The National Higher School of Artificial Intelligence
المدرسة الوطنية العليا للذكاء الاصطناعي

Course
« Computer Vision»

Sid-Ahmed Berrani

2024-2025

Image Filtering in the Frequency Domain

- During the past century, and especially in the past 50 years, entire industries and academic disciplines have flourished as a result of Fourier's ideas.
- The "discovery" of a fast Fourier transform (FFT) algorithm in the early 1960s revolutionized the field of signal processing.
- Today, we can say that there is no discipline of science or engineering that has not been profoundly impacted by the Fourier transform.
- The goal of this lesson is to give a working knowledge of how the Fourier transform and the frequency domain can be used for image filtering.

Image Filtering in the Frequency Domain

Who is Fourier?

Jean Baptiste Joseph Fourier

French mathematician and physicist

21 March 1768 -- 17 May 1830.

He was actually obsessed with heat. He was very interested in knowing how heat propagated through materials of different shapes.

=> That is what led him to develop the Fourier transform.

Image Filtering in the Frequency Domain

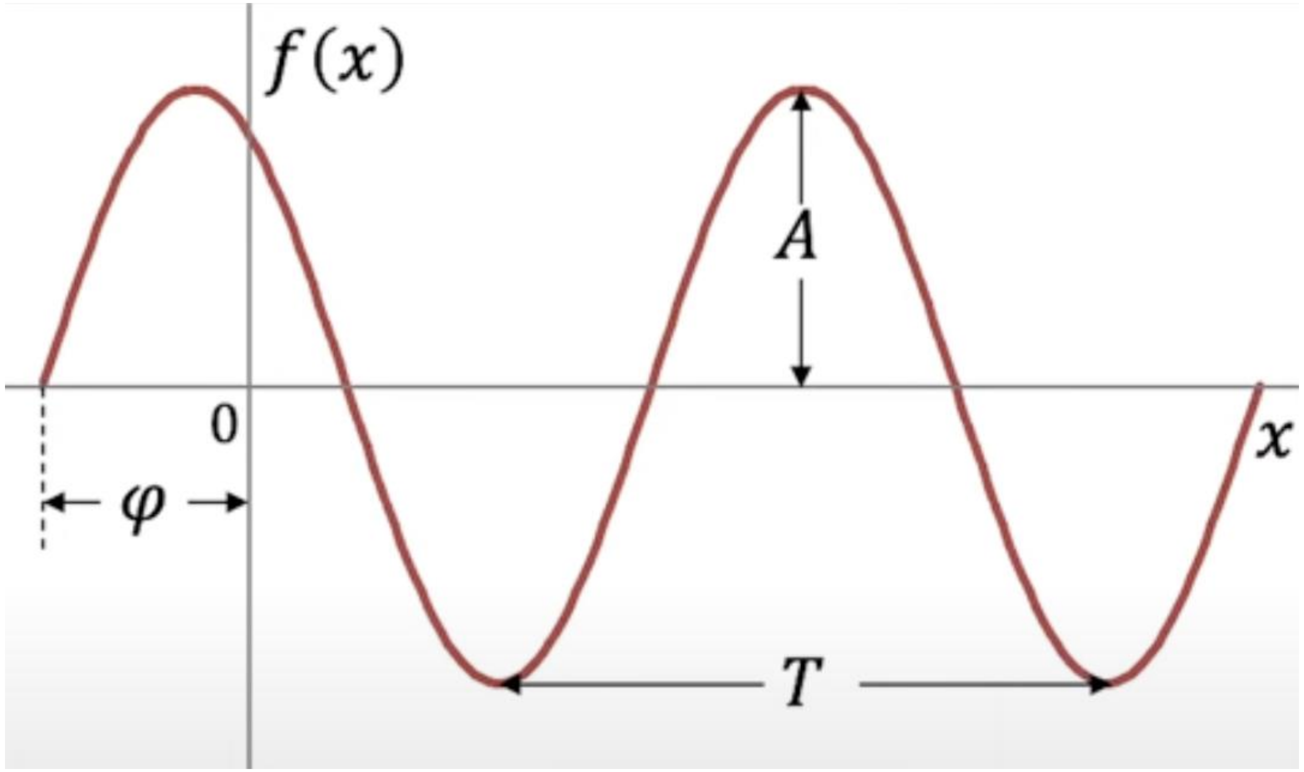
Fourier Transform

- Basically, any periodic function can actually be written as a weighted sum of infinite sinusoids of different frequencies.
- Functions that are not periodic (but whose area under the curve is finite) can also be expressed as the integral of sines and cosines multiplied by a weighting function

Image Filtering in the Frequency Domain

Sinusoid

$$f(x) = A\, sin(2\, \pi\, u\, x + \ \varphi)$$



x: the time

A: amplitude (the peak value of the wave)

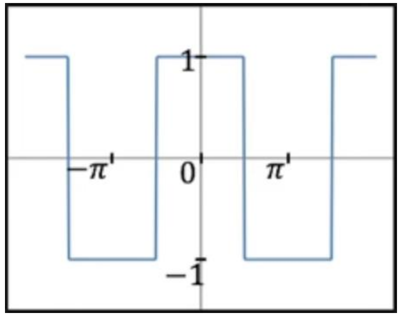
T: period

φ: phase (a shift in time, measured in radians)

u: frequency (how many cycles per second, measured in Hz $= 1/T$)

Image Filtering in the Frequency Domain

Fourier transform to decompose any periodic function



Square wave (period = 2 π).

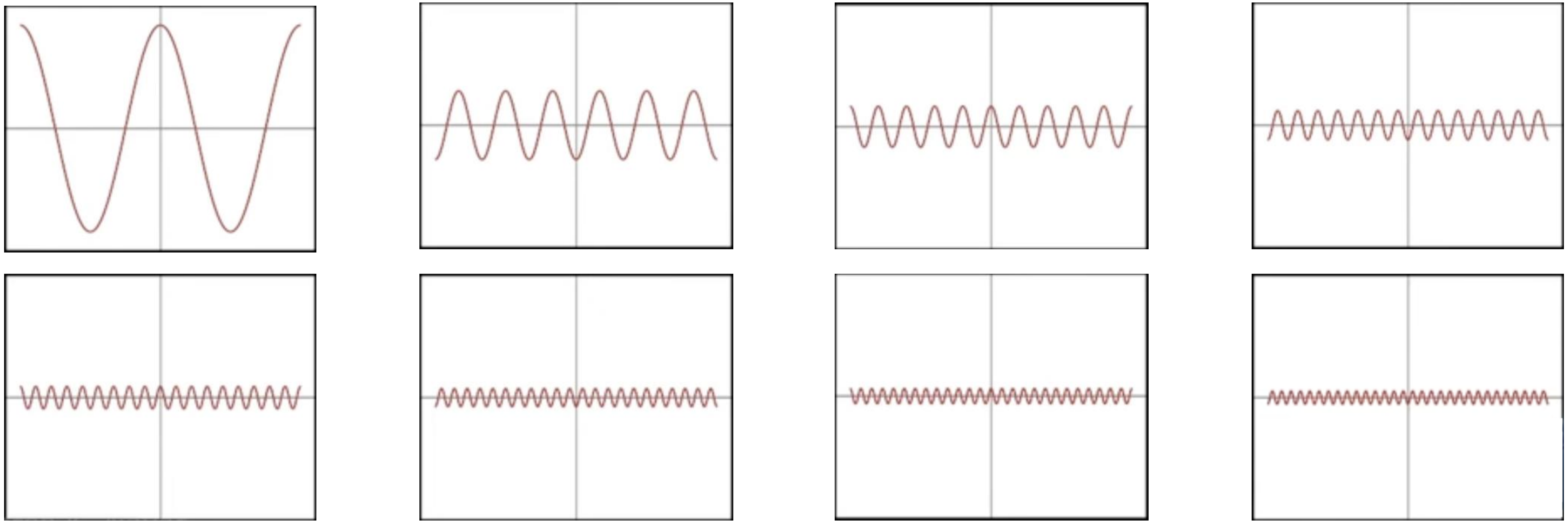
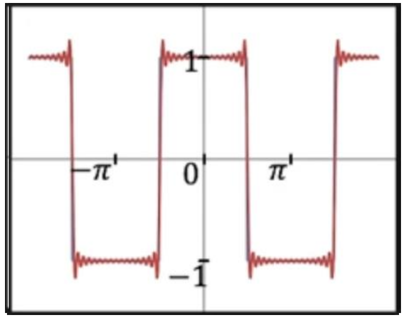


Image Filtering in the Frequency Domain

Fourier transform to decompose any periodic function



Square wave (period = 2 π).

Sum of sinusoids

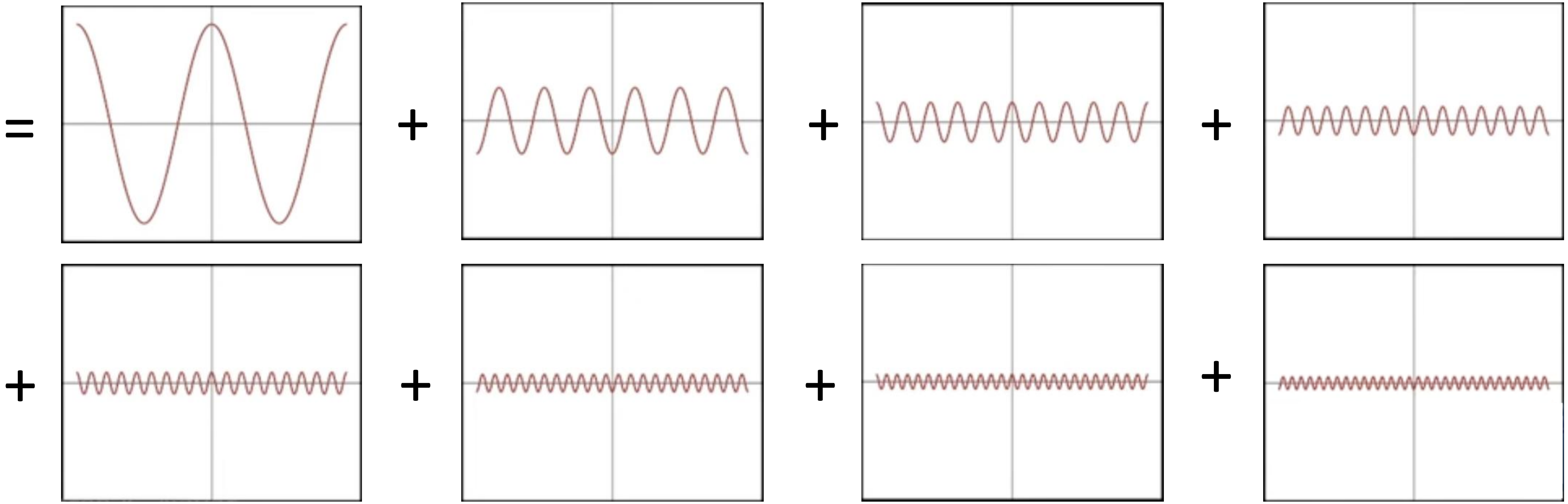
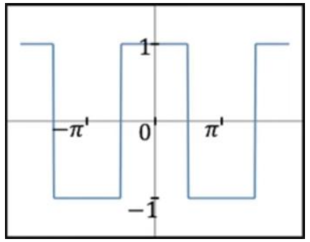
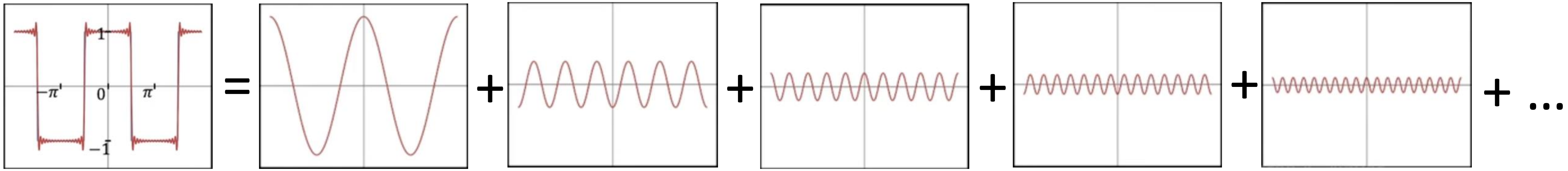


Image Filtering in the Frequency Domain

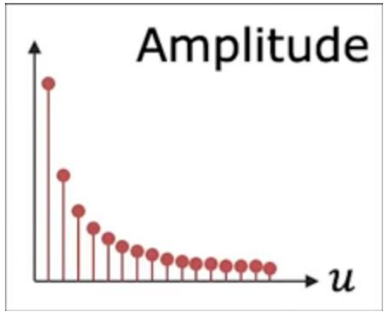
Fourier transform to decompose any periodic function



Square wave (period = 2 π).



Sum of sinusoids



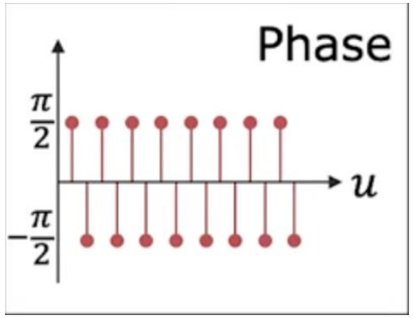
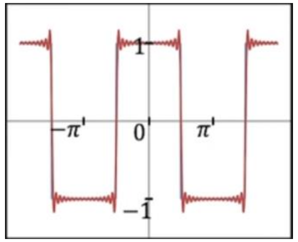


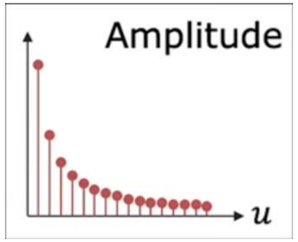
Image Filtering in the Frequency Domain

Fourier transform:

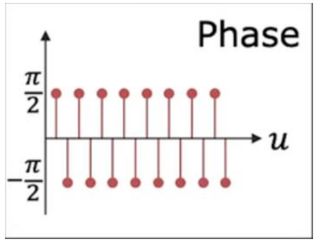


- It represents a signal ***f(x)*** in terms of **amplitudes** and **phases** of **its** constituent **sinusoids**.

$$f(x) \quad -> \quad \textbf{FT} \quad -> \quad F(u)$$



Inverse Fourier transform:



- It computes the spatial signal ***f(x)*** from the **amplitudes** and **phases** of the constituent **sinusoids**.

$$F(u) \quad -> \quad \textbf{IFT} \quad -> \quad f(x)$$

Image Filtering in the Frequency Domain

Fourier Transform (of a 1D signal):

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi ux} dx$$

x: space

u: frequency

$e^{i\theta} = \cos\theta + i\sin\theta$

$i = \sqrt{-1}$

Inverse Fourier Transform (of a 1D signal):

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{i2\pi ux} du$$

Image Filtering in the Frequency Domain

Why do we have $e^{i\theta} = \cos\theta + i\sin\theta$? $\quad (i = \sqrt{-1})$

Expand $e^{i\theta}$ using Taylor Series:

$$e^{i\theta} = 1 + i\theta + \frac{(i\theta)^2}{2!} + \frac{(i\theta)^3}{3!} + \frac{(i\theta)^4}{4!} + \frac{(i\theta)^5}{5!} + \frac{(i\theta)^6}{6!} + \cdots$$

$$e^{i\theta} = \underbrace{\left(1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \frac{\theta^6}{6!} + \cdots\right)}_{\cos\theta} + i\underbrace{\left(\theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \frac{\theta^7}{7!} + \cdots\right)}_{\sin\theta}$$

Image Filtering in the Frequency Domain

⇒ **The Fourier Transform is Complex!**

$F(u)$ holds the Amplitude and the Phase of the sinusoid of frequency u

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi ux} dx$$

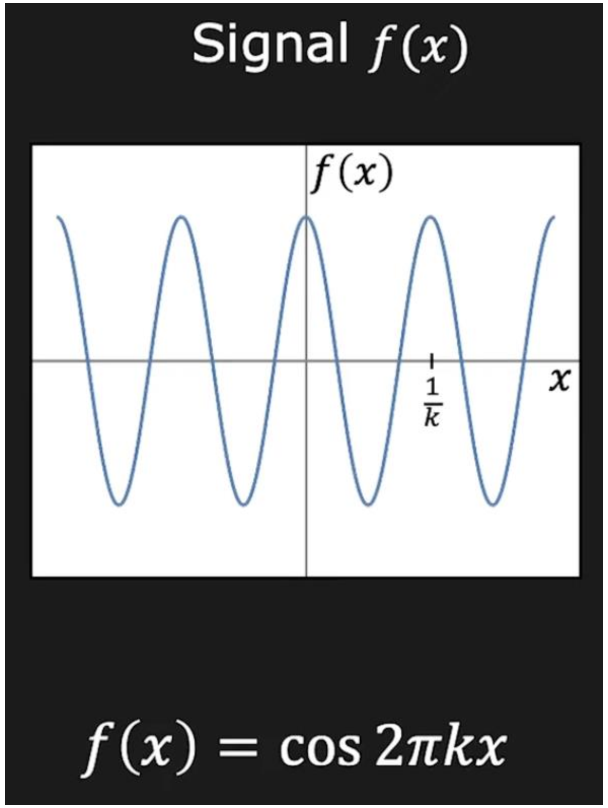
$$F(u) = \Re\{F(u)\} + i\,\Im\{F(u)\} = |A(u)|\, e^{i\varphi(u)}$$

Amplitude: $A(u) = \sqrt{\Re\{F(u)\}^2 + \Im\{F(u)\}^2}$

Phase: $\varphi(u) = \mathrm{atan2}(\Im\{F(u)\},\ \Re\{F(u)\})$

Image Filtering in the Frequency Domain

Few examples of the Fourier transform:



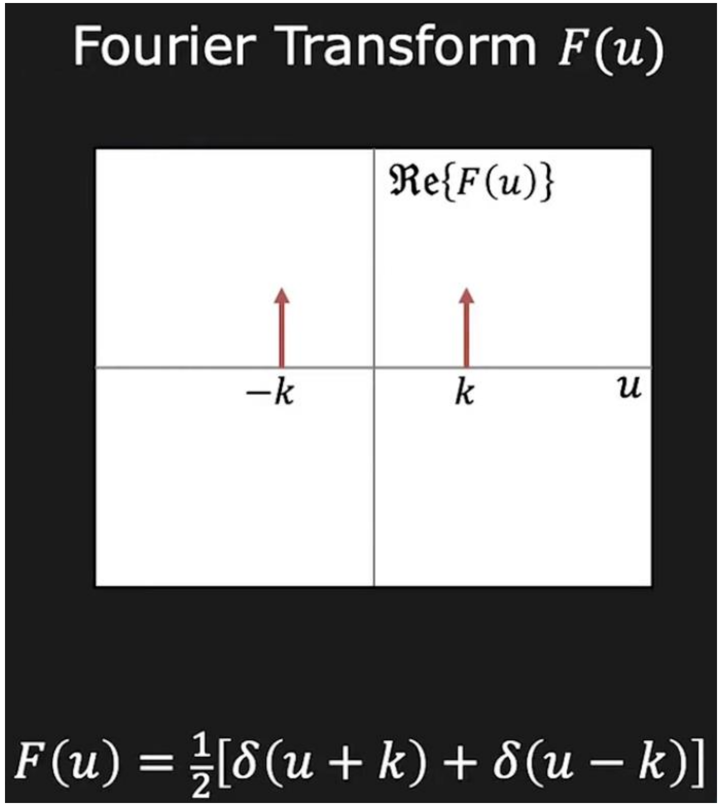


Image Filtering in the Frequency Domain

Few examples of the Fourier transform:

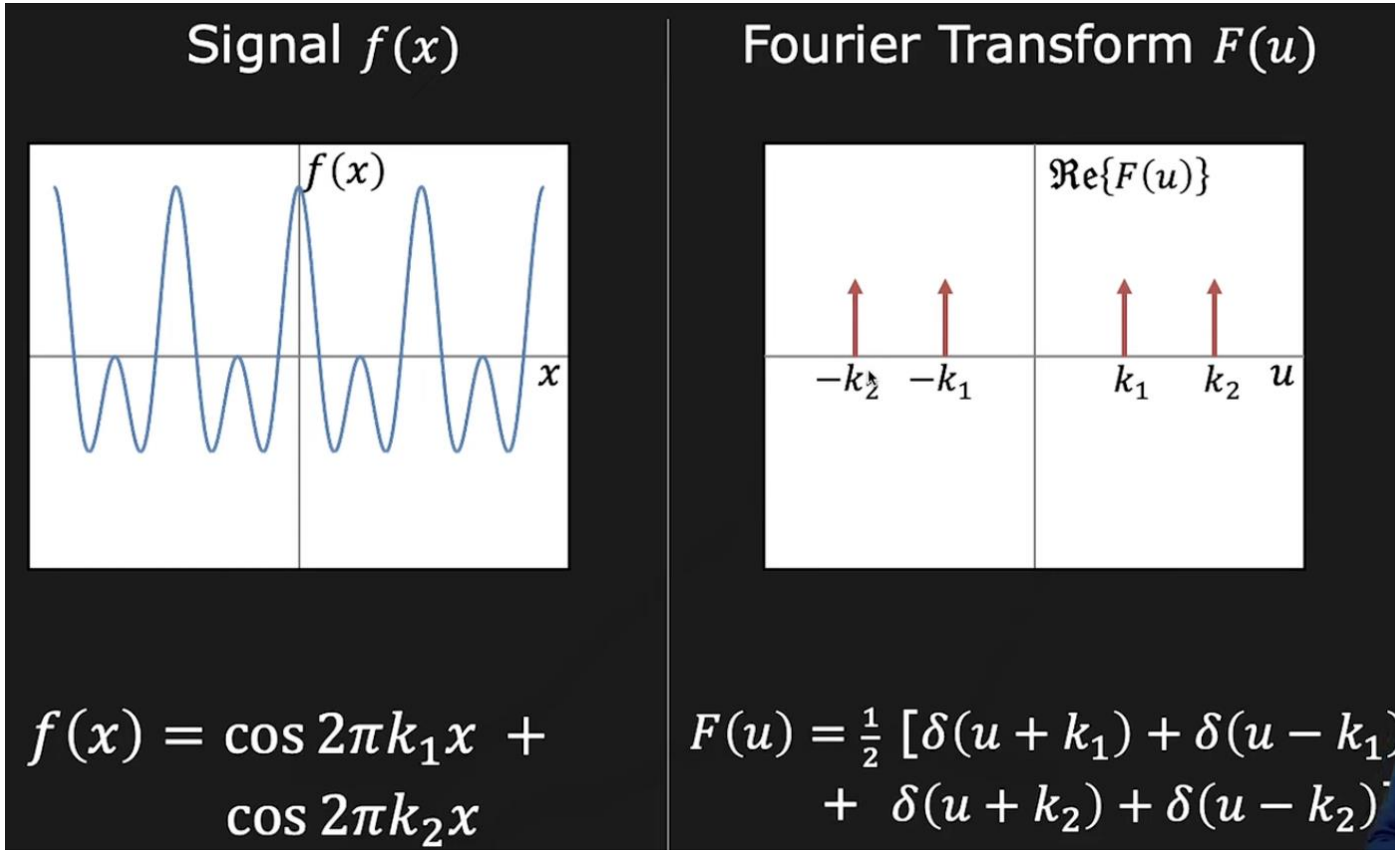


Image Filtering in the Frequency Domain

Few examples of the Fourier transform:

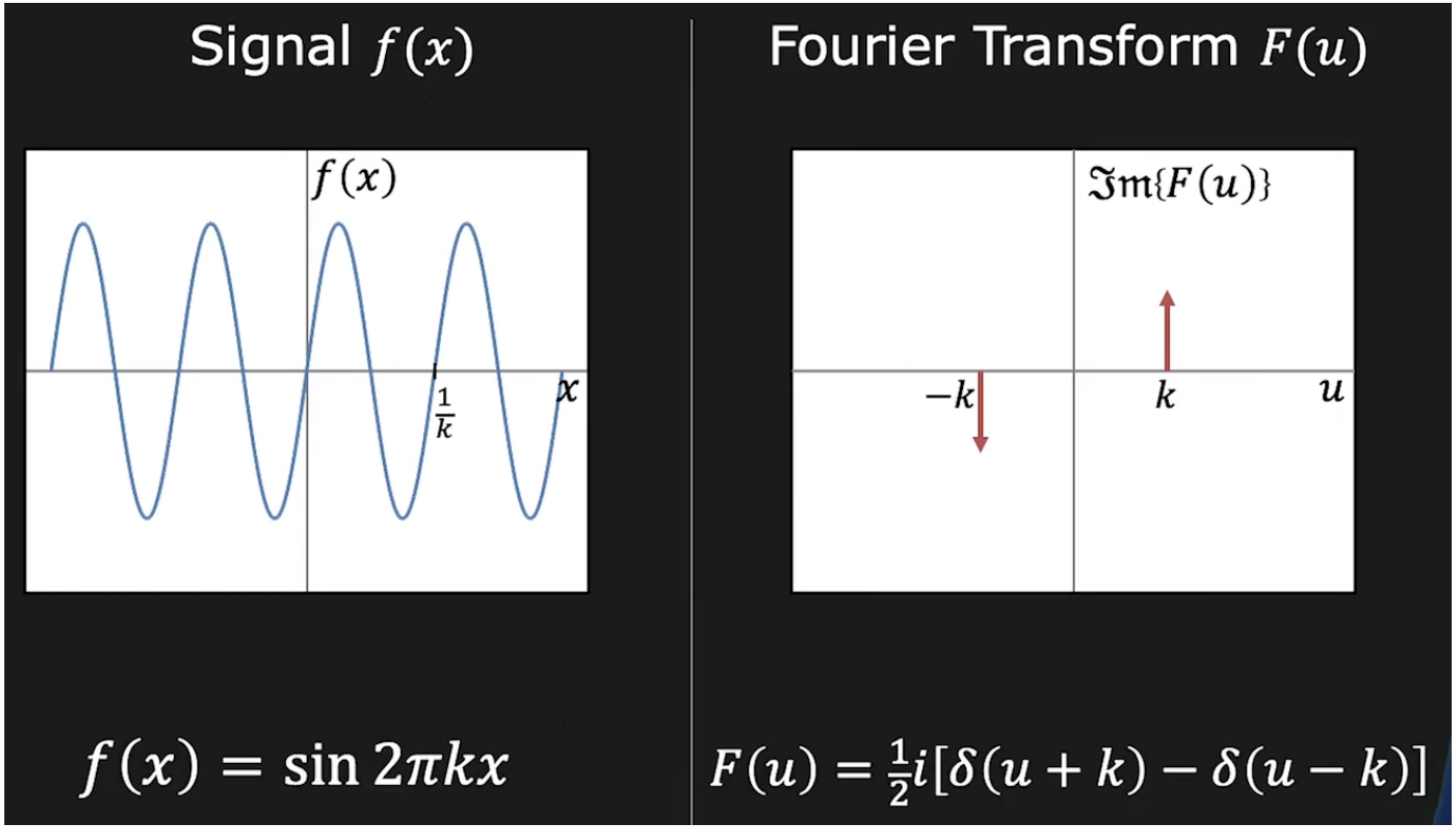


Image Filtering in the Frequency Domain

Few examples of the Fourier transform:

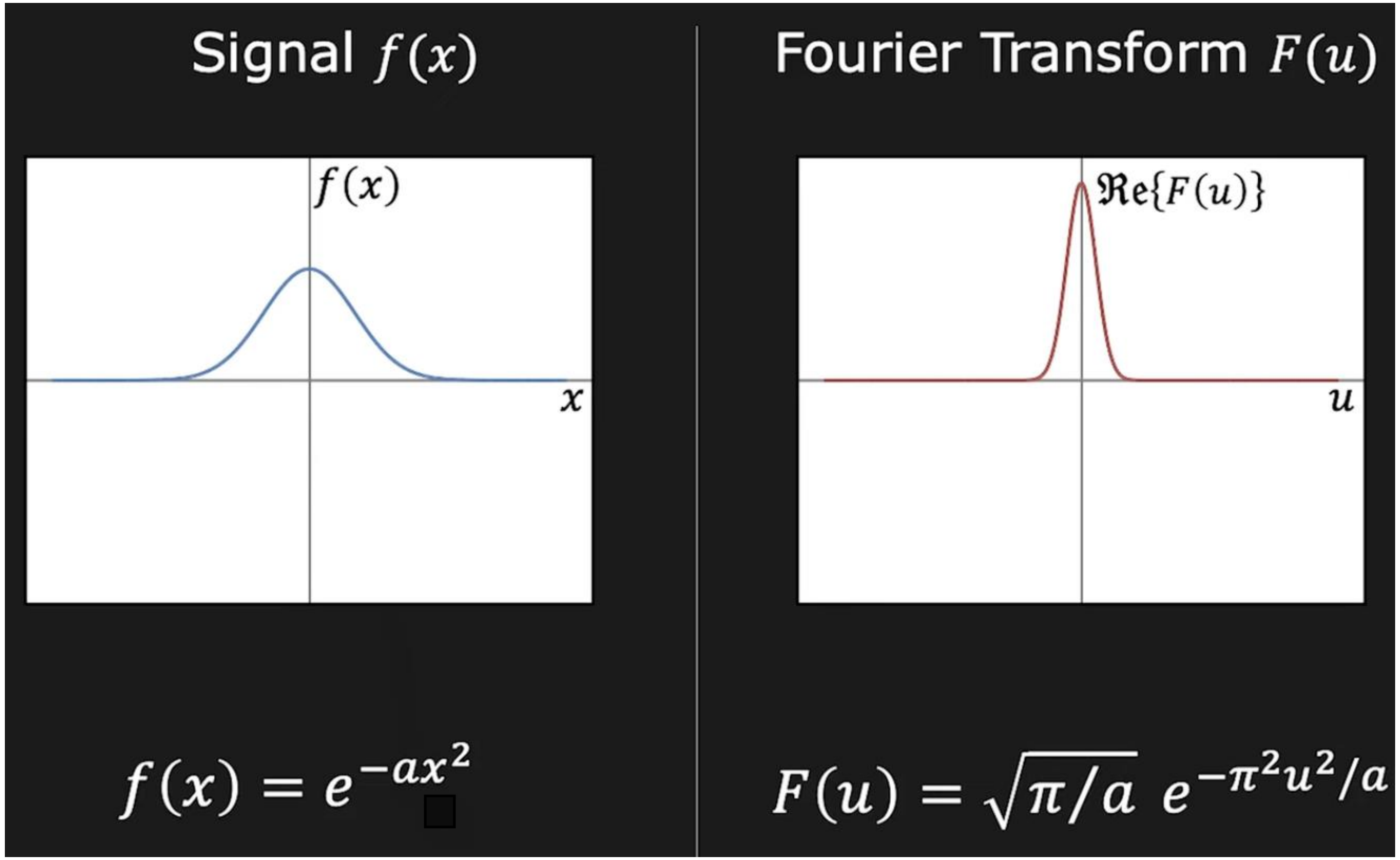


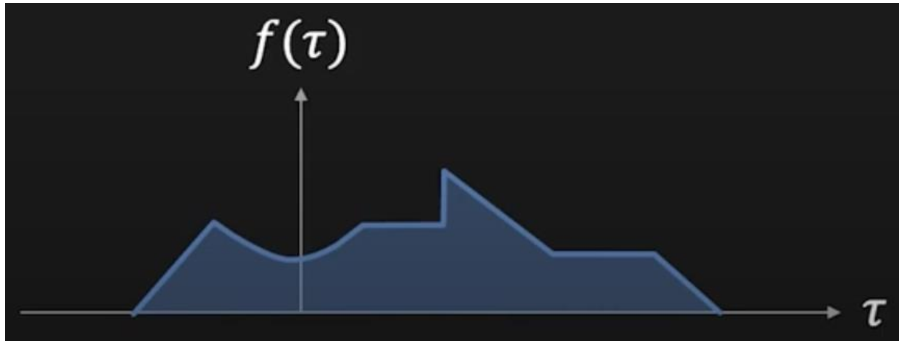
Image Filtering in the Frequency Domain

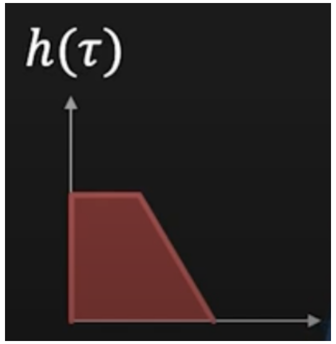
Fourier transform and convolution

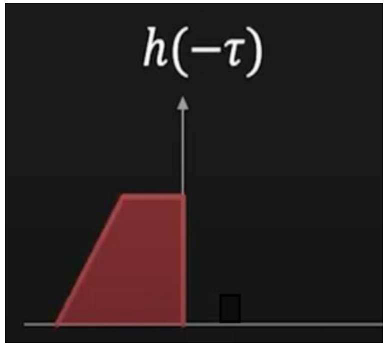
- A close and important relationship.

Convolution of 2 functions: *f(x)* and *h(x)*

$$g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)\, h(x-\tau)\, d\tau$$







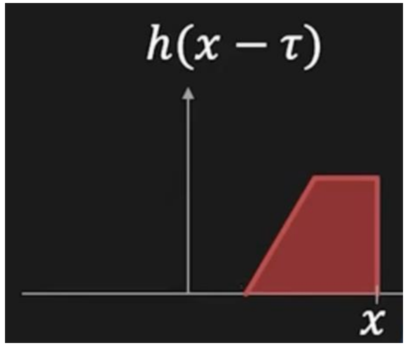


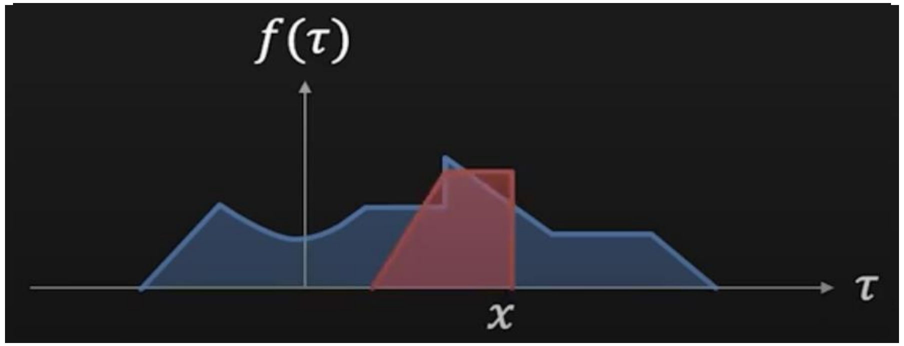
Image Filtering in the Frequency Domain

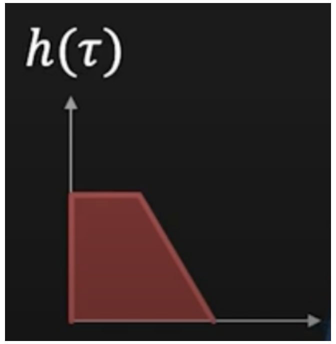
Fourier transform and convolution

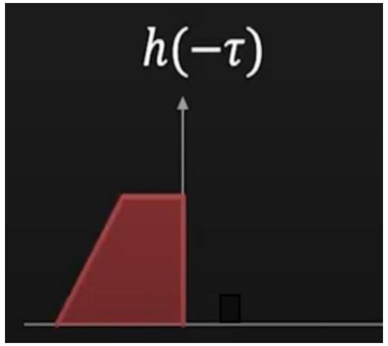
- A close and important relationship.

Convolution of 2 functions: *f(x)* and *h(x)*

$$g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)\, h(x-\tau)\; d\tau$$







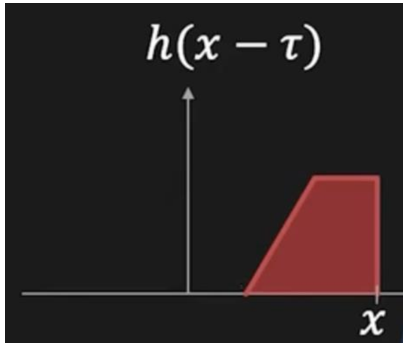


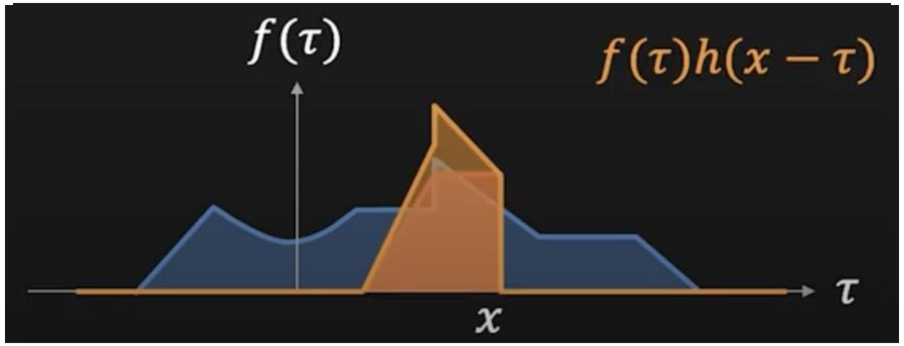
Image Filtering in the Frequency Domain

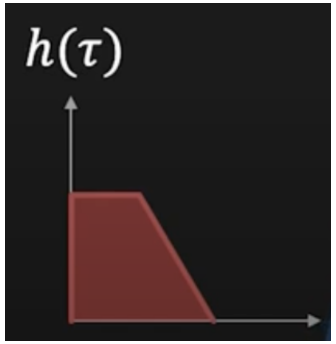
Fourier transform and convolution

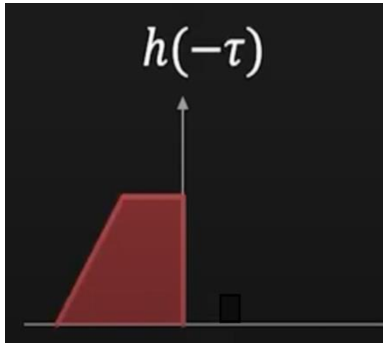
- A close and important relationship.

Convolution of 2 functions: *f(x)* and *h(x)*

$$g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)\, h(x-\tau)\ d\tau$$







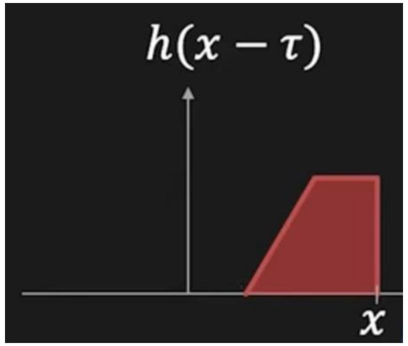


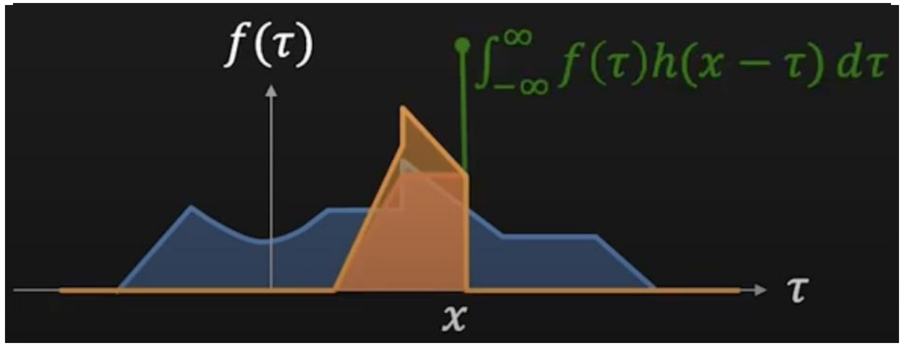
Image Filtering in the Frequency Domain

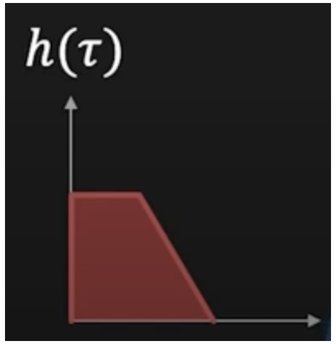
Fourier transform and convolution

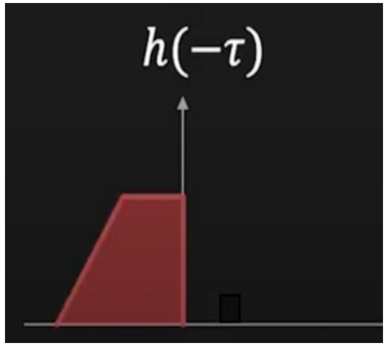
- A close and important relationship.

Convolution of 2 functions: *f(x)* and *h(x)*

$$g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)\, h(x-\tau)\; d\tau$$







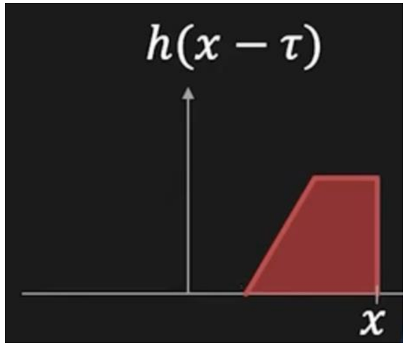


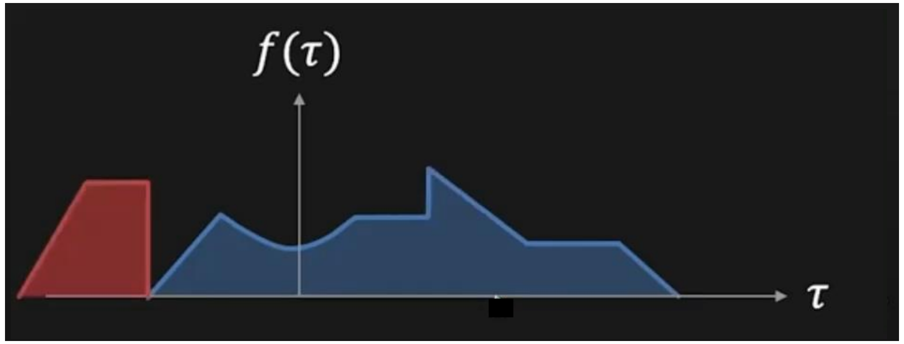
Image Filtering in the Frequency Domain

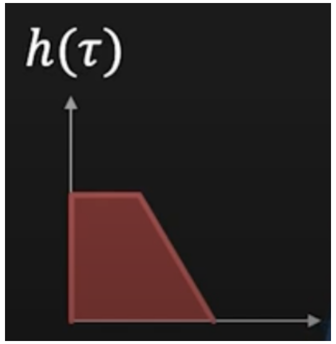
Fourier transform and convolution

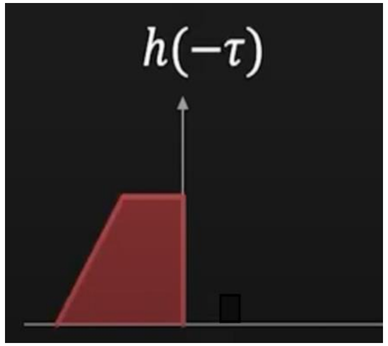
- A close and important relationship.

Convolution of 2 functions: *f(x)* and *h(x)*

$$g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)\, h(x-\tau)\; d\tau$$







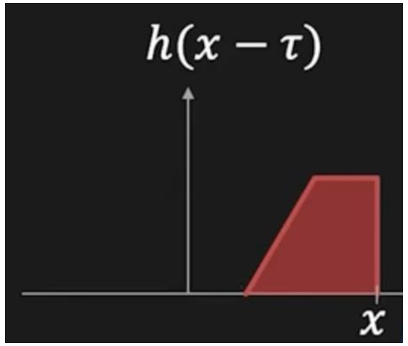
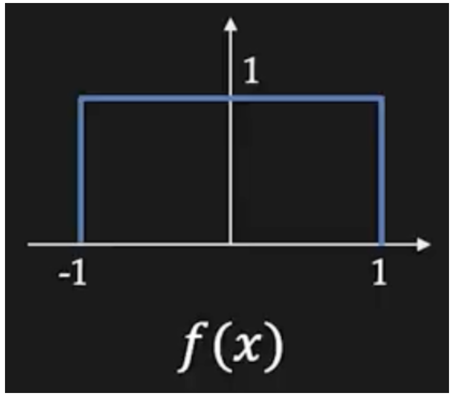
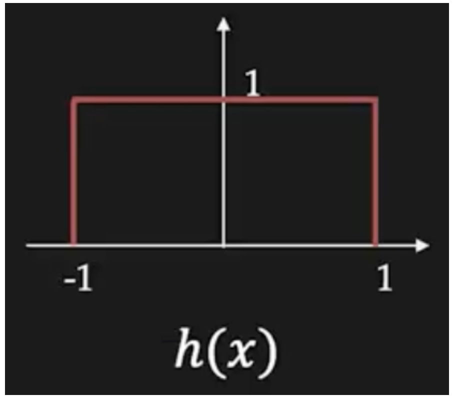


Image Filtering in the Frequency Domain

Fourier transform and convolution: Few examples





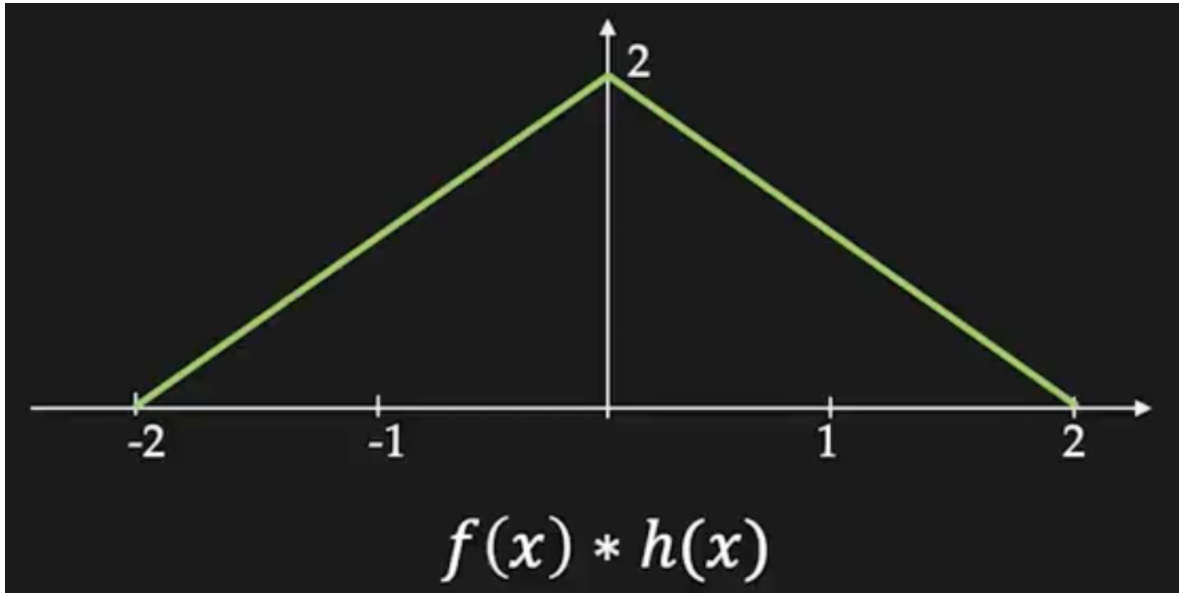


Image Filtering in the Frequency Domain

Fourier transform and convolution

$\Rightarrow$ The convolution theorem

Convolution: $g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)\, h(x-\tau)\ d\tau$

Fourier Transform of $g(x)$:

$$\mathrm{G}(u) = \int_{-\infty}^{\infty} g(x)\, e^{-i2\pi ux}\ dx = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\tau)\, h(x-\tau)\, e^{-i2\pi ux}\, d\tau\, dx$$

$$\mathrm{G}(u) = \underbrace{\int_{-\infty}^{\infty} f(\tau)\, e^{-i2\pi u\tau}\ d\tau}_{F(u)}\ \underbrace{\int_{-\infty}^{\infty} h(x-\tau)\, e^{-i2\pi u(x-\tau)}\, dx}_{H(u)} = F(u)\, H(u)$$

Image Filtering in the Frequency Domain

Fourier transform and convolution

Spatial Domain		Frequency Domain
$g(x) = f(x) * h(x)$ Convolution	$\longleftrightarrow$	$G(u) = F(u)\,H(u)$ Multiplication
$g(x) = f(x)\,h(x)$ Multiplication	$\longleftrightarrow$	$G(u) = F(u) * H(u)$ Convolution

Image Filtering in the Frequency Domain

Fourier transform and convolution

- Very efficient implementation of FT and IFT are available.
- Working in the frequency domain is more efficient.
- Analyze in the frequency domain, a filter that is designed in the spatial domain.

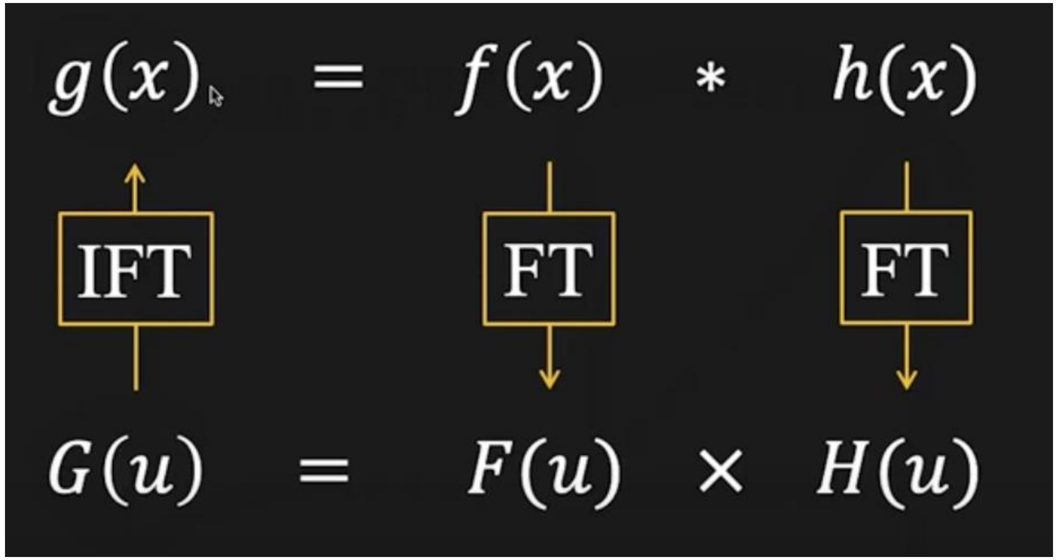
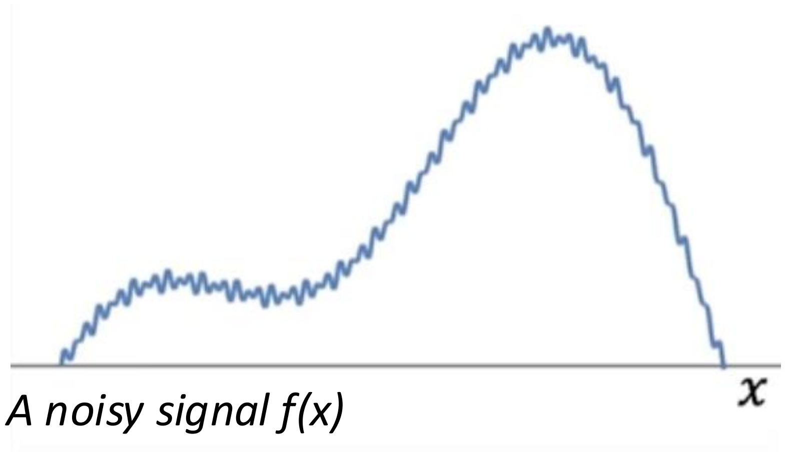
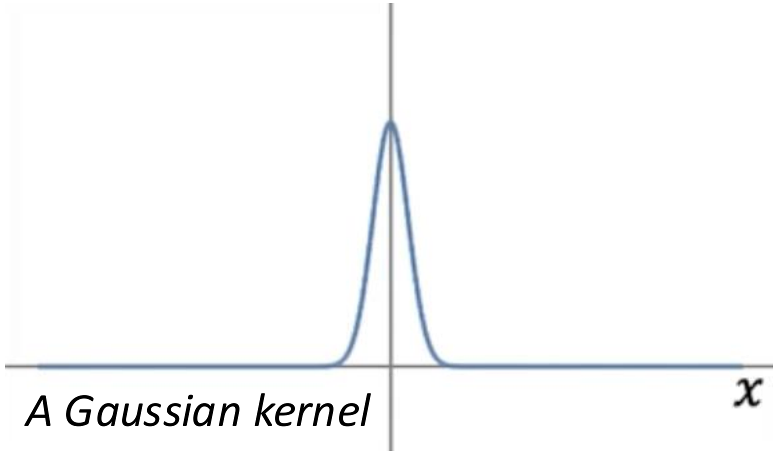


Image Filtering in the Frequency Domain

Fourier transform and convolution



A noisy signal f(x)



A Gaussian kernel

$\Rightarrow$ Convolving the noisy signal with the Gaussian kernel.

Image Filtering in the Frequency Domain

Fourier transform and convolution

⇒ Convolving the noisy signal with the Gaussian kernel.

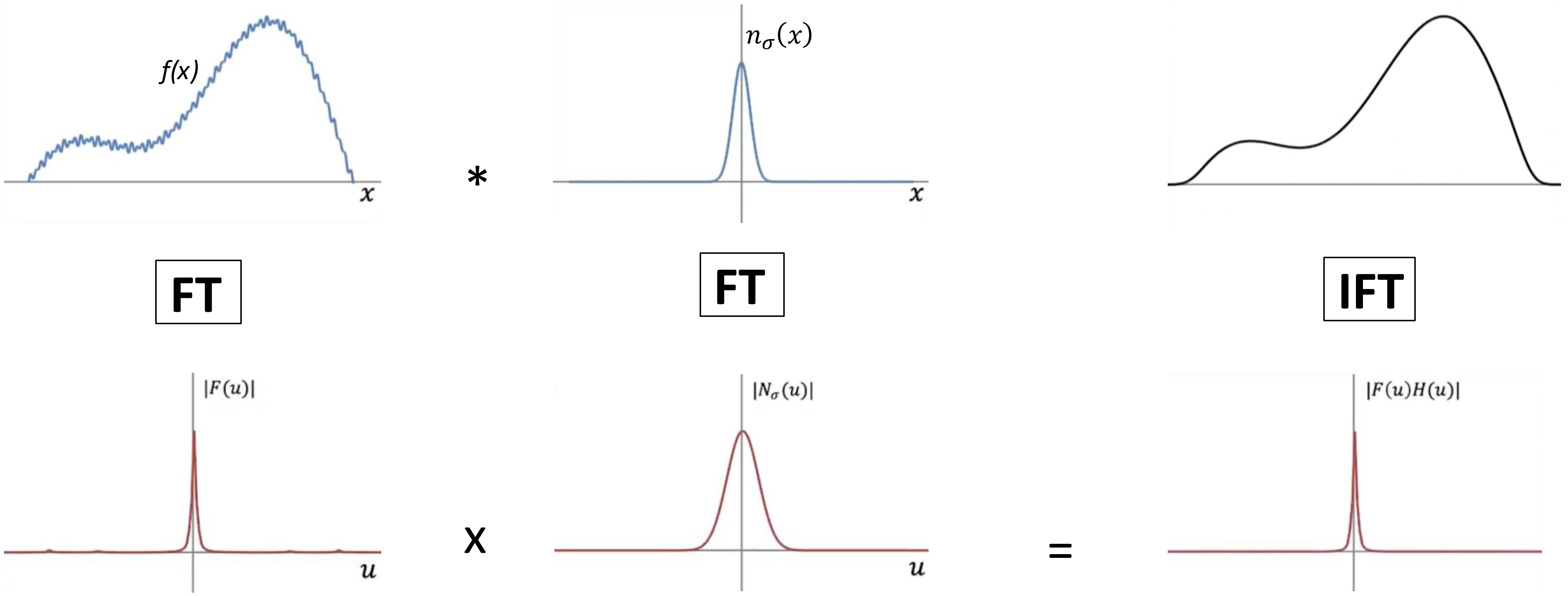


Image Filtering in the Frequency Domain

Application to images – The 2D Fourier transform

Fourier Transform: $F(u, v) = \iint_{-\infty}^{\infty} f(x, y)\, e^{-i2\pi(ux+vy)} dx\, dy$

u and *v* are the frequencies along x and y, respectively.

Inverse Fourier Transform: $f(x, y) = \iint_{-\infty}^{\infty} F(u, v)\, e^{i2\pi(ux+vy)} du\, dv$

Image Filtering in the Frequency Domain

Application to images – The 2D Discrete Fourier transform (DFT)

Discrete Fourier Transform:

$$F[p,q] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f[m,n] e^{-i2\pi pm/M} e^{-i2\pi qn/N}$$

$p = 0 \ldots M-1$ and $q = 0 \ldots N-1$,

p and *q* are the frequencies along m and n, respectively.

Image Filtering in the Frequency Domain

Application to images – The 2D Discrete Fourier transform (DFT)

Inverse Discrete Fourier Transform:

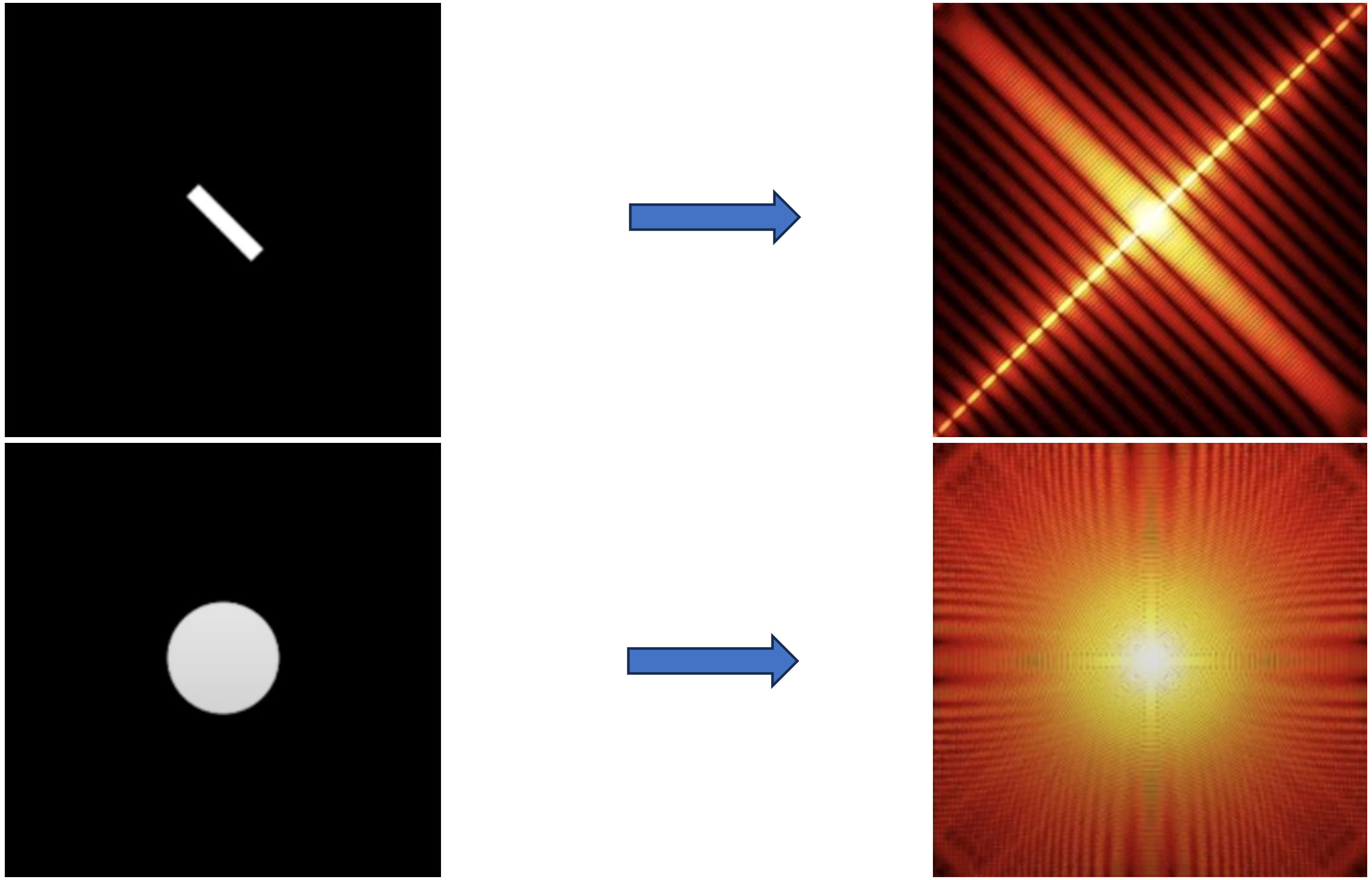
$$f[m,n] = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} F[p,q]e^{i2\pi pm/M}e^{i2\pi qn/N}$$

$m = 0 \ldots M-1$ and $n = 0 \ldots N-1$

Image Filtering in the Frequency Domain

Application to images – The 2D Discrete Fourier transform (DFT)

f(m, n) *log(|F(p, q)|)*



Min Max

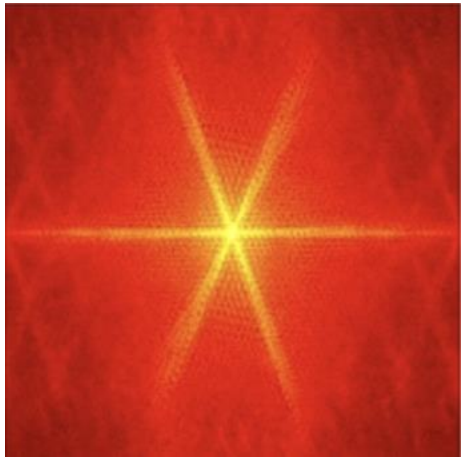
Color to show the magnitude

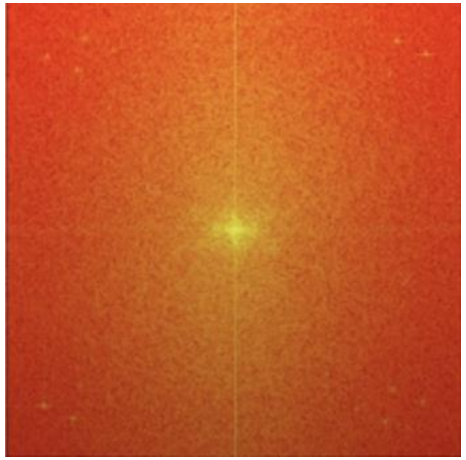
Image Filtering in the Frequency Domain

Application to images – The 2D Discrete Fourier transform (DFT)

f(m, n)

log(|F(p, q)|)





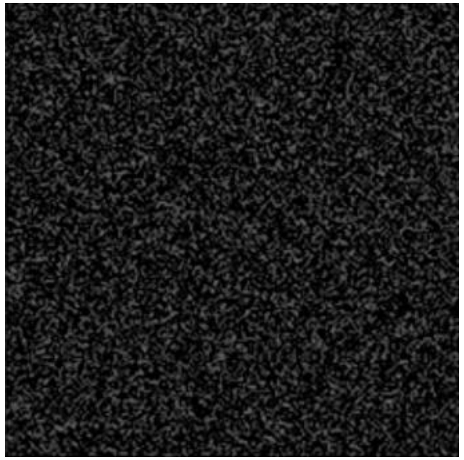
Min Max

Color to show the magnitude

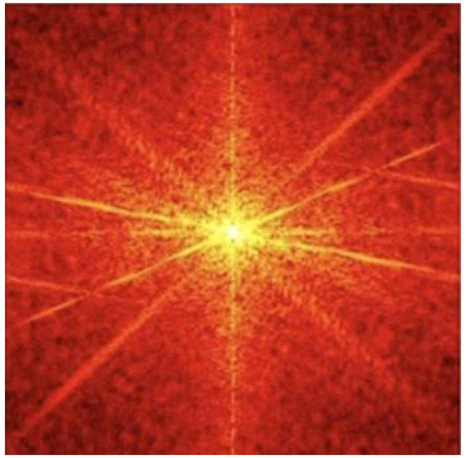
Image Filtering in the Frequency Domain

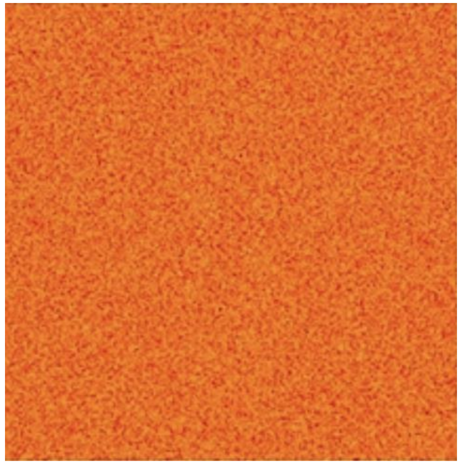
Application to images – The 2D Discrete Fourier transform (DFT)

f(m, n)



log(|F(p, q)|)





Min Max

Color to show the magnitude

Image Filtering in the Frequency Domain

Application to image filtering: The low pass filter and the high pass filter

- Low frequencies in the transform are related to slowly varying intensity components.
- High frequencies are caused by sharp transitions in intensity, such as edges, corners and noise

⇒ A filter ***H(u, v)*** that attenuates high frequencies while passing low frequencies (low-pass filter) blurs an image

⇒ A high-pass filter (which attenuates low frequencies) enhances sharp detail, but cause a reduction in contrast in the image.

Image Filtering in the Frequency Domain

Application to image filtering: The low pass filter

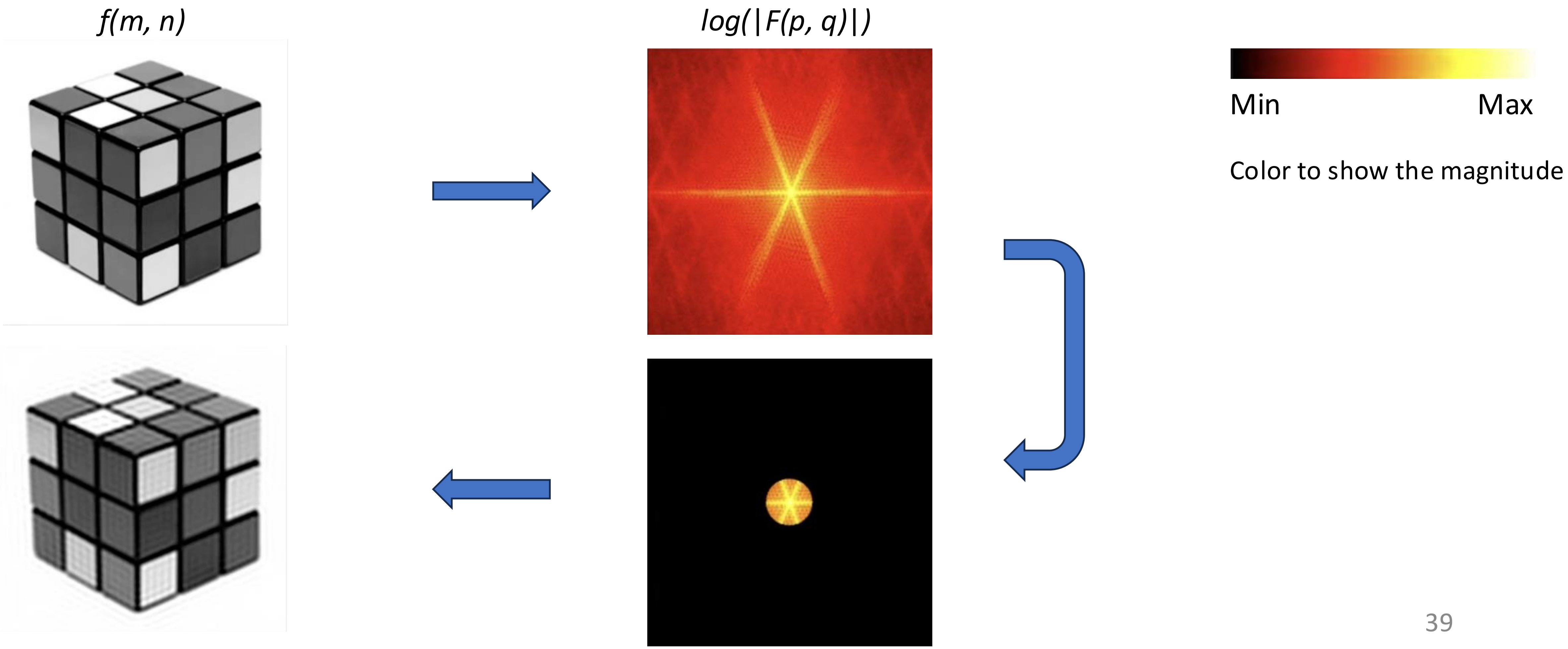
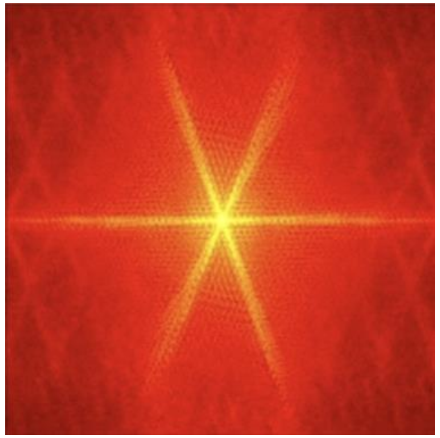


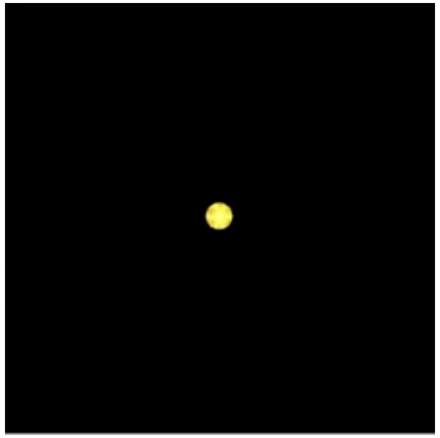
Image Filtering in the Frequency Domain

Application to image filtering: The low pass filter

f(m, n)

log(|F(p, q)|)





Min Max

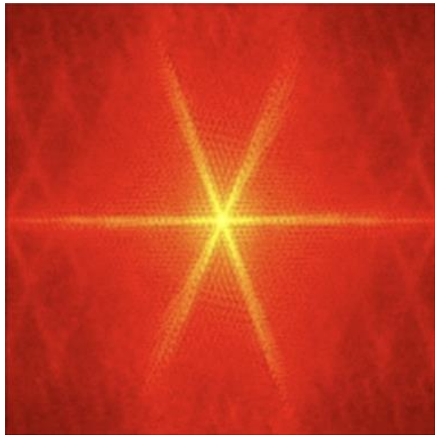
Color to show the magnitude

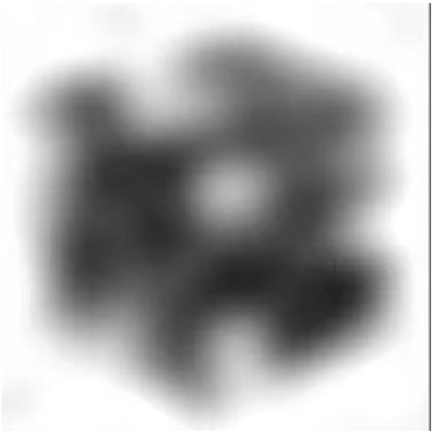
Image Filtering in the Frequency Domain

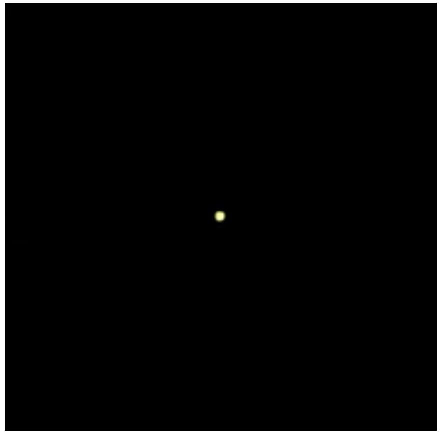
Application to image filtering: The low pass filter

f(m, n)

log(|F(p, q)|)







Min Max

Color to show the magnitude

Image Filtering in the Frequency Domain

Application to image filtering: The high pass filter

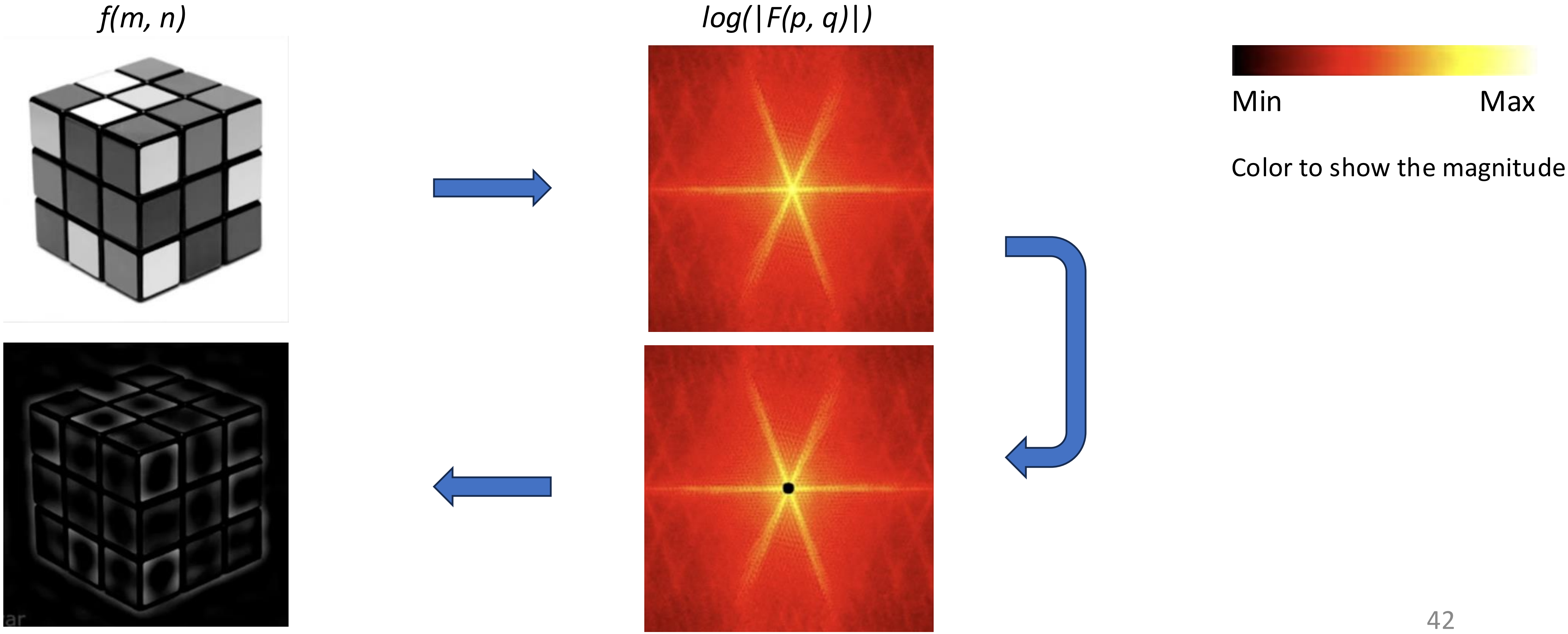
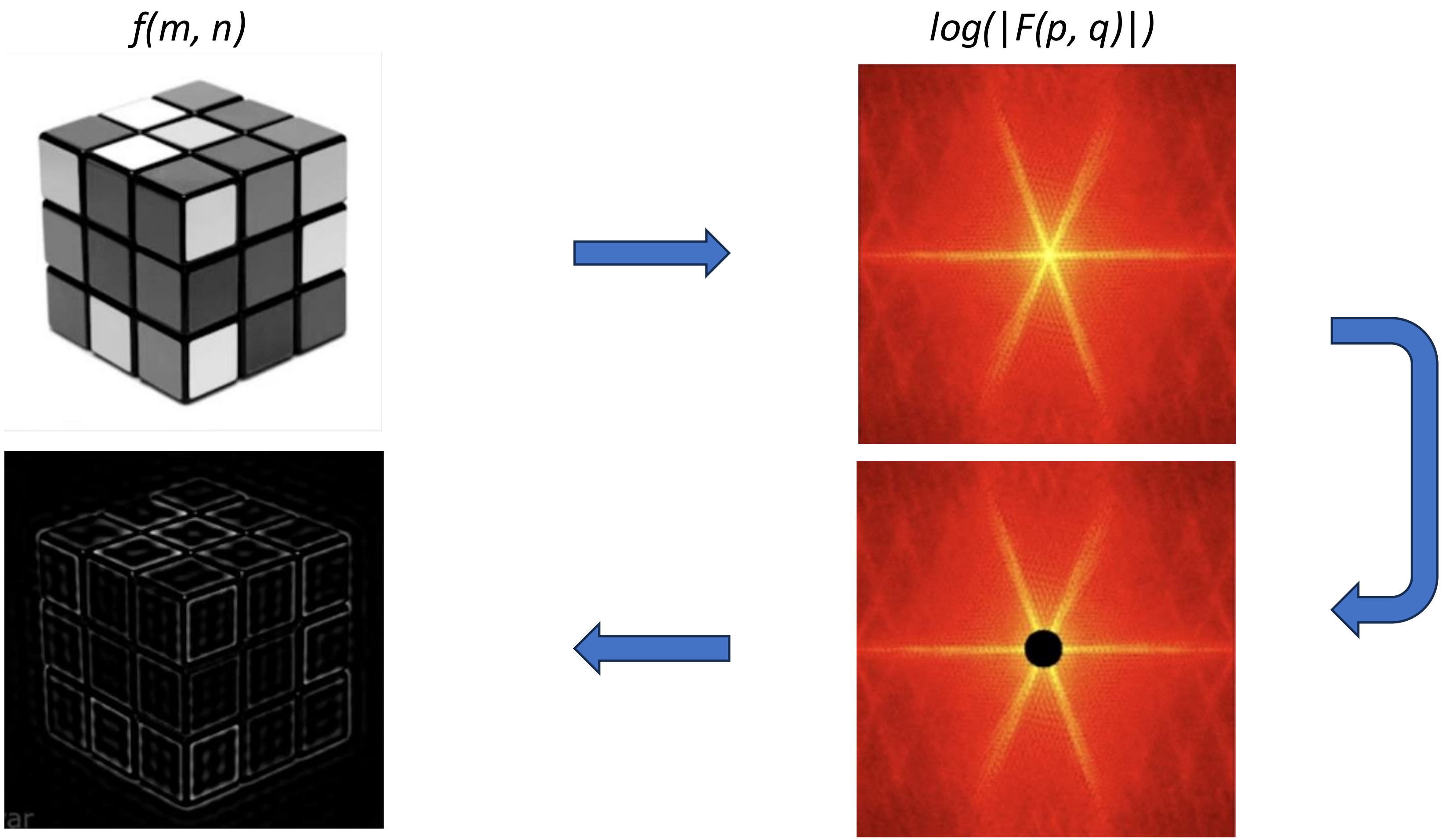


Image Filtering in the Frequency Domain

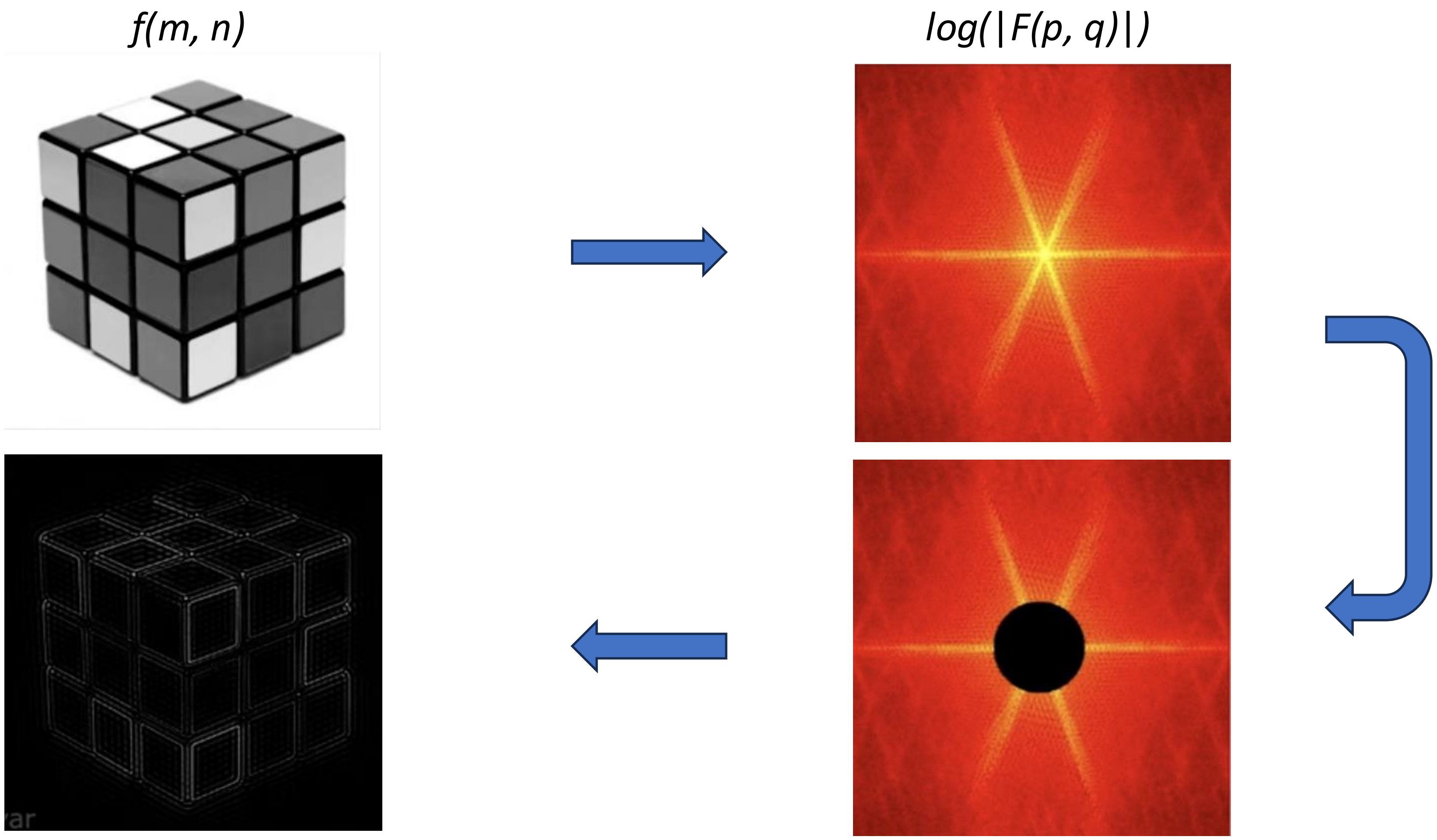
Application to image filtering: The high pass filter



Color to show the magnitude

Image Filtering in the Frequency Domain

Application to image filtering: The high pass filter



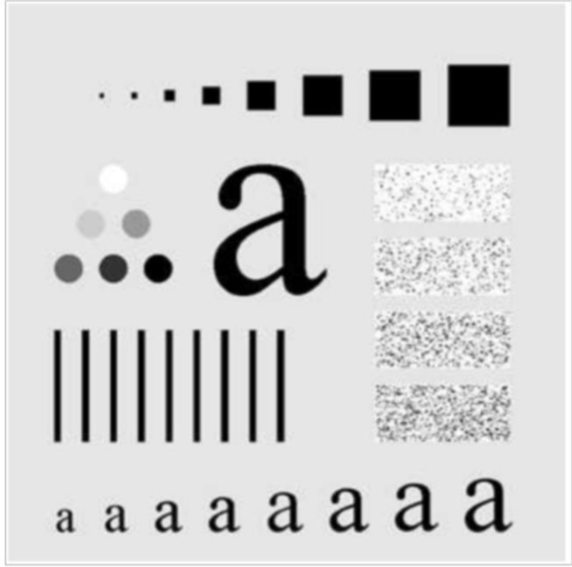
Color to show the magnitude

Image Filtering in the Frequency Domain

Low-pass filter seen so far is called Ideal Low-pass filter (ILPF):

$$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \le D_0 \\ 0 & \text{if } D(u, v) > D_0 \end{cases}$$

The point of transition between $H(u, v) = 1$ and $H(u, v) = 0$ is called the **cut-off frequency**.



ILPF with cut-off frequency = 60

Image Filtering in the Frequency Domain

Low-pass filter seen so far is called Ideal Low-pass filter (ILPF):

$$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \le D_0 \\ 0 & \text{if } D(u, v) > D_0 \end{cases}$$

The point of transition between $H(u, v) = 1$ and $H(u, v) = 0$ is called the **cut-off frequency**.

ILPF with cut-off frequency = 30

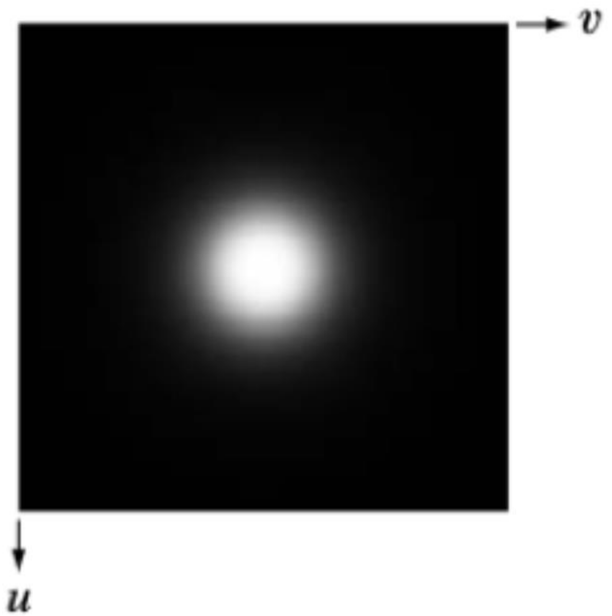
Image Filtering in the Frequency Domain

The butterworth Low-pass filters

$$H(u,v) = \frac{1}{1 + [D(u,v)/D_0]^{2n}}$$

n is the order of the filter.

D_0 is the cutoff frequency.



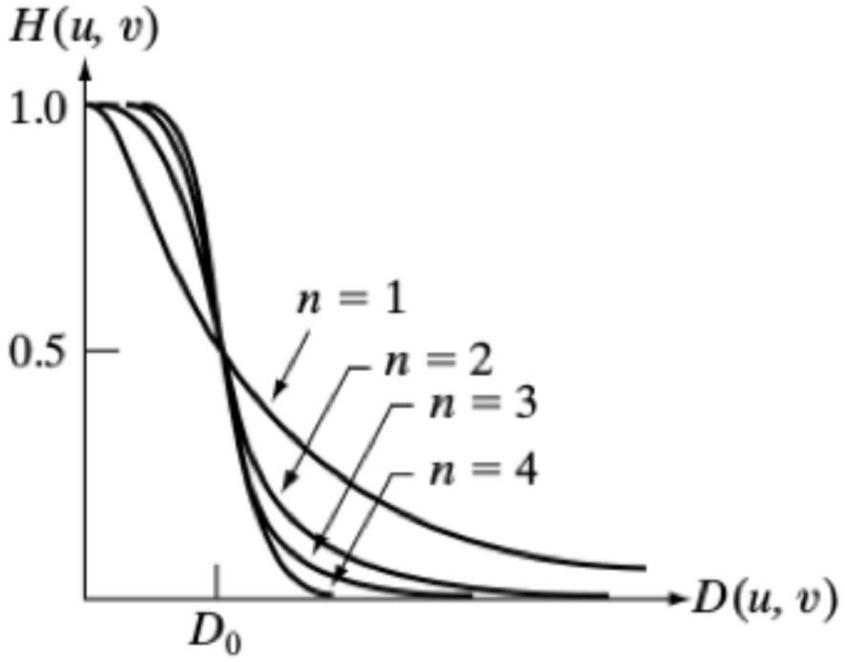
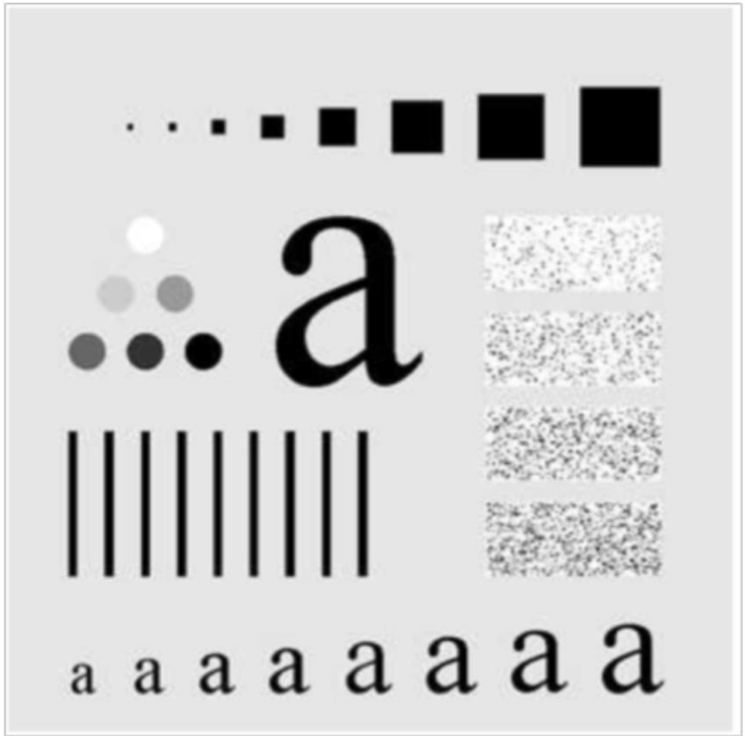


Image Filtering in the Frequency Domain

The butterworth Low-pass filters

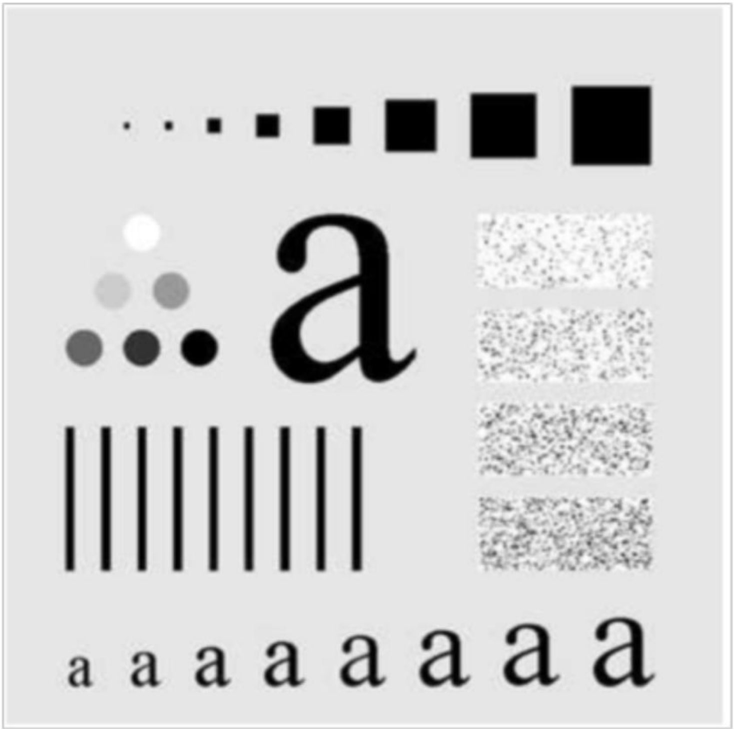


Order of the filter (n) = 2.

Cutoff frequency (D_0) = 60.

Image Filtering in the Frequency Domain

The butterworth Low-pass filters



Order of the filter (n) = 2.

Cutoff frequency (D_0) = 30.

Image Filtering in the Frequency Domain

Application to image filtering: The Gaussian filtering

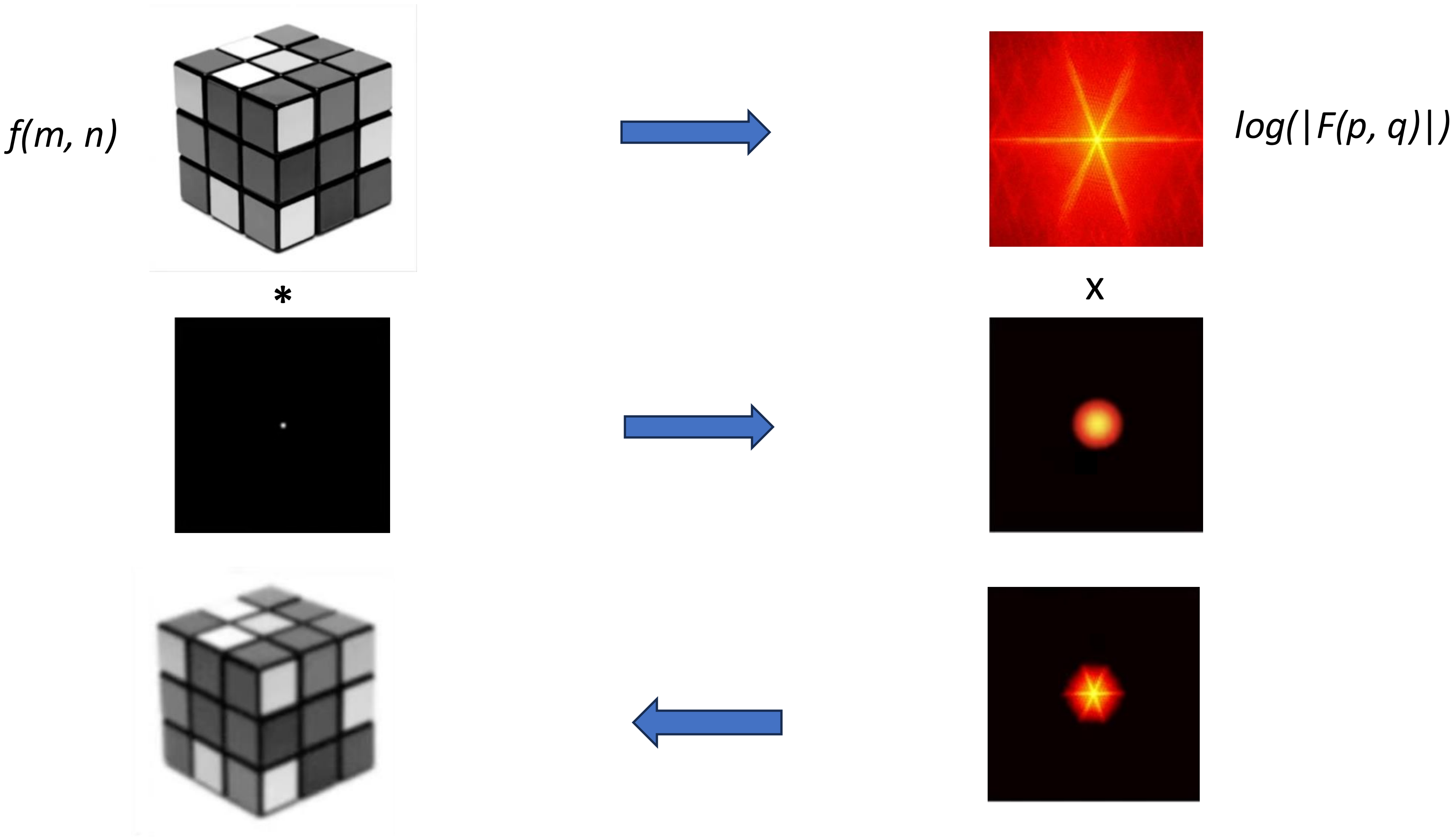


Image Filtering in the Frequency Domain

Application to image filtering: The Gaussian filtering

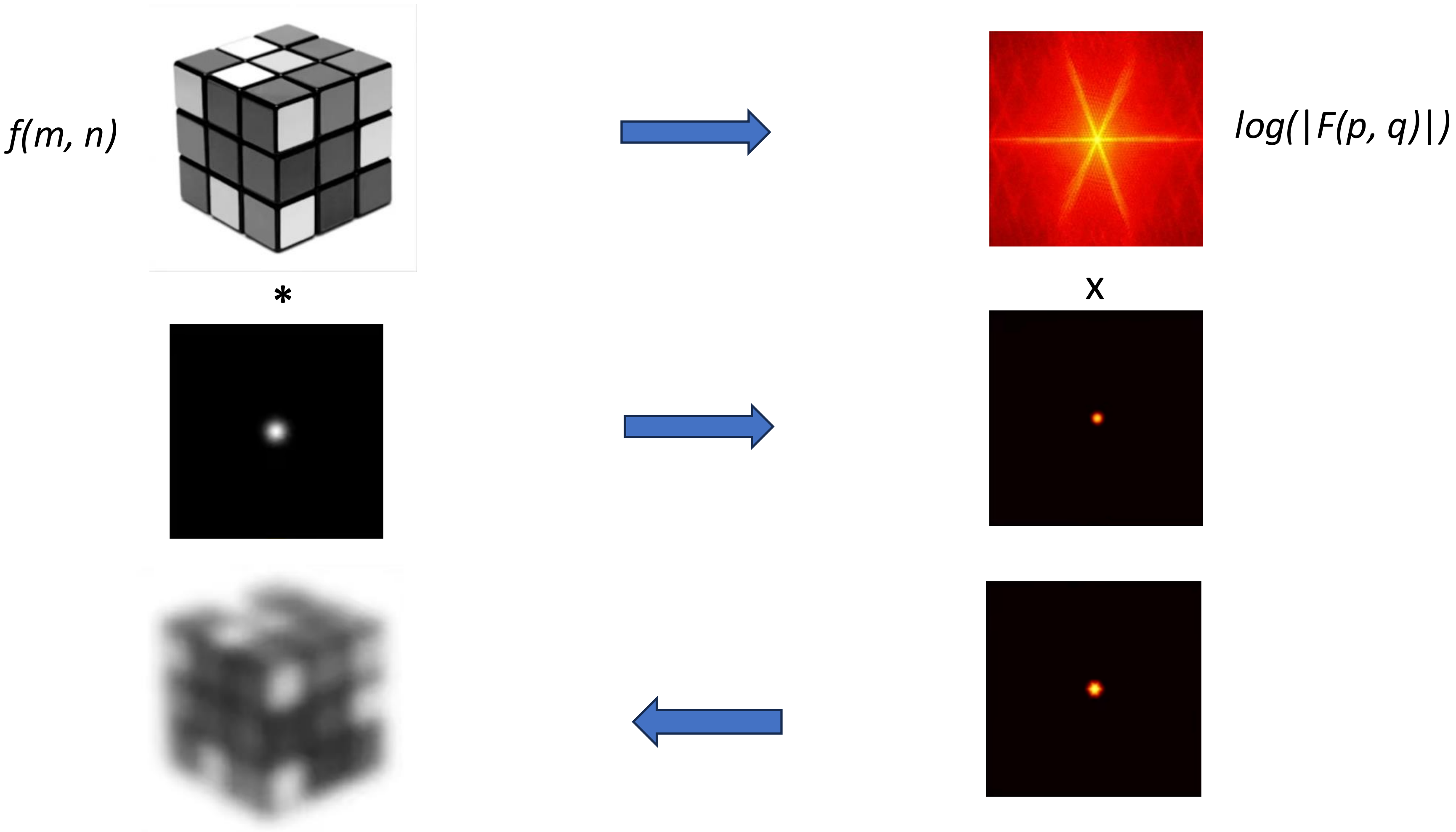


Image Filtering in the Frequency Domain

Application to image filtering: The Gaussian filtering

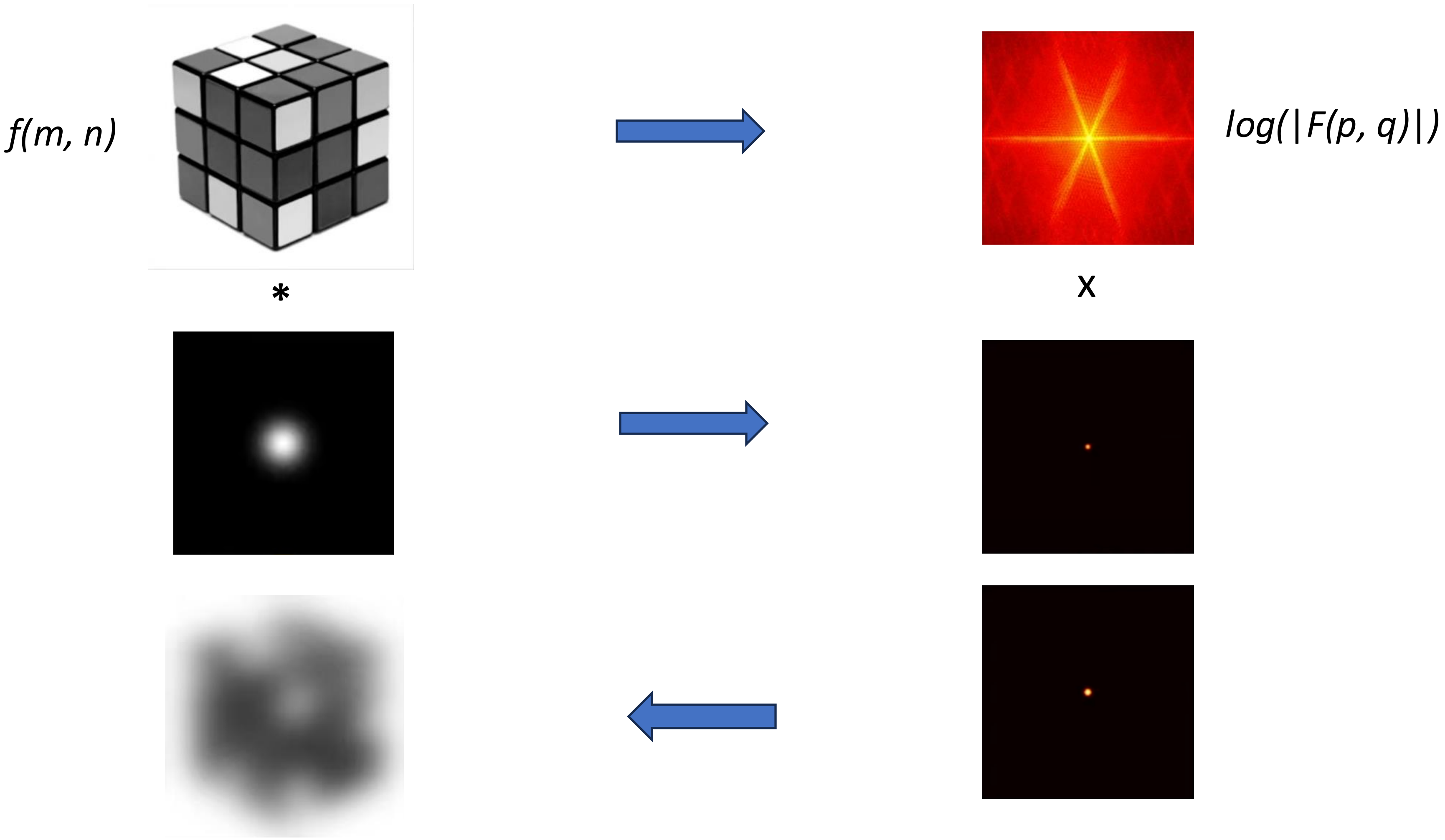
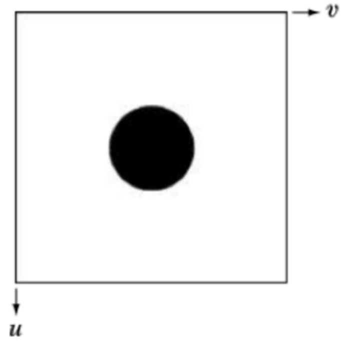
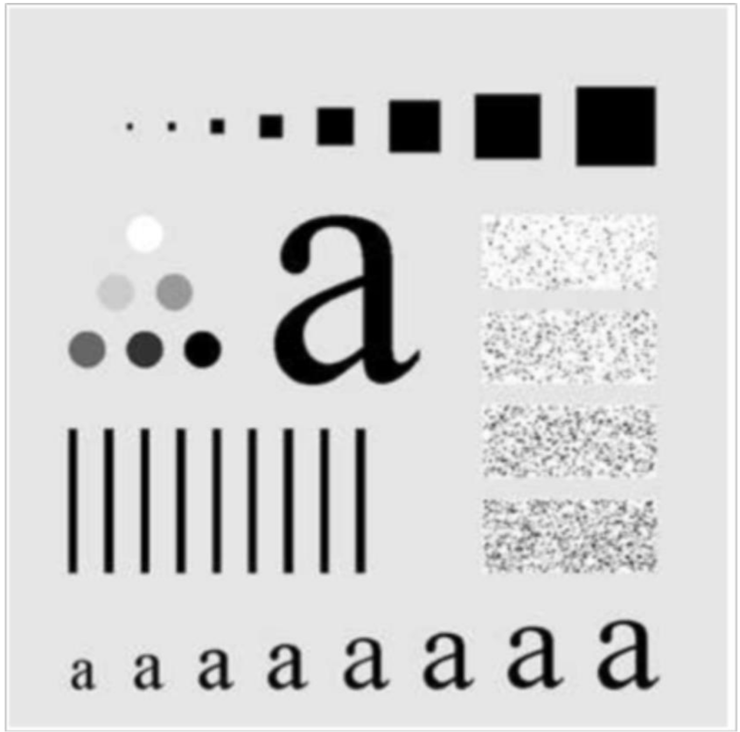


Image Filtering in the Frequency Domain

High-pass filter seen so far is called Ideal High-pass filter (IHPF):

$$H(u, v) = \begin{cases} 0 & \text{if } D(u, v) \leq D_0 \\ 1 & \text{if } D(u, v) > D_0 \end{cases}$$

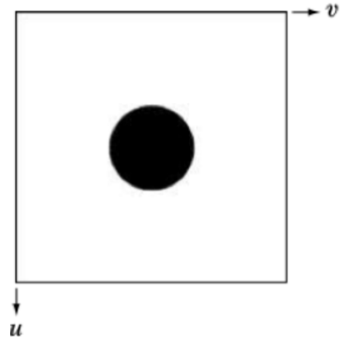




IHPF with cut-off frequency = 60

Image Filtering in the Frequency Domain

High-pass filter seen so far is called Ideal High-pass filter (IHPF):



$$H(u, v) = \begin{cases} 0 & \text{if } D(u, v) \leq D_0 \\ 1 & \text{if } D(u, v) > D_0 \end{cases}$$

IHPF with cut-off frequency = 30

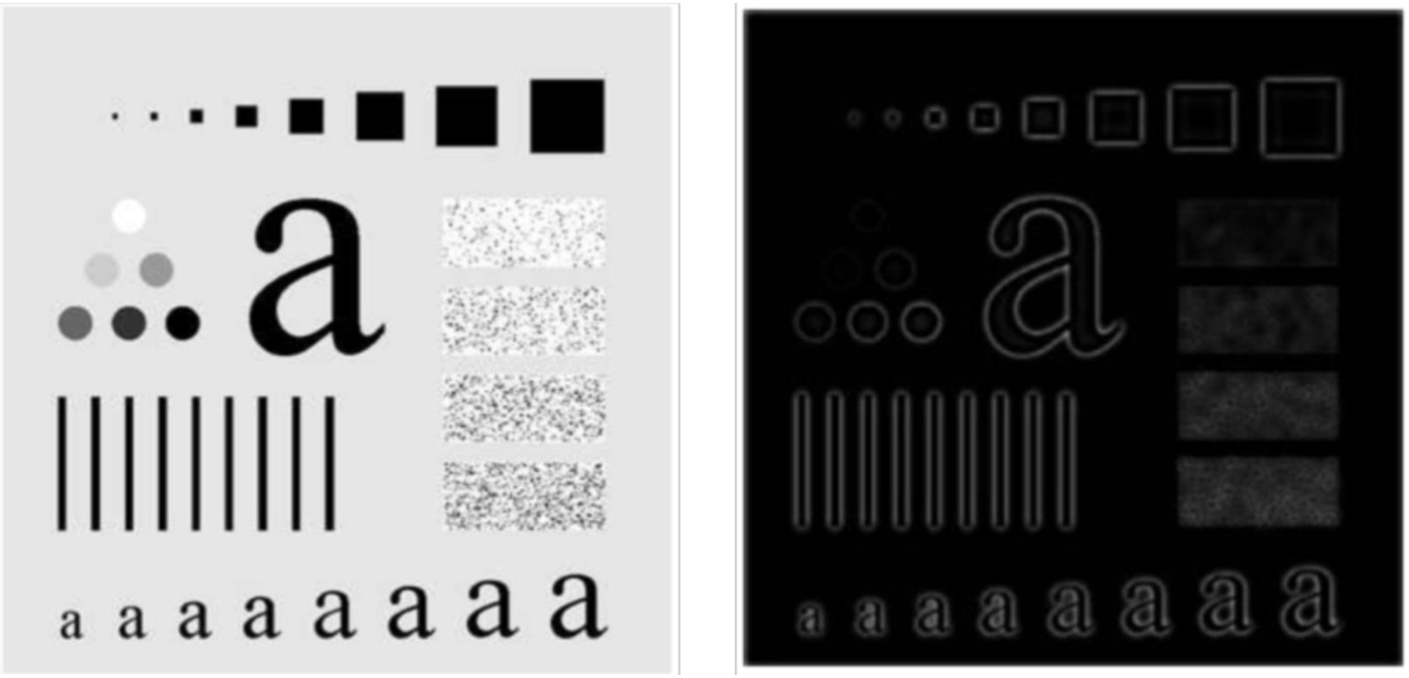
Image Filtering in the Frequency Domain

The butterworth High-pass filters

$$H(u, v) = \frac{1}{1 + [D_0/D(u, v)]^{2n}}$$

n is the order of the filter.

D_0 is the cutoff frequency.



BHPF with cut-off frequency = 60

Image Filtering in the Frequency Domain

The butterworth High-pass filters

$$H(u, v) = \frac{1}{1 + [D_0/D(u, v)]^{2n}}$$

n is the order of the filter.

D_0 is the cutoff frequency.

BHPF with cut-off frequency = 30

Image Filtering in the Frequency Domain

In all of the previous examples, we have used only the magnitude.

Filtering has been performed considering only the magnitude.

What about the phase?

In many cases, the phase is more important than the amplitude to preserve the visual information.

Image Filtering in the Frequency Domain

In general,

- The amplitudes of the sinusoids determine the intensities in the image.

- The phase is a measure of displacement of the various sinusoids with respect to their origin.

    Carry much of the information about where discernable objects are located

Image Filtering in the Frequency Domain

Original image

Apply DFT

Set all the phases to 0

Preserve the magnitude

Reconstruct the image using the IDFT

Reconstructed image

Réf. : *Signal reconstruction from Fourier transform sign information.* S. Curtis, S. Lim, A. Oppenheim. 1984. https://dspace.mit.edu/handle/1721.1/4243

Image Filtering in the Frequency Domain

Original image

Apply DFT

Preserve the phases

Using an average of magnitudes computed on a set of images

Reconstruct the image using the IDFT

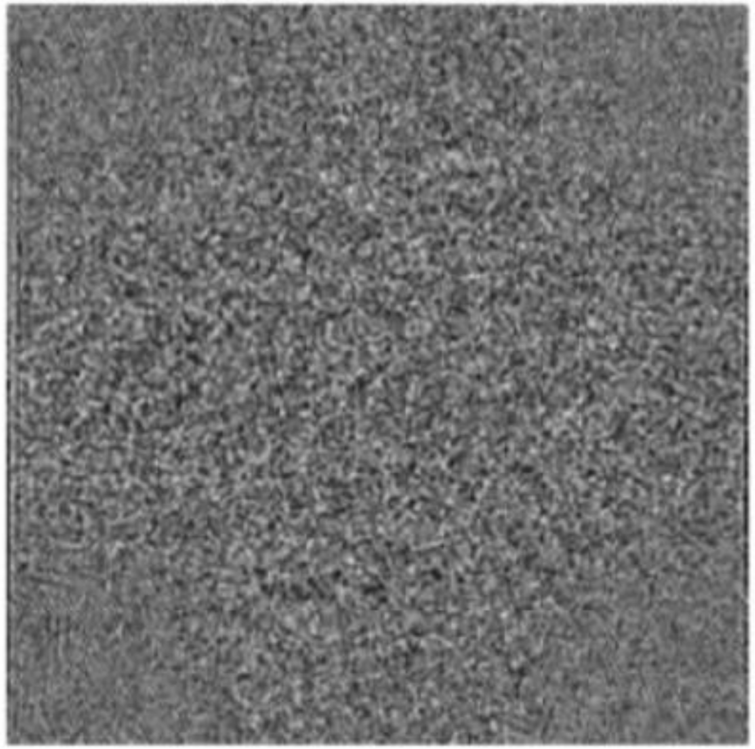
Reconstructed image

Réf. : *Signal reconstruction from Fourier transform sign information.* S. Curtis, S. Lim, A. Oppenheim. 1984.
https://dspace.mit.edu/handle/1721.1/4243

Image Filtering in the Frequency Domain

Example of image reconstruction using only the phase

The original image

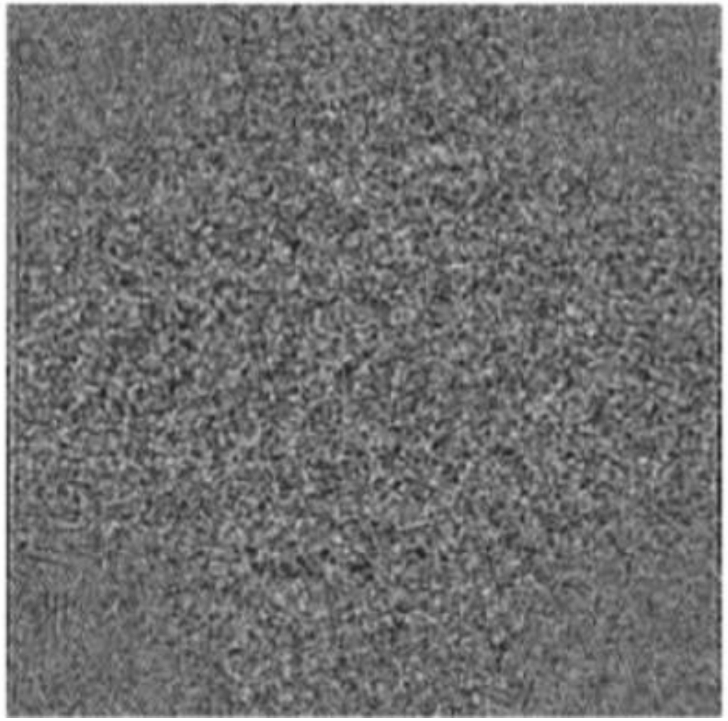


The phase

Image Filtering in the Frequency Domain

Example of image reconstruction using only the amplitude

The original image



The phase

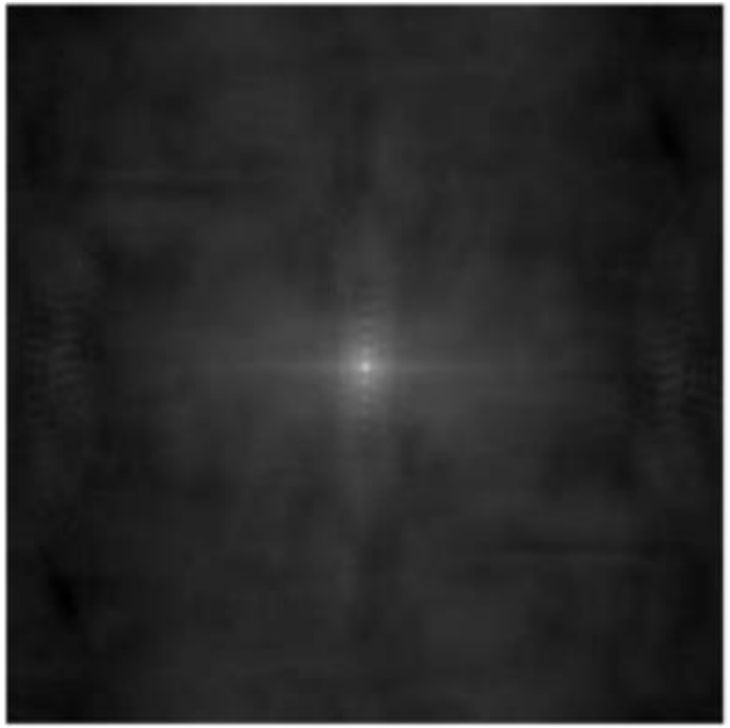


Image Filtering in the Frequency Domain

Example of image reconstruction using the phase and another amplitude

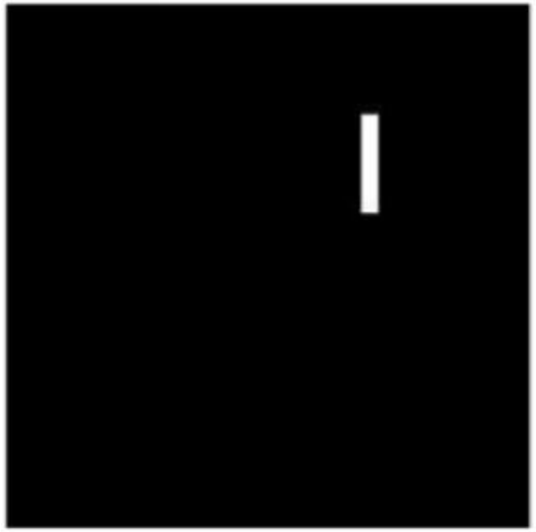


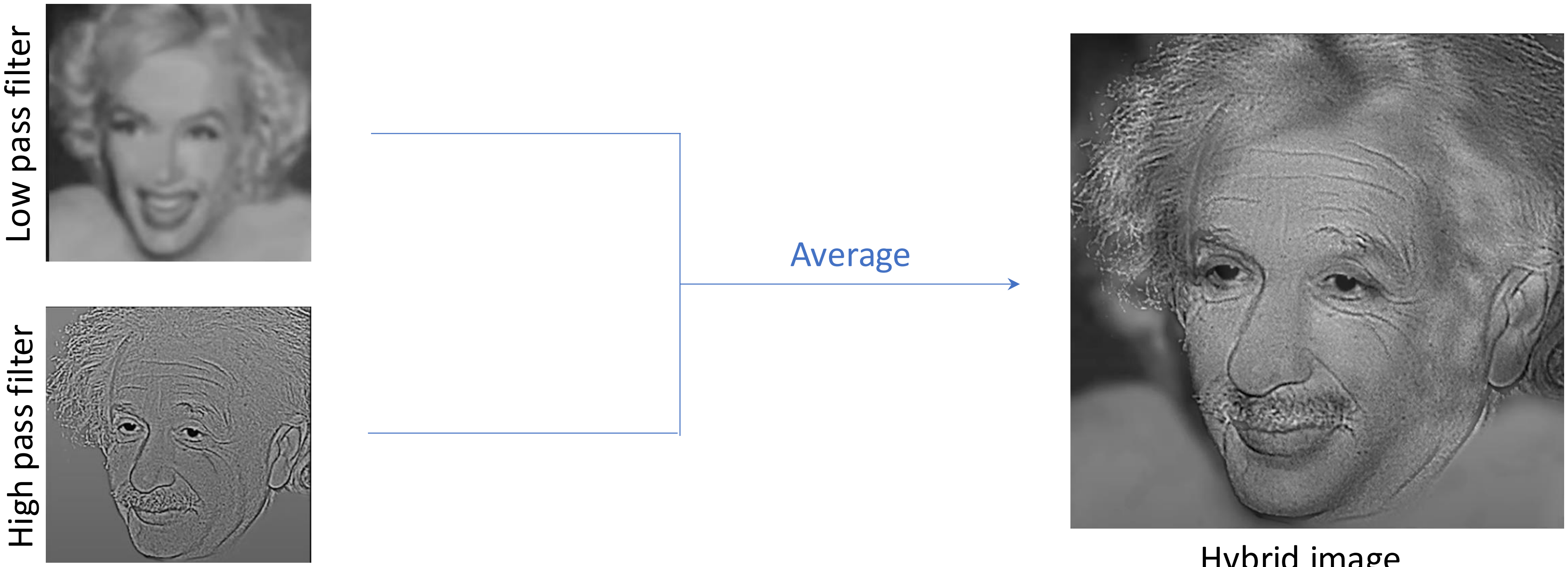
Image from which we compute the amplitude

Image from which we compute the phase

Reconstructed Image

Image Filtering in the Frequency Domain

Hybrid images

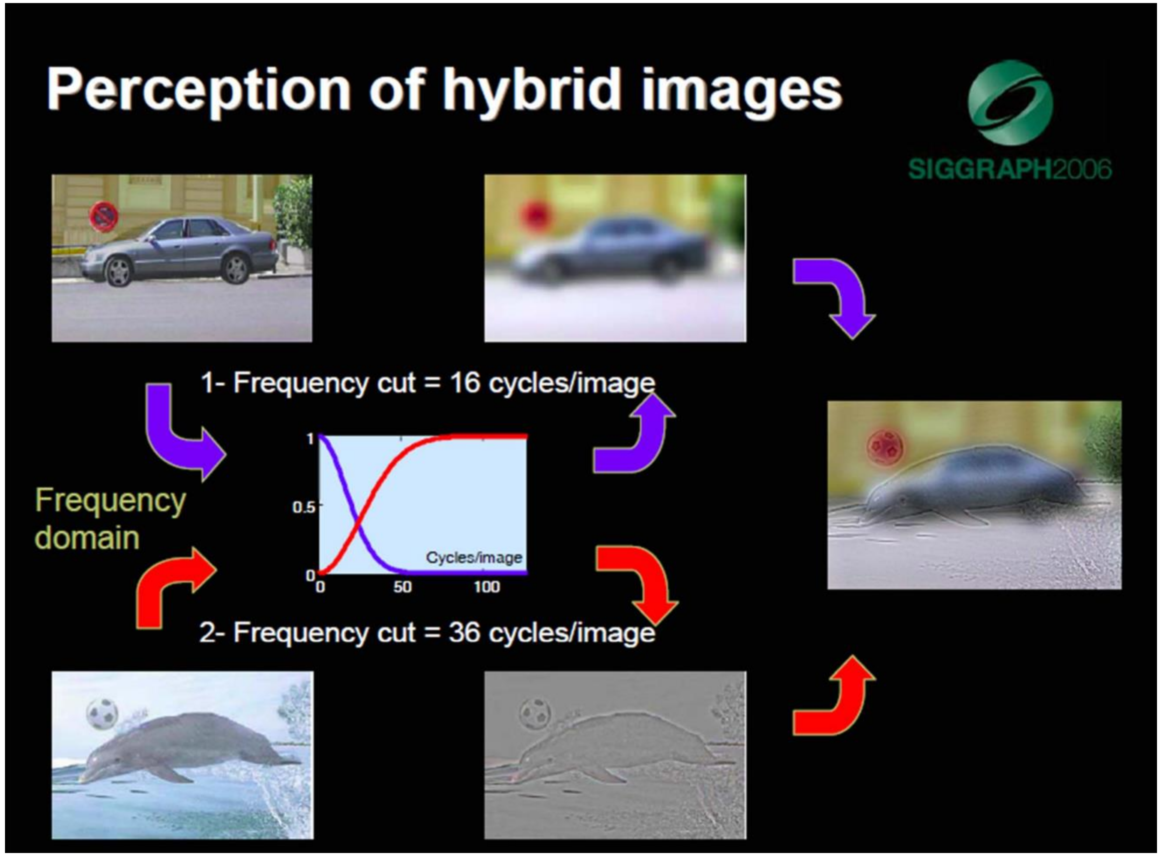


Hybrid image

Réf. : *Hybrid images.* A. Oliva, A. Torralba, P. Schyns. ACM Transactions on Graphics, 2006.
https://stanford.edu/class/ee367/reading/OlivaTorralb_Hybrid_Siggraph06.pdf

Image Filtering in the Frequency Domain

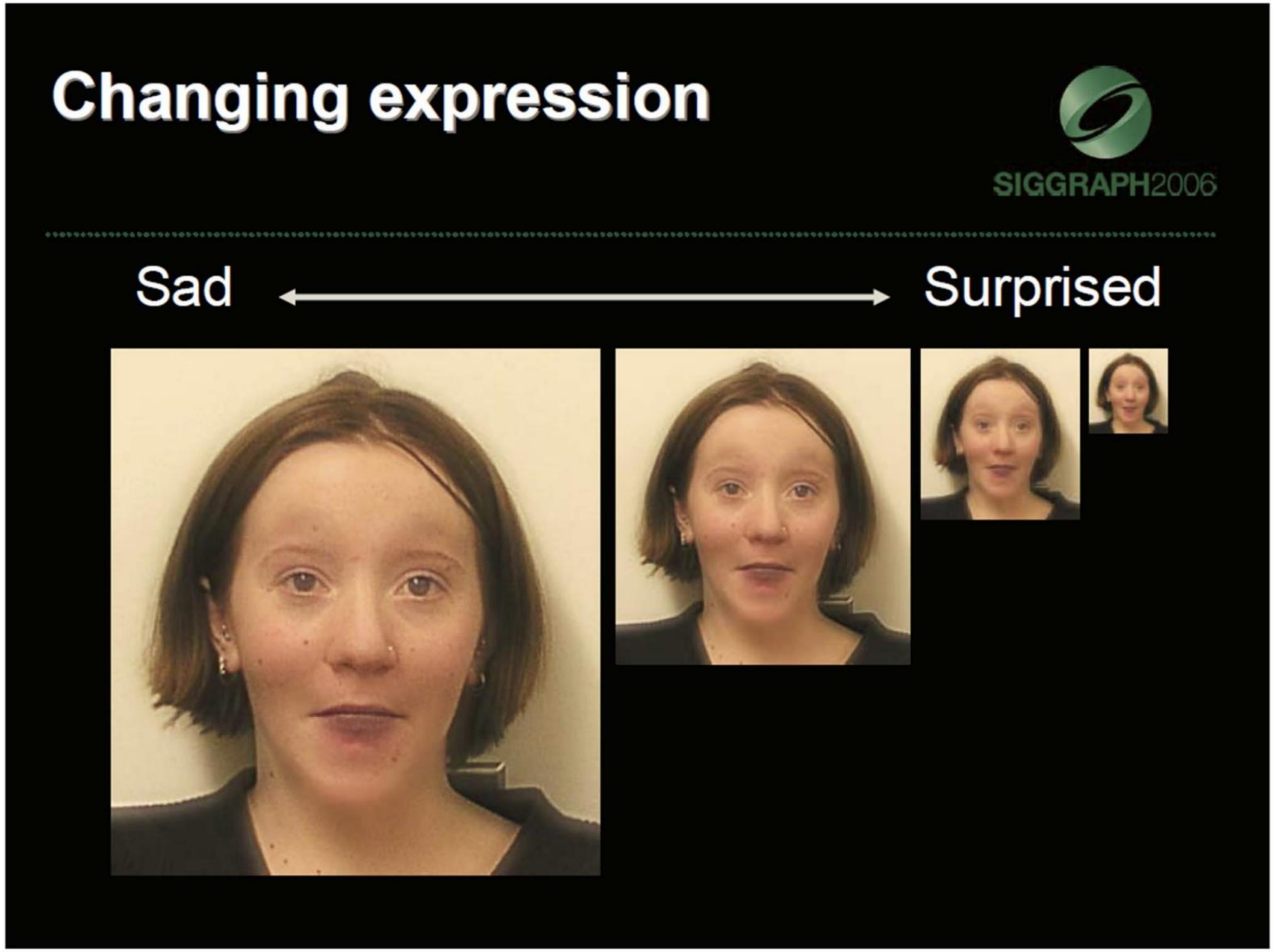
Hybrid images



Réf. : *Hybrid images.* A. Oliva, A. Torralba, P. Schyns. ACM Transactions on Graphics, 2006.
https://stanford.edu/class/ee367/reading/OlivaTorralb_Hybrid_Siggraph06.pdf

Image Filtering in the Frequency Domain

Hybrid images



Aude Oliva & Antonio Torralba & Philippe G Schyns, SIGGRAPH 2006