Chapter 3 – Frequency-domain Fourier Transform

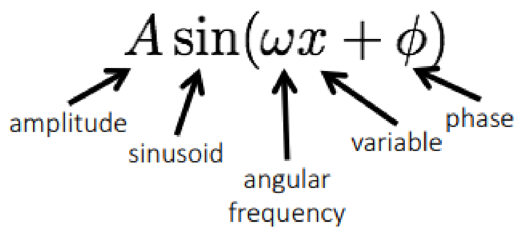
Author: Gianmarco Scarano

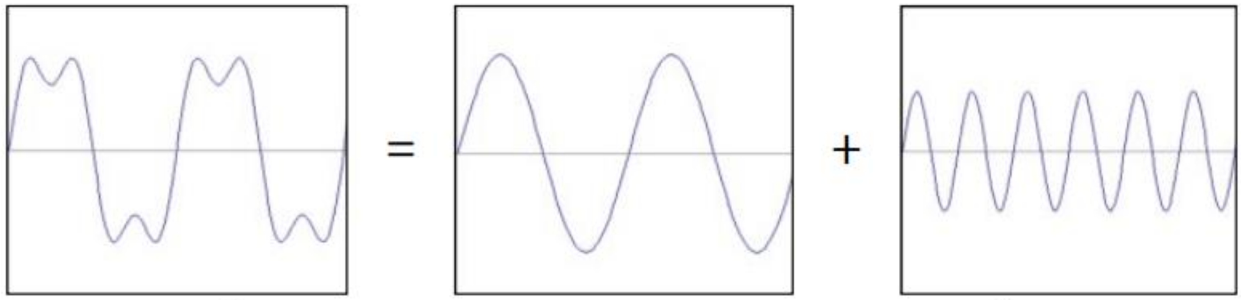
gianmarcoscarano@gmail.com

1. Fourier Series

Fourier in 1807 stated: ‘Any univariate function can be rewritten as a weighted sum of sines and cosines of different frequencies’.

The basic building block is composed as follows:

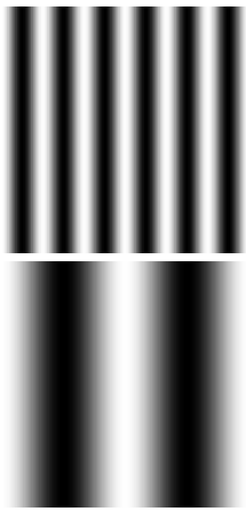


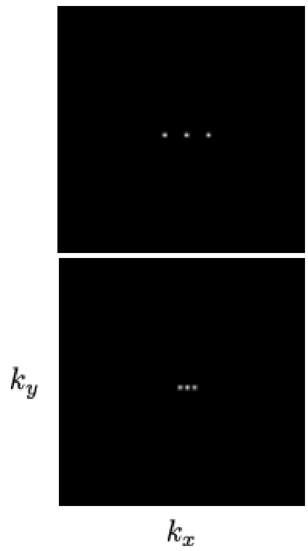


$f(x) = \sin(2\pi x) + \frac{1}{3}\sin(2\pi 3x)$ $\qquad$ $\sin(2\pi x)$ $\qquad$ $\frac{1}{3}\sin(2\pi 3x)$

2. Frequency domain

Spatial domain visualization $\qquad$ Frequency domain visualization





Scaling property of Fourier transform:
if we stretch a function by a factor in the time domain then squeeze the Fourier transform by the same factor in the frequency domain

Also, rotation of the image results in the equivalent rotation of its Fourier Transform.

3. Fourier transform

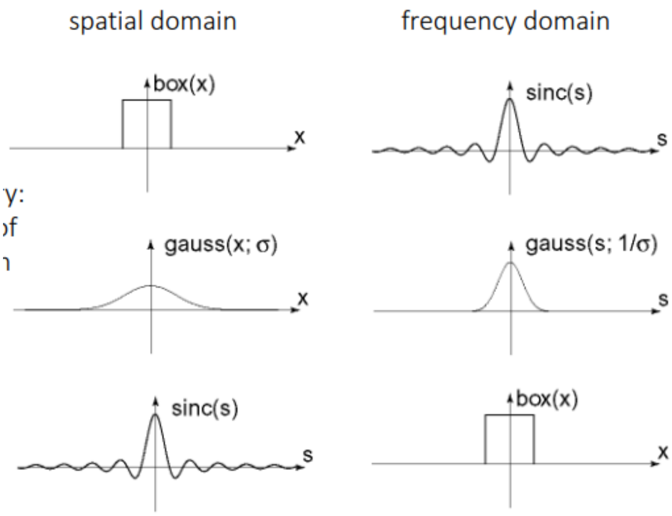
As for background, we know from the Fourier series that any function that periodically repeats itself can be expressed as the sum of sines and/or cosines of different frequencies, each multiplied by a different coefficient. Even functions that are not periodic (but whose area under the curve is finite) can be expressed as the integral of sines and/or cosines multiplied by a weighting function.

Fourier transform

	Fourier transform	inverse Fourier transform
continuous	$F(k) = \int_{\infty}^{-\infty} f(x) e^{-j2\pi kx} dx$	$f(x) = \int_{\infty}^{-\infty} F(k) e^{j2\pi kx} dk$
discrete	$F(k) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-j2\pi kx/N}$ $k = 0, 1, 2, \ldots, N-1$	$\boxed{f(x) = \sum_{k=0}^{N-1} F(k) e^{j2\pi kx/N}}$ $x = 0, 1, 2, \ldots, N-1$

Here, from the $e^{j2\pi kx/N}$ part, we can arrive to the connection to the 'summation of sine waves' idea, through the Euler's formula $e^{j\theta} = \cos\theta + j \cdot \sin\theta$

Fourier transform pairs



$F(k) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-j2\pi kx/N}$ is just a matrix multiplication:

$$\boldsymbol{F} = \boldsymbol{W} \boldsymbol{f}$$

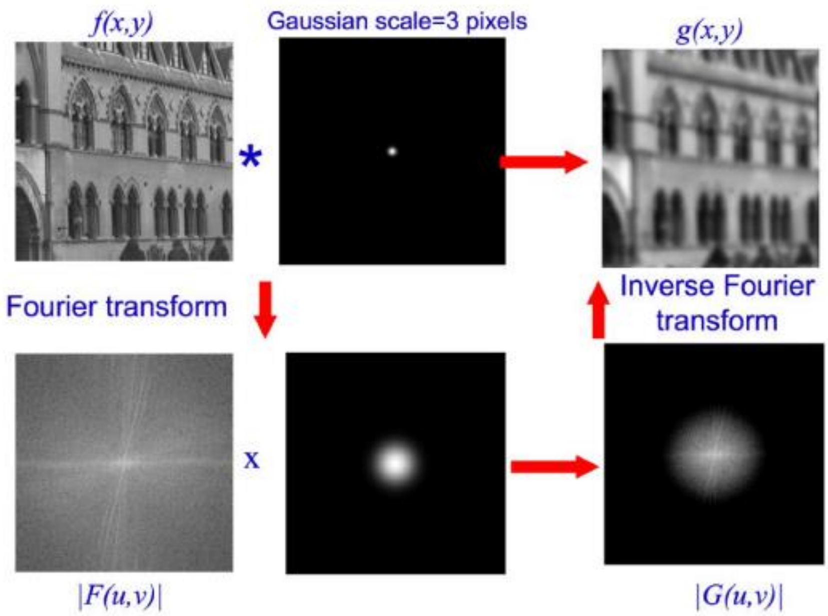
- Basically a matrix-vector product:

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ \vdots \\ X_{N-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N & W_N^2 & W_N^3 & \cdots & W_N^{N-1} \\ 1 & W_N^2 & W_N^4 & W_{N6} & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & \cdots & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix} \times \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{N-1} \end{bmatrix}$$

$$(W_N = e^{-j2\pi/N})$$

4. Frequency-domain filtering

Thinking about the fact that the Fourier transform of the convolution of two functions, is the product of their Fourier transforms: $F\{g * h\} = F\{g\} \cdot F\{h\}$



So, in the end, why does the box filter perform way worse than the Gaussian filter? The answer is related to the Frequency-domain filter as we can see that Box blur keeps a lot of unnecessary details, while the Gaussian filter has a single central concentration of heatmap as we can see from these two images: