix}$$

The **advantage** of using the WHT is the simplicity in its computation in view of the **binary nature** of the transform kernel. The WHT has been used for **shape analysis**, and other signal and image processing applications.

Applications

3. **Signal Processing**: WHT is used for filtering, signal analysis, and feature extraction due to its orthogonality and fast computation.
4. **Image Processing**: It is employed for image compression, denoising, and feature extraction in images.
5. **Communications**: WHT is used in coding theory, error detection and correction, and spread spectrum techniques.
6. **Speech Processing**: It is used for speech signal analysis and synthesis.

7. **Cryptography**: The transform finds application in cryptographic algorithms and secure communication protocols.

https://en.wikipedia.org/wiki/Hadamard_transform

****

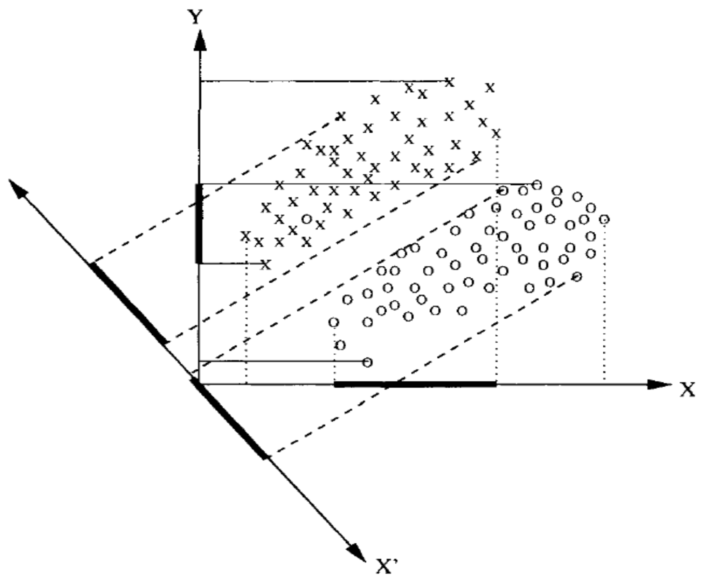
KARHAUNEN-LOEVE TRANSFORM

Karhaunen.-Loeve Transform, or *Principal Component Analysis* **(PCA)** has been a popular technique for many **image processing** and **pattern recognition applications**

This transform which is also known as ***Hotelling Transform*** is based on the concepts of **statistical** properties of image pixels or pattern features

Principal component analysis (PCA) forms the basis of the Karhunen-Loeve (KL) transform for compact representation of data.

The KL transform and the theory behind the principal component analysis are of fundamental importance in signal and image processing. The principle has also found its place in **data mining** for reduction of **large-dimensional datasets.**



- In Figure there are a number of two **dimensional pattern points,** belonging to two different pattern classes (shown by X and 0 symbols), where each pattern is described by only two features *X* and *Y*
- It may be observed that the projection of the pattern points both on X and Y axis are overlapping. As a result, the two features *X* and *Y* do not exhibit **good discriminability**
- It is possible to find a reduced set of features that may result in better discrimination between the **two classes**. This is shown by the non-overlapping projections of the patterns belonging to two classes on the new feature axis *(X ')* as shown in Figure
- PCA is one such tool which yields an extremely powerful technique for dimensionality reduction and many image processing applications such as compression, classification, feature selection.

Applications

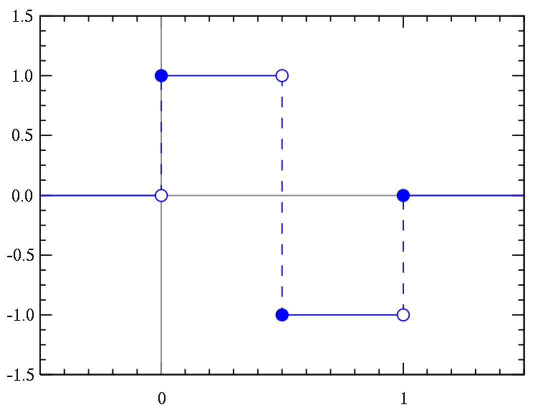
8. **Signal Processing**: KLT is used for noise reduction, feature extraction, and signal compression.

9. **Image Processing**: It is applied in image compression, denoising, and pattern recognition.
10. **Data Analysis**: PCA (a form of KLT) is widely used for exploratory data analysis, data visualization, and multivariate analysis.
11. **Communications**: KLT helps in reducing redundancy in communication signals, improving efficiency.
12. **Finance**: Used for risk management, portfolio optimization, and financial data analysis.

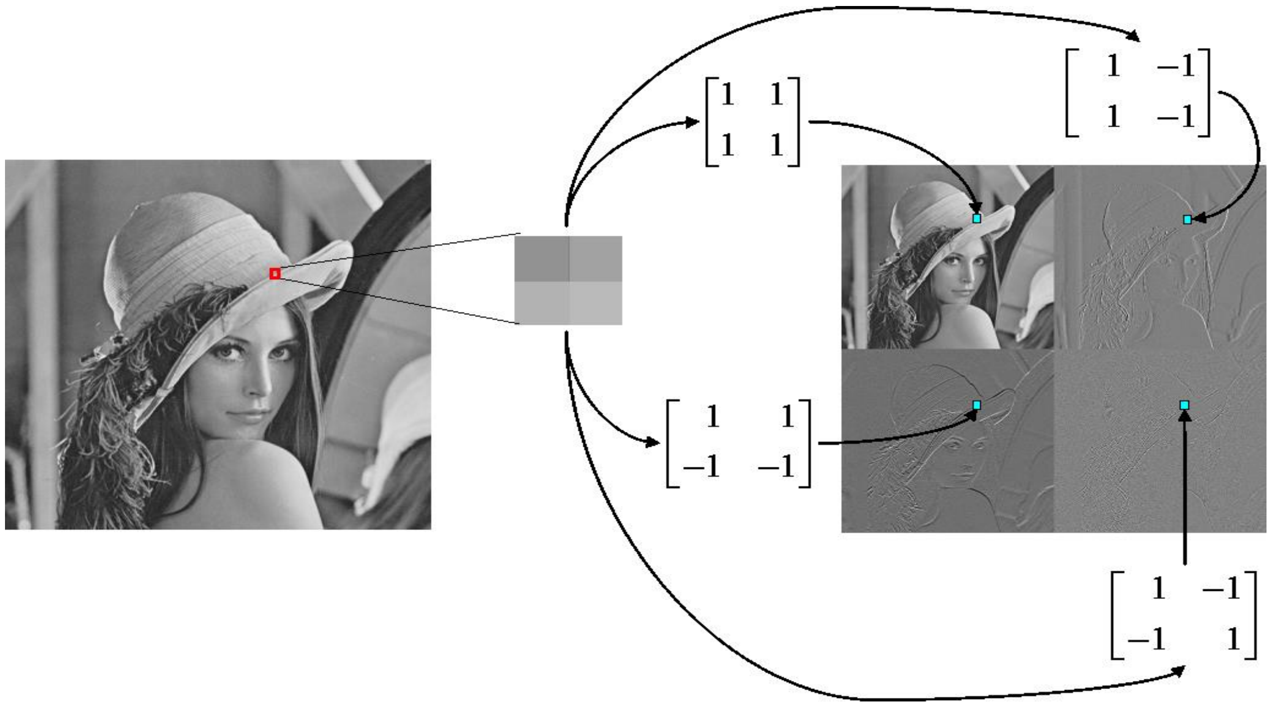
****

The Haar Transform

- Haar proposed the Haar Transform in 1910, more than 70 years before the **wavelet theory** was born.
- Actually, Haar Transform employs the **Haar wavelet filters** but is expressed in a matrix form.
- Haar wavelet is the oldest and simplest wavelet basis.
- Haar wavelet is the only one wavelet basis, which holds the properties of ***orthogonal, (anti-)symmetric*** and compactly supported.
- In mathematics, the Haar wavelet is a sequence of rescaled **"square-shaped" functions** which together form a wavelet family or basis.
- Wavelet analysis is similar to Fourier analysis in that it allows a target function over an interval to be represented in terms of an orthonormal basis.
- The Haar sequence is now recognised as the first known wavelet basis and extensively used as a teaching example.



- Haar used these functions to give an example of an orthonormal system for the space of square-integrable functions on the unit interval [0, 1]. The study of wavelets, and even the term "wavelet", did not come until much later.
- As a special case of the Daubechies wavelet, the Haar wavelet is also known as Db1.
- The Haar wavelet is also the simplest possible wavelet.
- The technical disadvantage of the Haar wavelet is that it is not continuous, and therefore not differentiable. This property can, however, be an advantage for the analysis of signals with sudden transitions, such as monitoring of tool failure in machines.



The Haar transform itself is both separable and symmetric and can be expressed in matrix form

$$T = H F H$$

where F is an image matrix, H is an transformation matrix and T is the resulting transform.

For Haar transform, transformation matrix H contain haar basis function $h_k(z)$. They are defined over the continuous closed interval for **k=0,1,2,…….,N-1 where N=2^n**

To generate H, we define integer k such that k=2^p+q-1, where

$$h_0(z) = h_{00}(z) = \frac{1}{\sqrt{N}}, \quad z \in [0, 1]$$

Haar basis function are

$$h_k(z) = h_{pq}(z) = \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2} & (q-1)/2^p \le z < (q-0.5)/2^p \\ -2^{p/2} & (q-0.5)/2^p \le z < q/2^p \\ 0 & \text{otherwise}, z \in [0, 1]. \end{cases}$$

Applications of the Haar Transform

13. **Image Compression**: Haar Transform is used in JPEG 2000 and other image compression standards due to its ability to compact energy.
14. **Signal Denoising**: By thresholding the detail coefficients, noise can be reduced while preserving important signal features.

15. **Feature Extraction**: Useful in pattern recognition and image processing for extracting features at different resolution levels.
16. **Multiresolution Analysis**: Employed in various applications requiring analysis at multiple scales, such as texture analysis in images.