Image Processing II

Introduction to Computational Photography:

EECS 395/495

Northwestern University

Image Processing I

Transform image to new one that is easier to manipulate.

Topics:

(1) Pixel Processing

(2) Convolution

(3) Linear Filtering

(4) Non-Linear Filtering

(5) Correlation

$\left.\right\}$ Lecture 1

Image Processing II

Transform image to new one that is easier to manipulate.

Topics:

(6) Frequency Representation of Signals

(7) Fourier Transform

(8) Convolution and Fourier Transform

(9) Deconvolution in Frequency Domain

(10) Sampling Theory

Lecture 2

Jean Baptiste Joseph Fourier

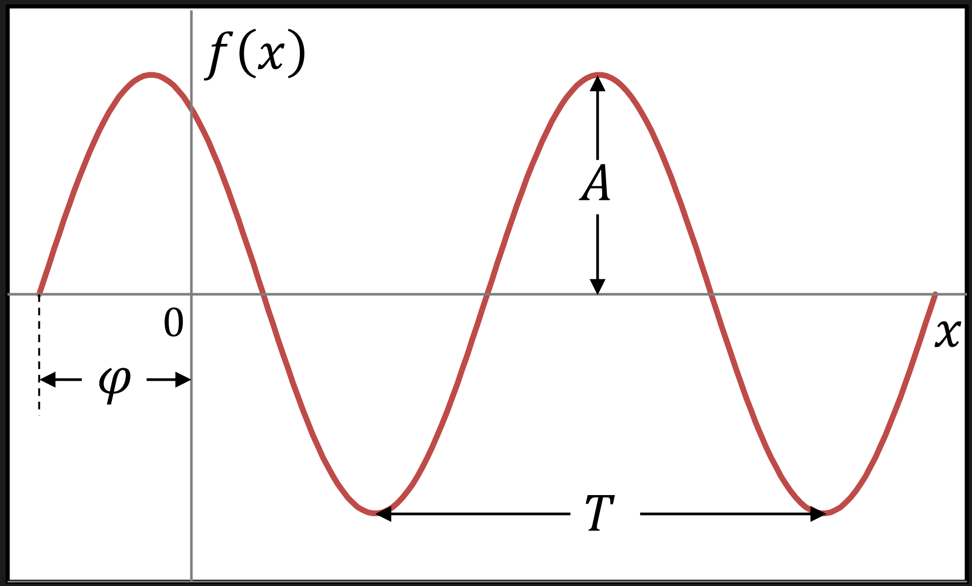
I.1

(1768-1830)

Any Periodic Function can be rewritten as a Weighted Sum of Infinite Sinusoids of Different Frequencies.

Sinusoid

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



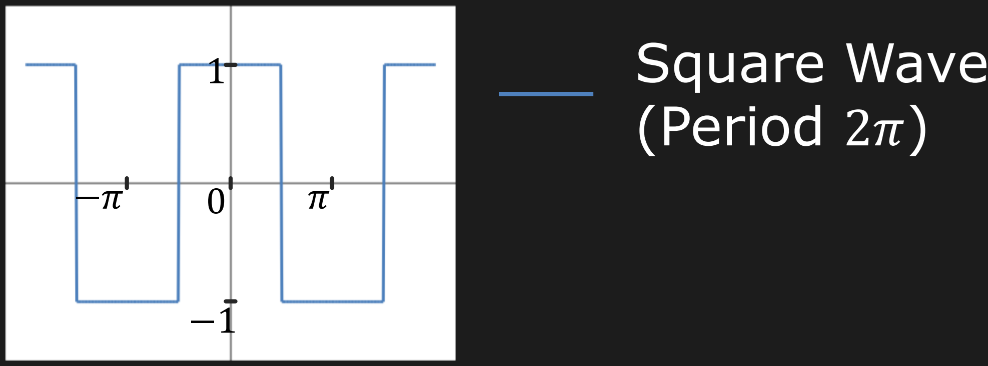
A: Amplitude

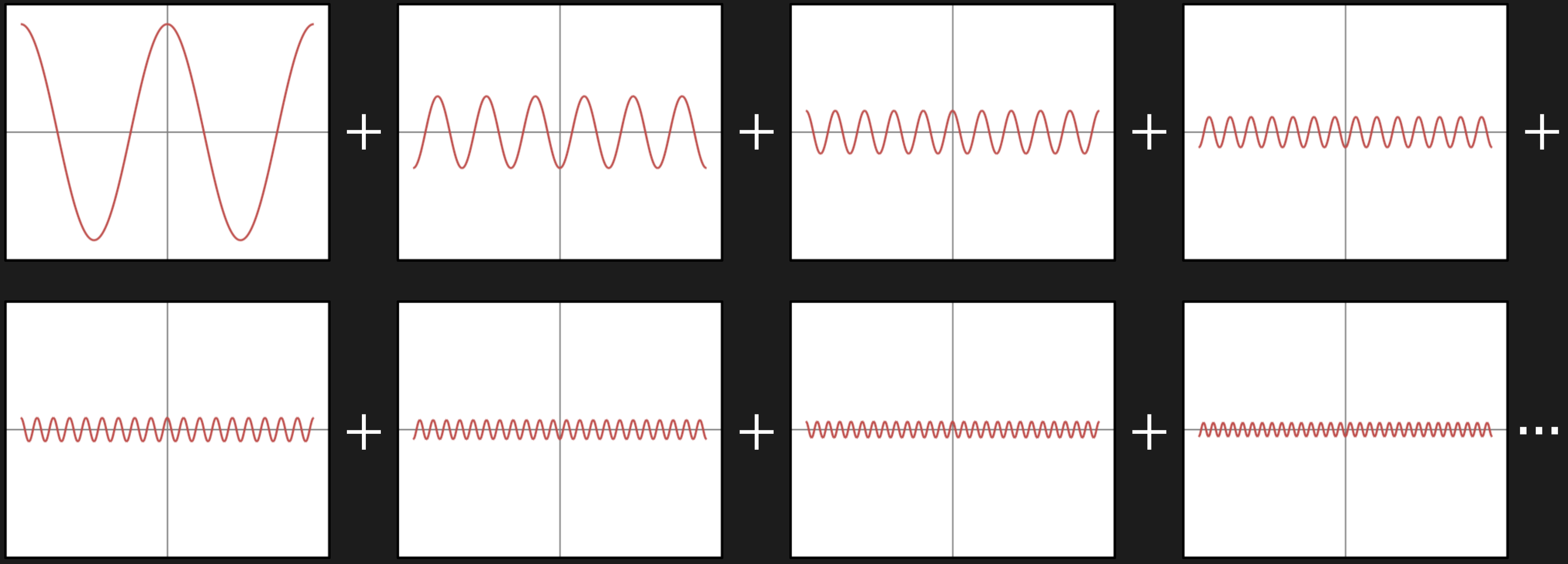
T: Period

φ: Phase

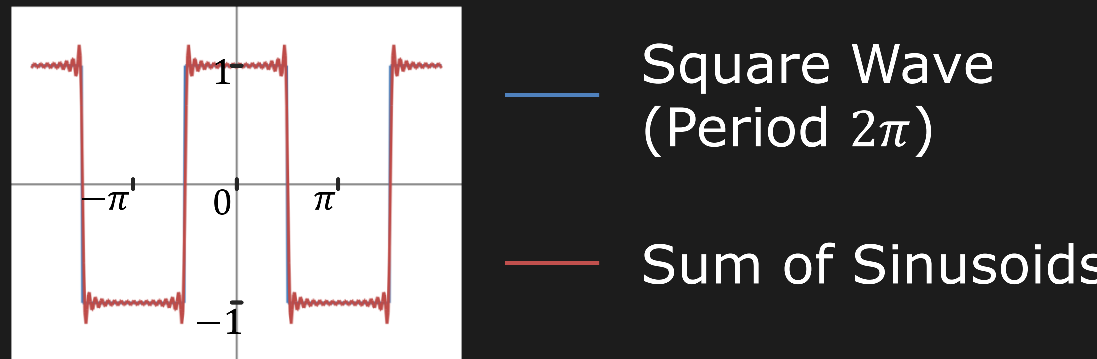
u: Frequency $(1/T)$

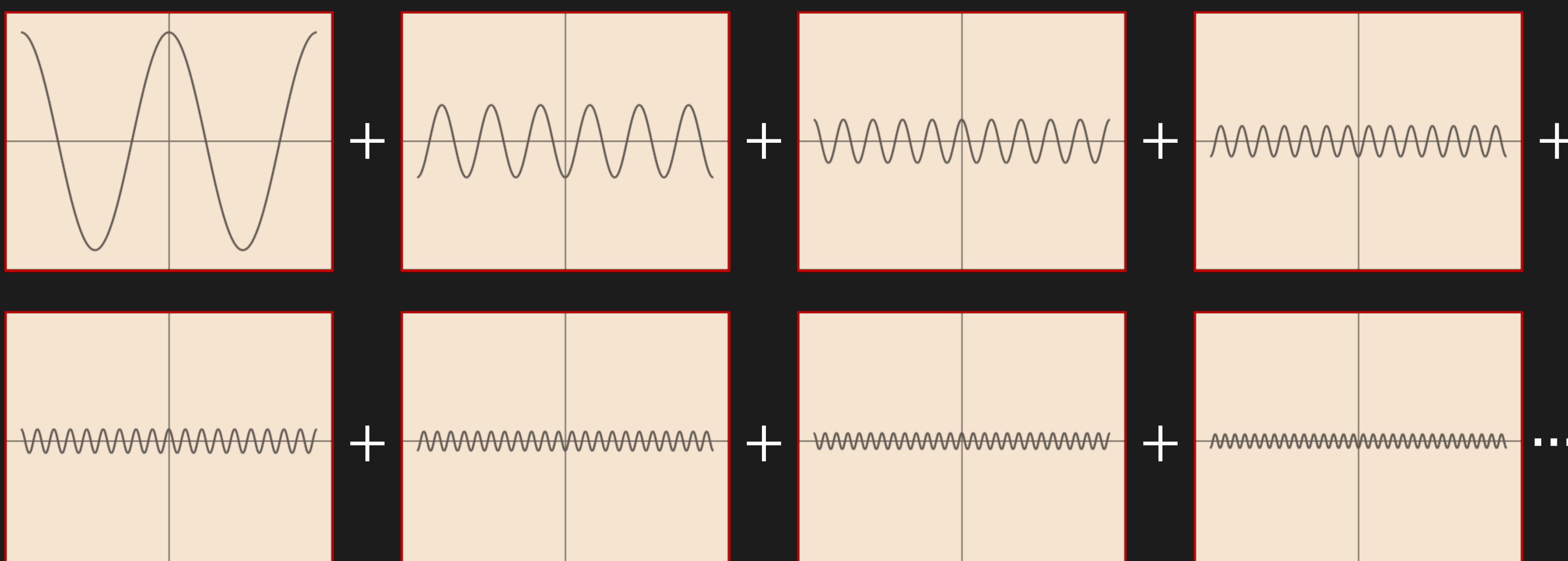
Fourier Series



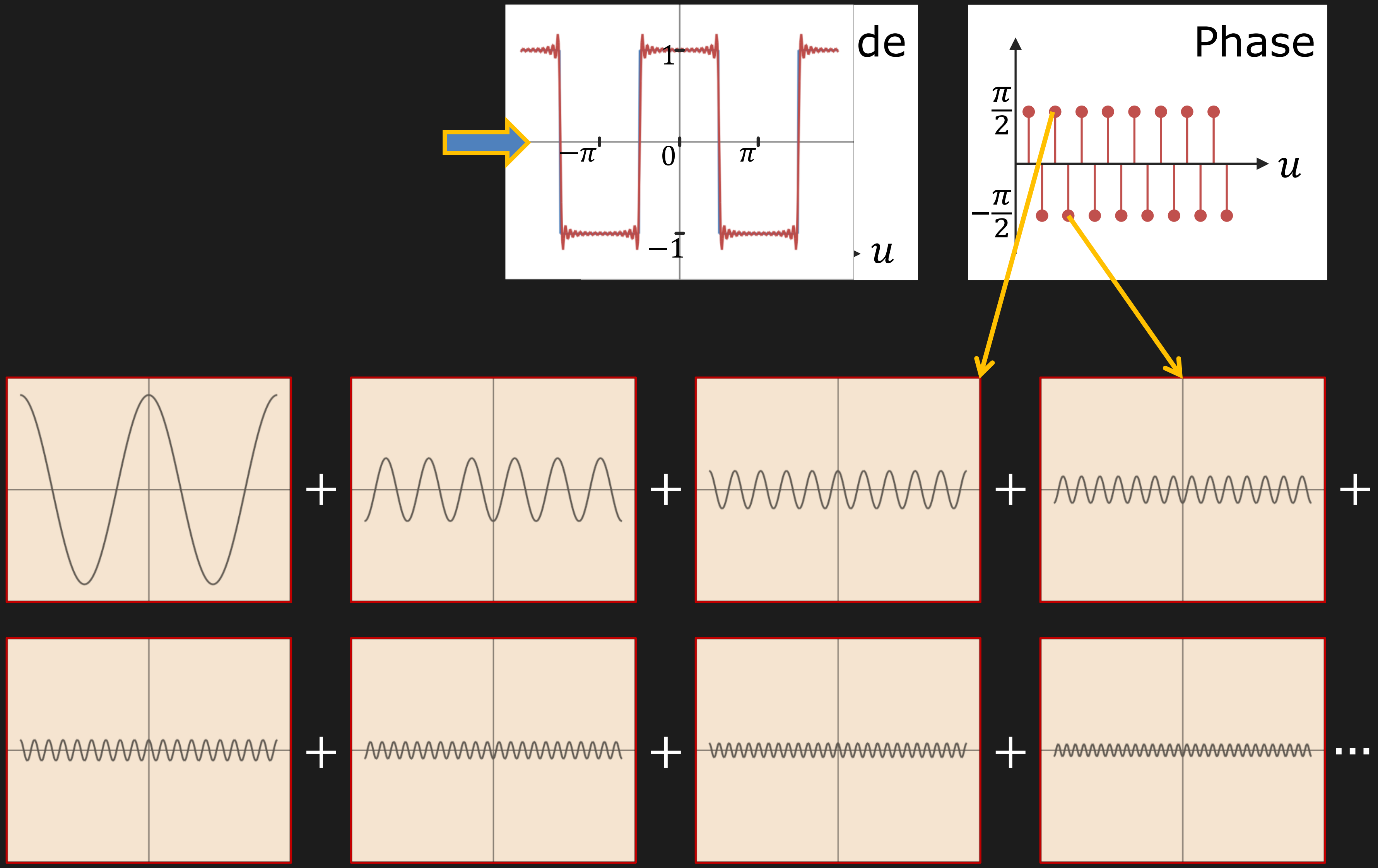


Fourier Series

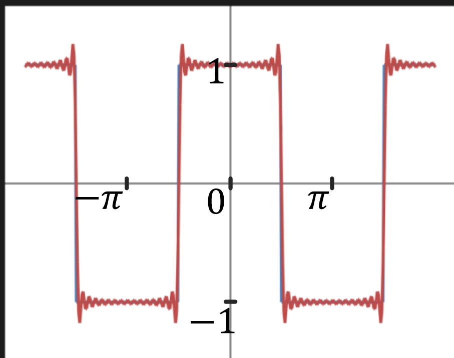


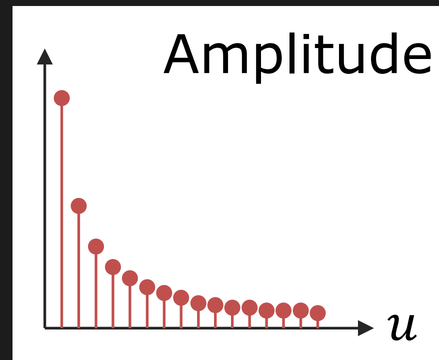


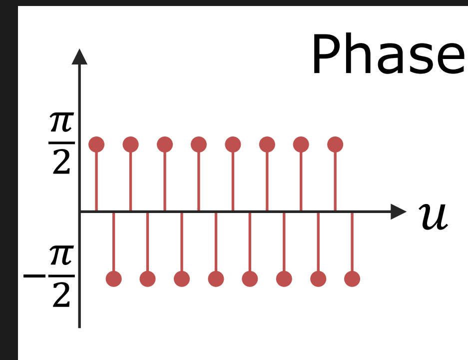
An Alternate Representation of Signal



Fourier Transform (FT)



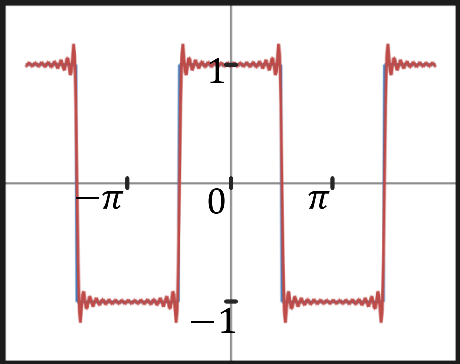


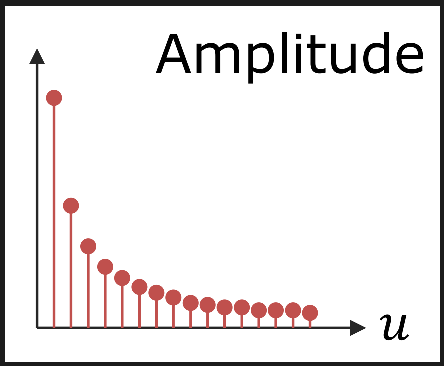


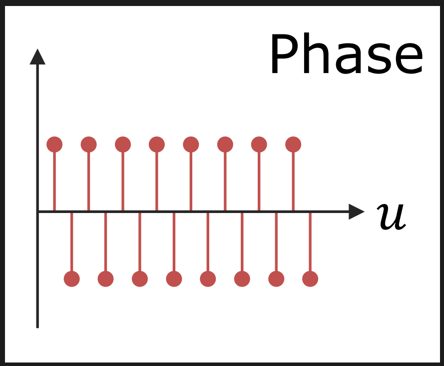
Represents a signal $f(x)$ in terms of Corresponding Amplitudes and Phases of its Constituent Sinusoid.

$$f(x) \longrightarrow \boxed{\mathrm{FT}} \longrightarrow F(u)$$

Inverse Fourier Transform (IFT)







Computes the signal $f(x)$ from the Corresponding Amplitudes and Phases of its Constituent Sinusoid.

$$f(x) \longleftarrow \boxed{\text{IFT}} \longleftarrow F(u)$$

Finding FT and IFT

Fourier Transform:

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

x: space

u: frequency

Inverse Fourier Transform:

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

Exponential Sinusoid (Euler Formula)

$$\boxed{e^{i\theta} = \cos\theta + i\sin\theta} \qquad i = \sqrt{-1}$$

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

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

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

Fourier Transform is Complex!

$F(u)$ holds the Amplitude and Phase of the Exponential Sinusoid of frequency u.

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

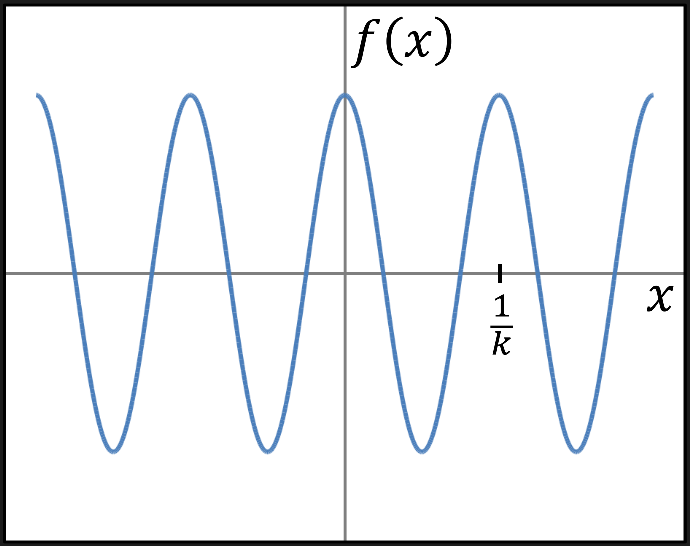
$$F(u) = \mathfrak{Re}\{F(u)\} + i\,\mathfrak{Im}\{F(u)\}$$

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

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

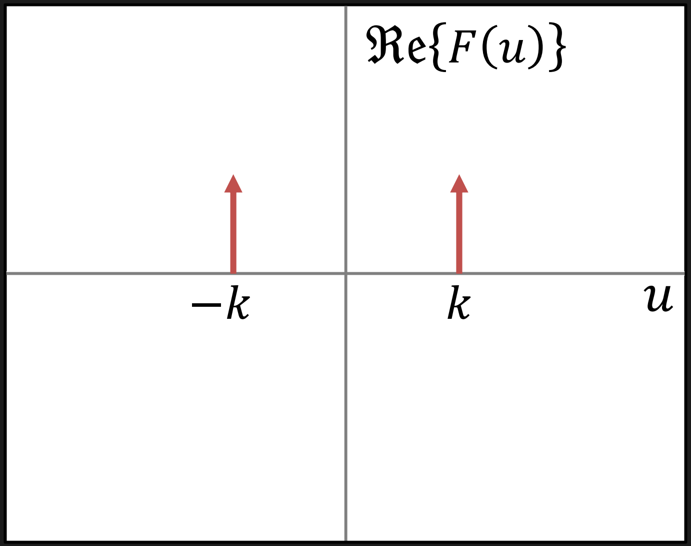
Fourier Transform Examples

Signal $f(x)$



$$f(x) = \cos 2\pi kx$$

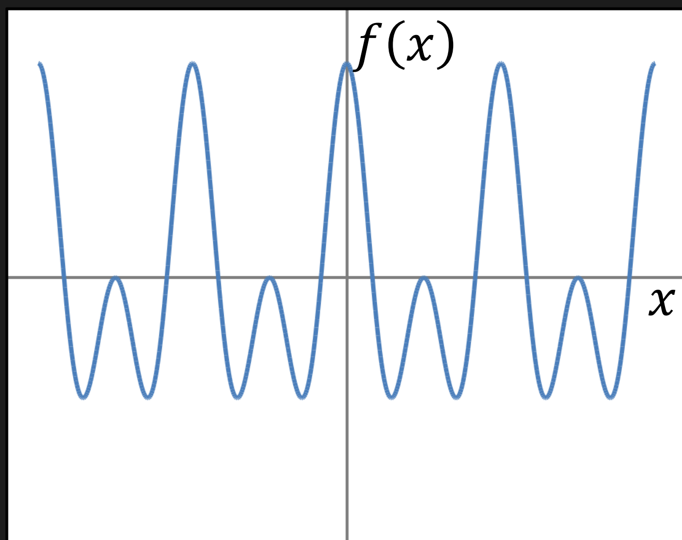
Fourier Transform $F(u)$



$$F(u) = \frac{1}{2}[\delta(u+k) + \delta(u-k)]$$

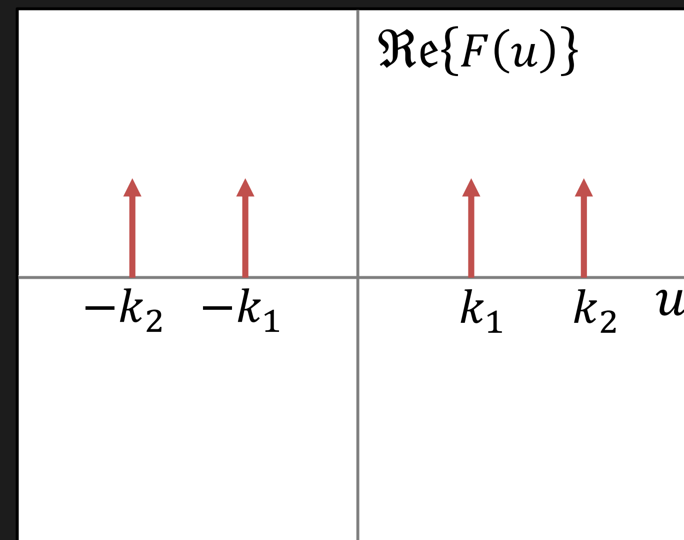
Fourier Transform Examples

Signal $f(x)$



$$f(x) = \cos 2\pi k_1 x \ + \ \cos 2\pi k_2 x$$

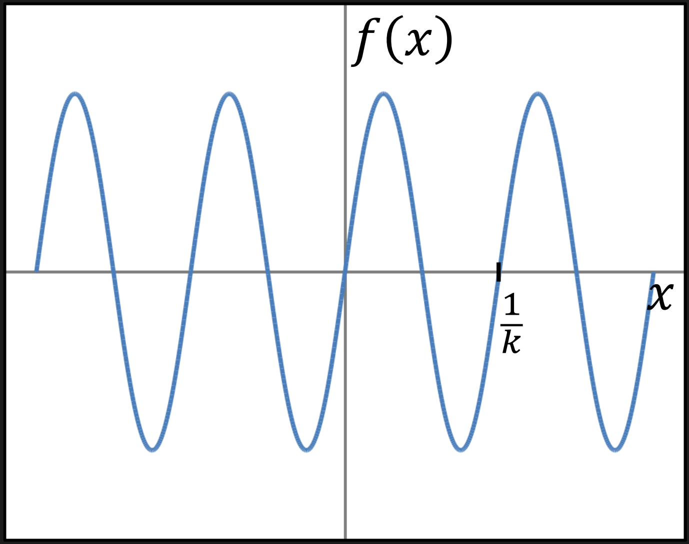
Fourier Transform $F(u)$



$$F(u) = \frac{1}{2}[\delta(u+k_1) \ + \delta(u-k_1)$$

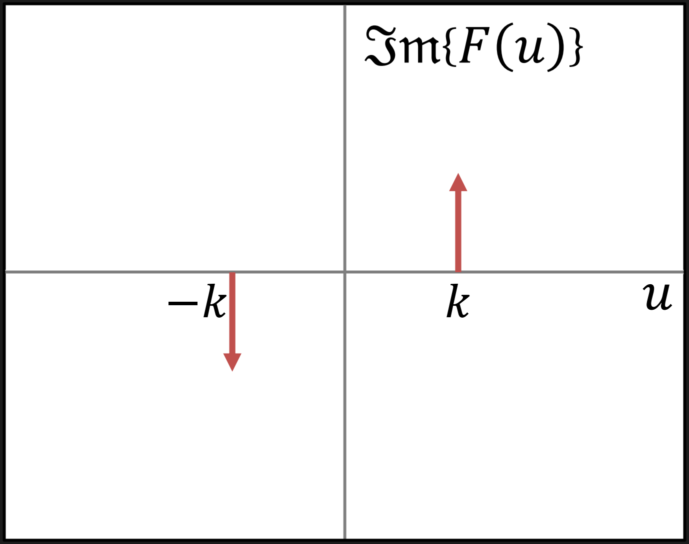
Fourier Transform Examples

Signal $f(x)$



$$f(x) = \sin 2\pi k x$$

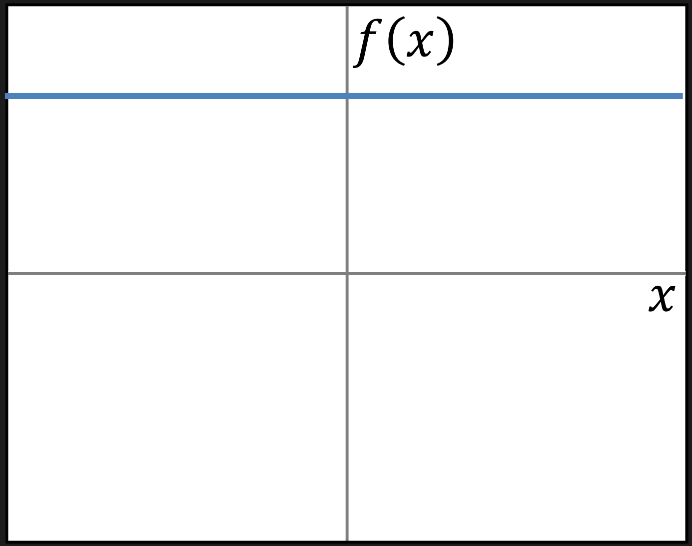
Fourier Transform $F(u)$



$$F(u) = \frac{1}{2} i [\delta(u+k) - \delta(u-k)]$$

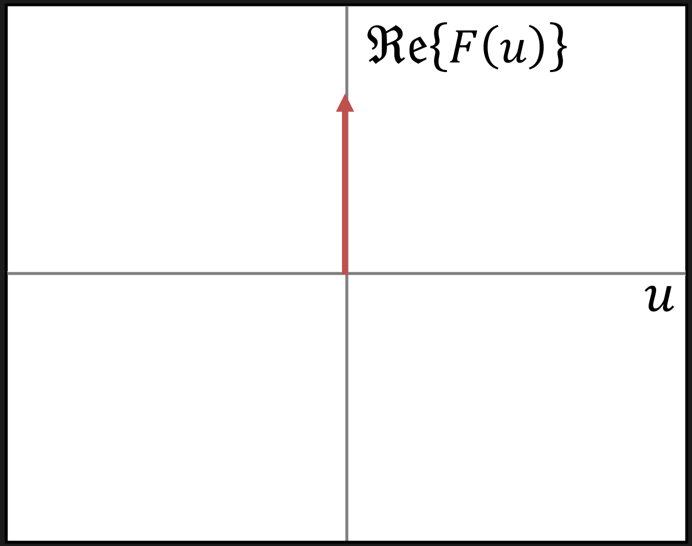
Fourier Transform Examples

Signal $f(x)$



$$f(x) = 1$$

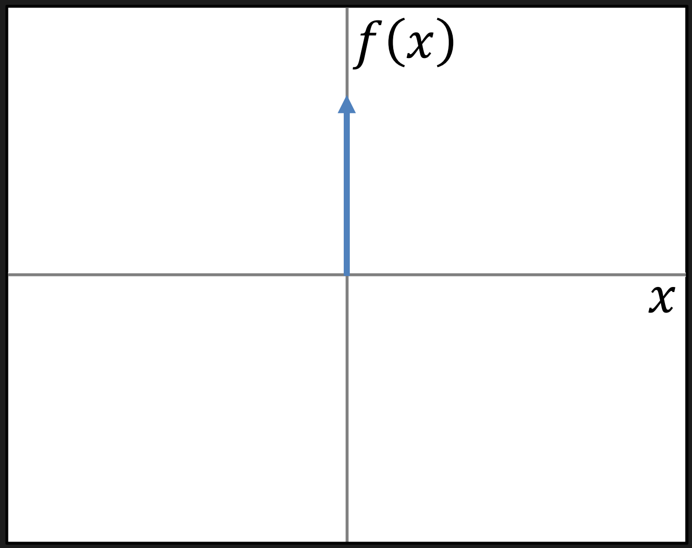
Fourier Transform $F(u)$



$$F(u) = \delta(u)$$

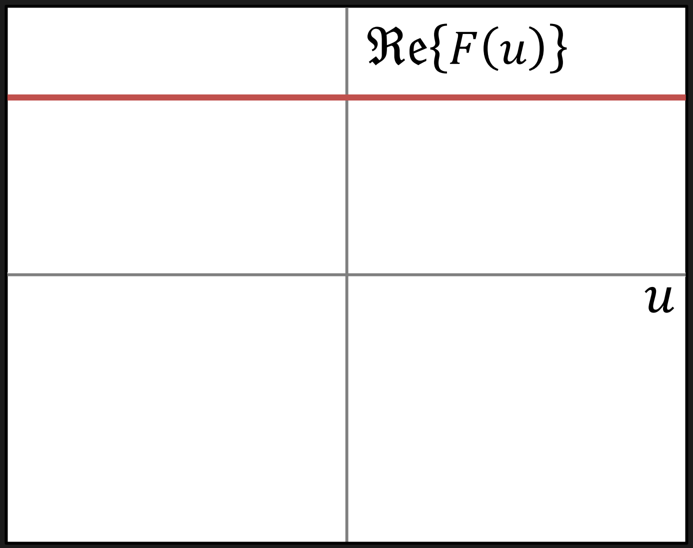
Fourier Transform Examples

Signal $f(x)$



$$f(x) = \delta(x)$$

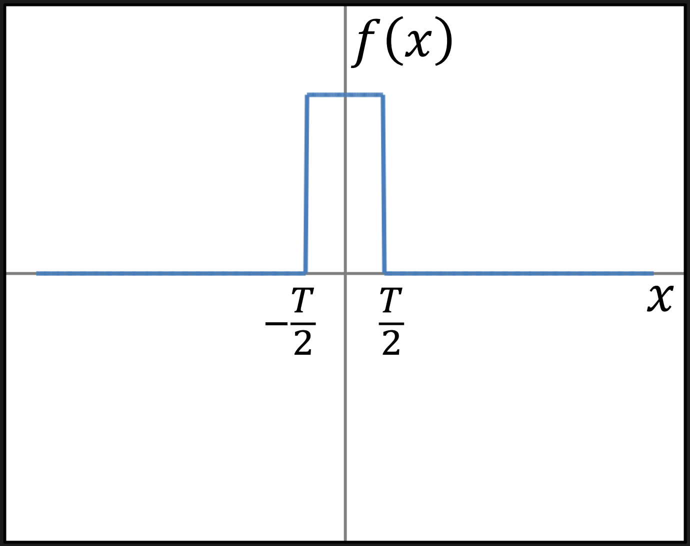
Fourier Transform $F(u)$



$$F(u) = 1$$

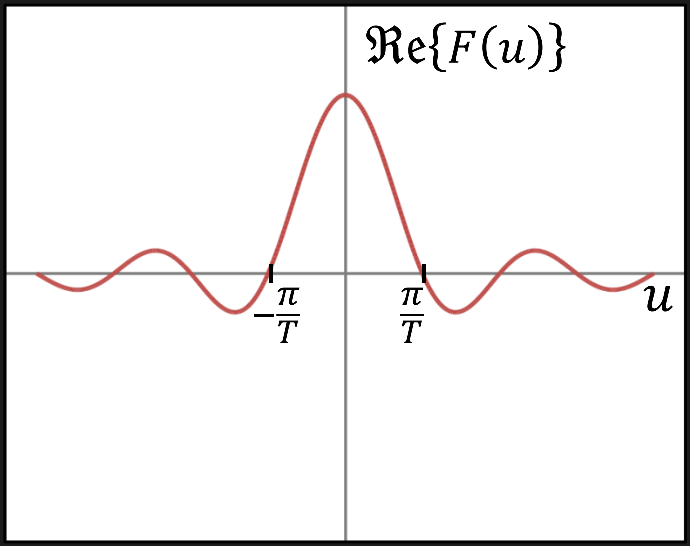
Fourier Transform Examples

Signal $f(x)$



$$f(x) = \mathrm{Rect}\left(\frac{x}{T}\right)$$

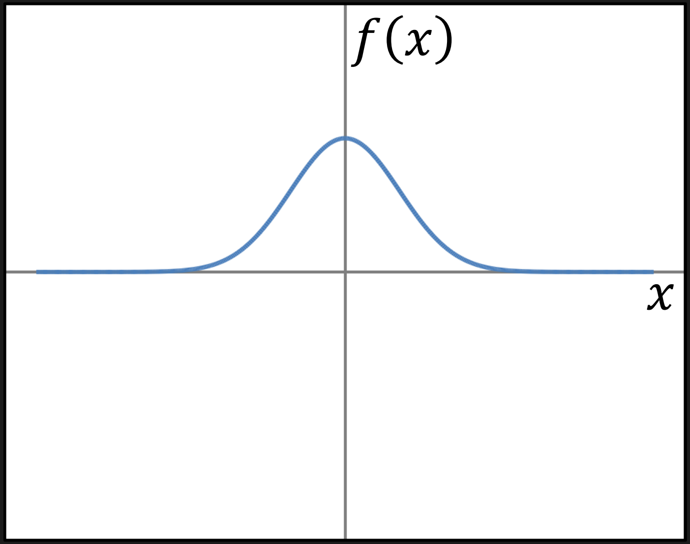
Fourier Transform $F(u)$



$$F(u) = T\,\mathrm{sinc}\,Tu$$

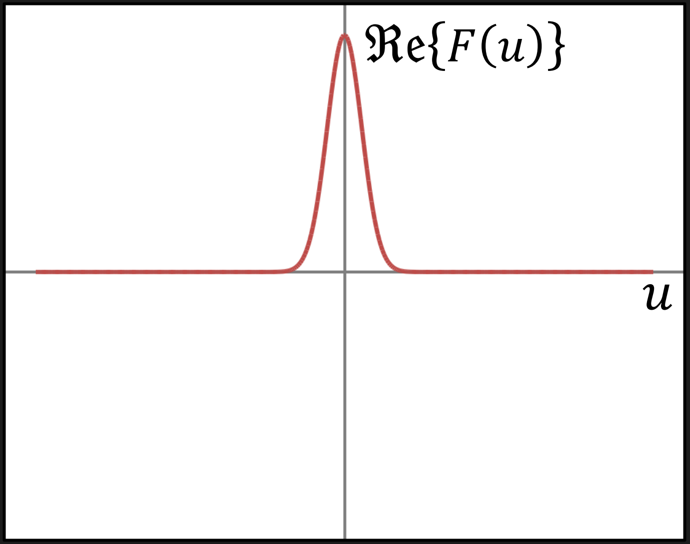
Fourier Transform Examples

Signal $f(x)$



$$f(x) = e^{-ax^2}$$

Fourier Transform $F(u)$

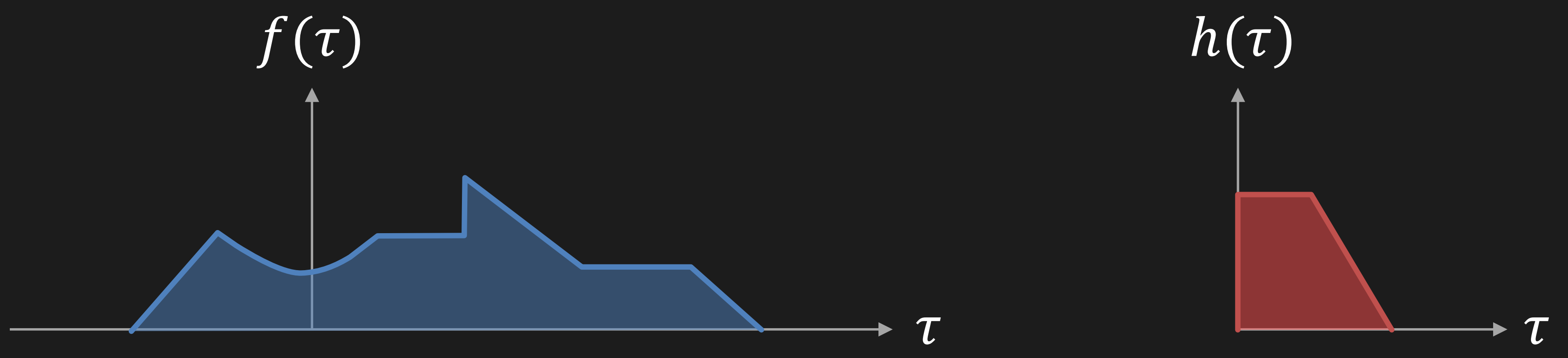


$$F(u) = \sqrt{\pi/a}\; e^{-\pi^2 x^2/a}$$

Convolution (Review)

Convolution of two functions $f(x)$ and $h(x)$

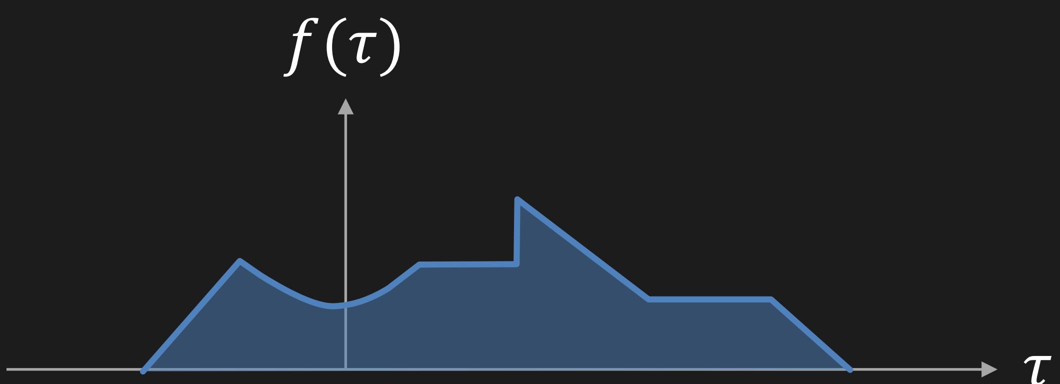
$$\boxed{g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\,d\tau}$$

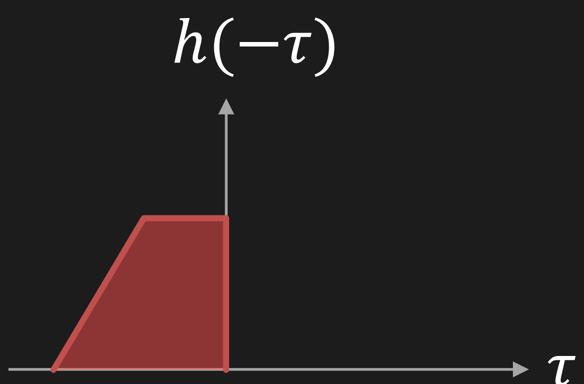


Convolution (Review)

Convolution of two functions $f(x)$ and $h(x)$

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

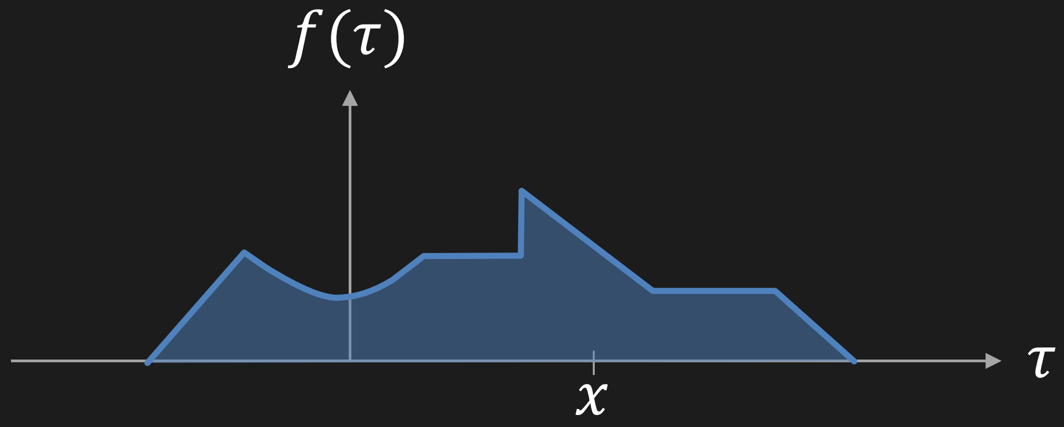


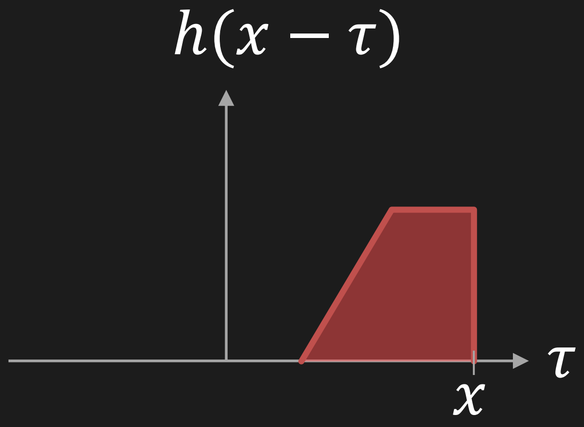


Convolution (Review)

Convolution of two functions $f(x)$ and $h(x)$

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



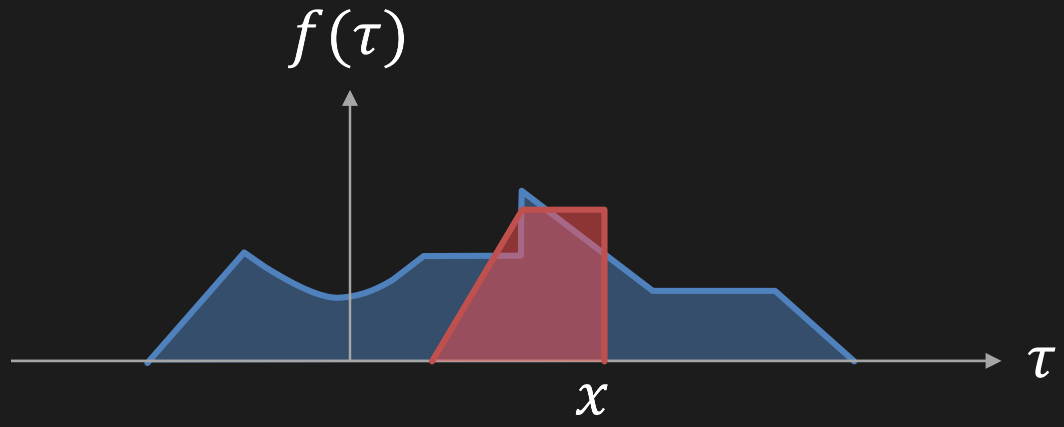


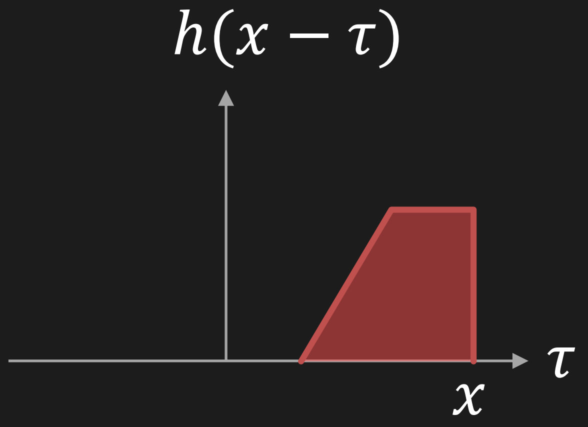
REVIEW

Convolution (Review)

Convolution of two functions $f(x)$ and $h(x)$

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

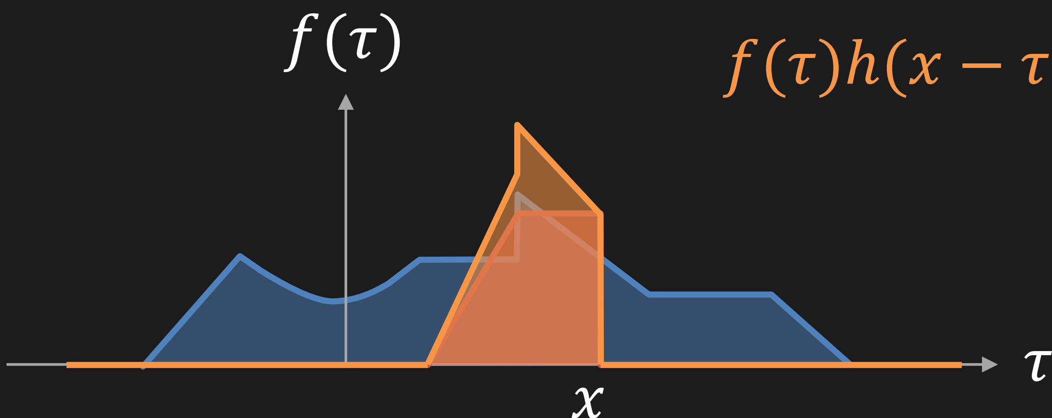


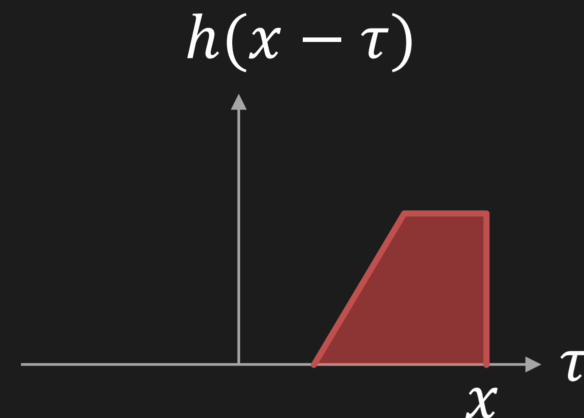


Convolution (Review)

Convolution of two functions $f(x)$ and $h(x)$

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

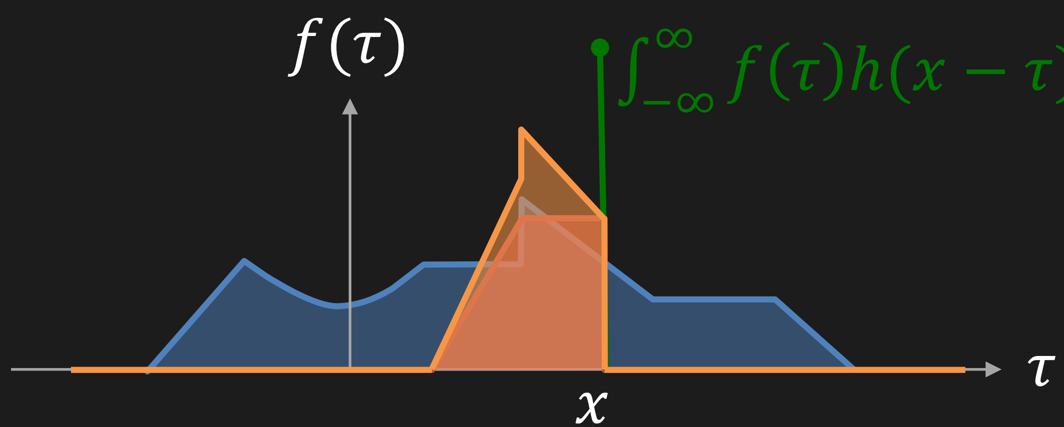


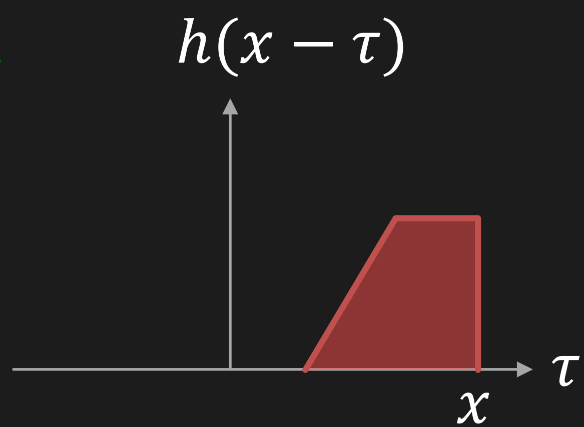


Convolution (Review)

Convolution of two functions $f(x)$ and $h(x)$

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

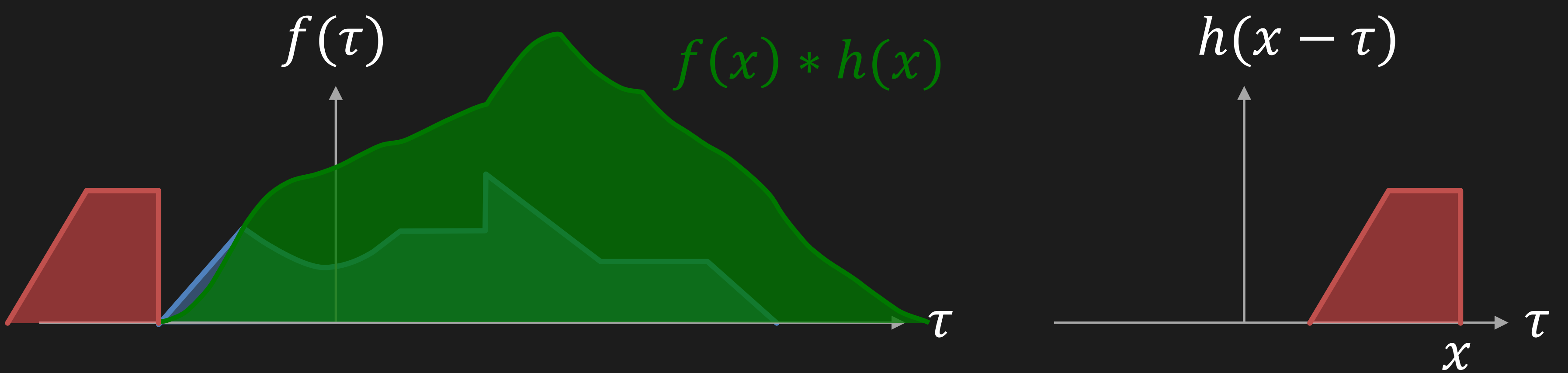




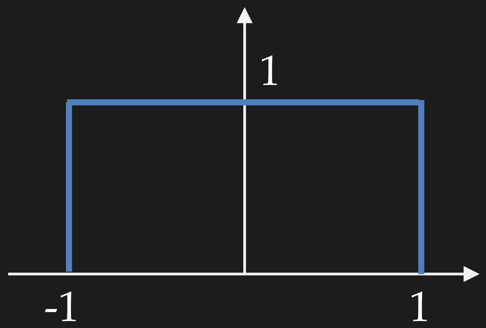
Convolution (Review)

Convolution of two functions $f(x)$ and $h(x)$

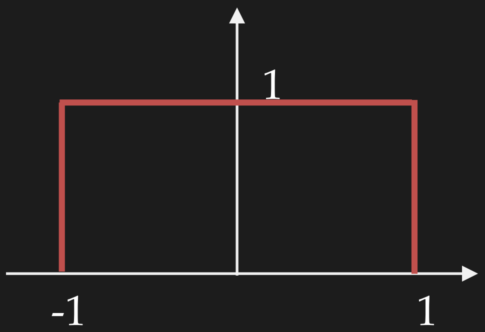
$$g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\,d\tau$$



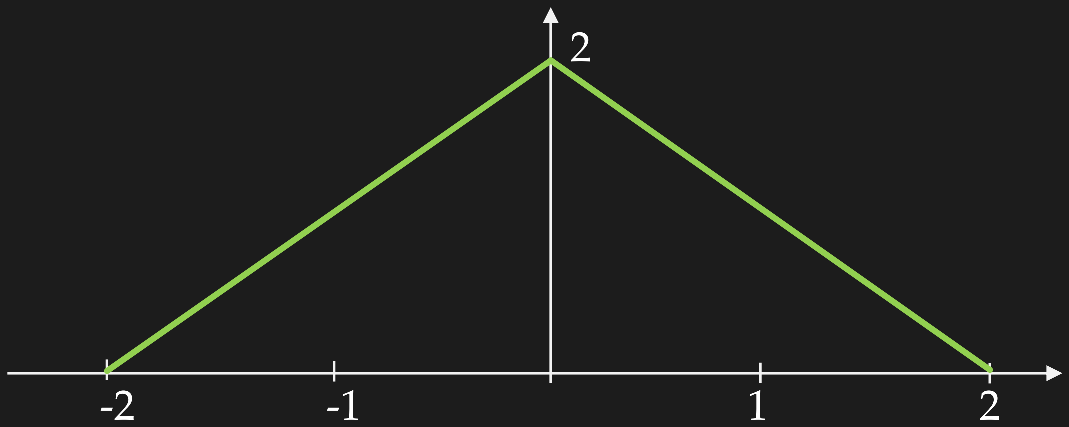
Convolution: Example



$f(x)$



$h(x)$



$f(x) * h(x)$

REVIEW

Convolution and Fourier Transform

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

Then FT of $g(x)$:

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

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

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

Convolution and Fourier Transform

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

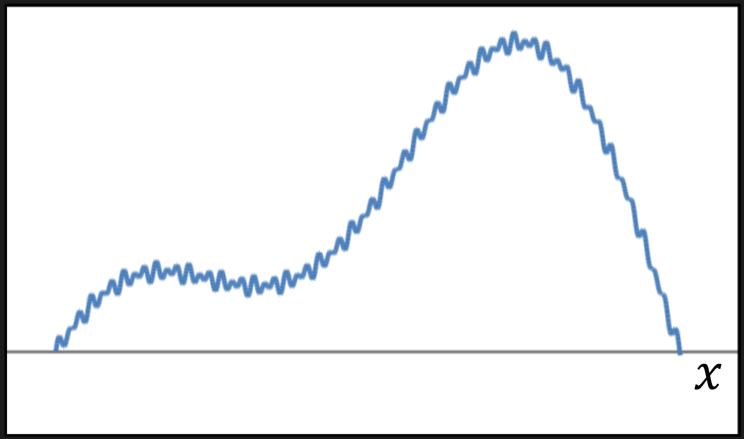
Properties of Fourier Transform

Property	Spatial Domain	Frequency Domain
Linearity	$\alpha f_1(x) + \beta f_2(x)$	$\alpha F_1(u) + \beta F_2(u)$
Scaling	$f(ax)$	$\frac{1}{\lvert a \rvert} F\left(\frac{u}{a}\right)$
Shifting	$f(x-a)$	$e^{-i2\pi ua} F(u)$
Differentiation	$\frac{d^n}{dx^n}\left(f(x)\right)$	$(i2\pi u)^n F(u)$

Convolution Using Fourier Transform

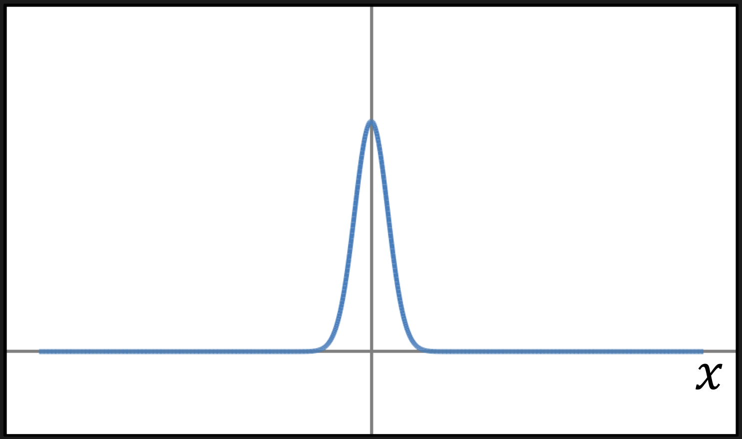
$$
\begin{tikzcd}
g(x) & = & f(x) \arrow[d, "\mathrm{FT}"] & * & h(x) \arrow[d, "\mathrm{FT}"] \\
G(u) \arrow[u, "\mathrm{IFT}"] & = & F(u) & \times & H(u)
\end{tikzcd}
$$

Gaussian Smoothing in Fourier Domain



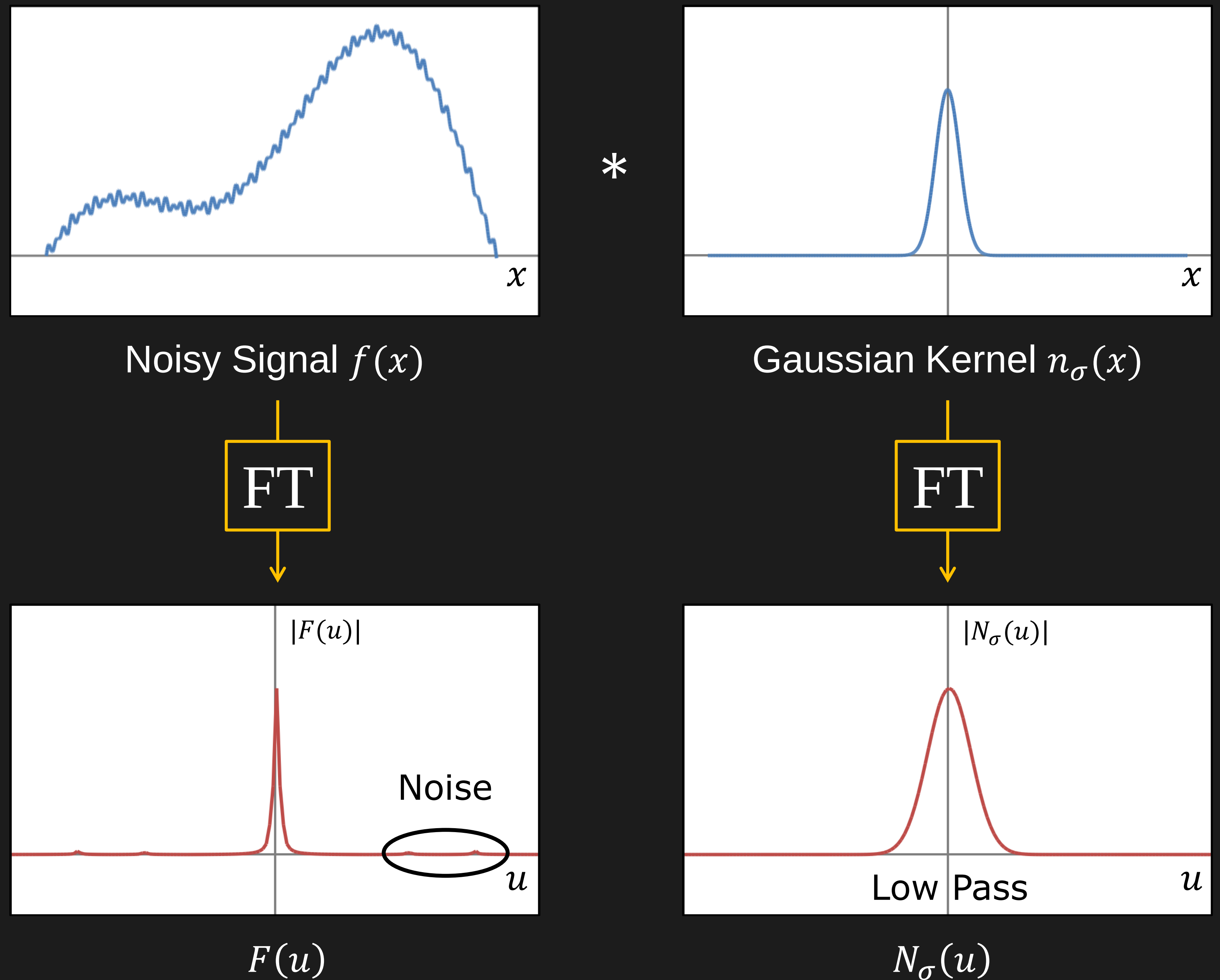
Noisy Signal $f(x)$

$*$

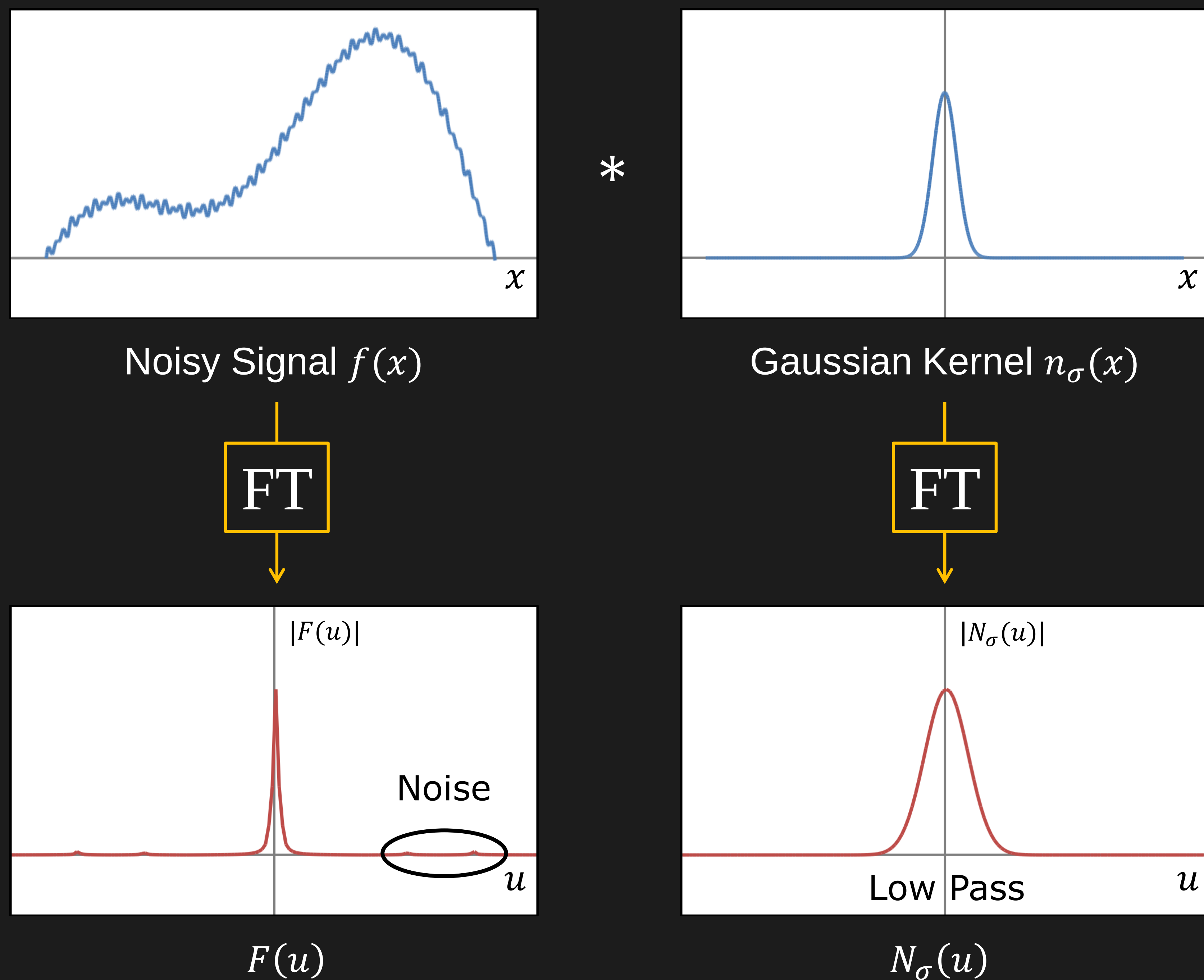


Gaussian Kernel $n_\sigma(x)$

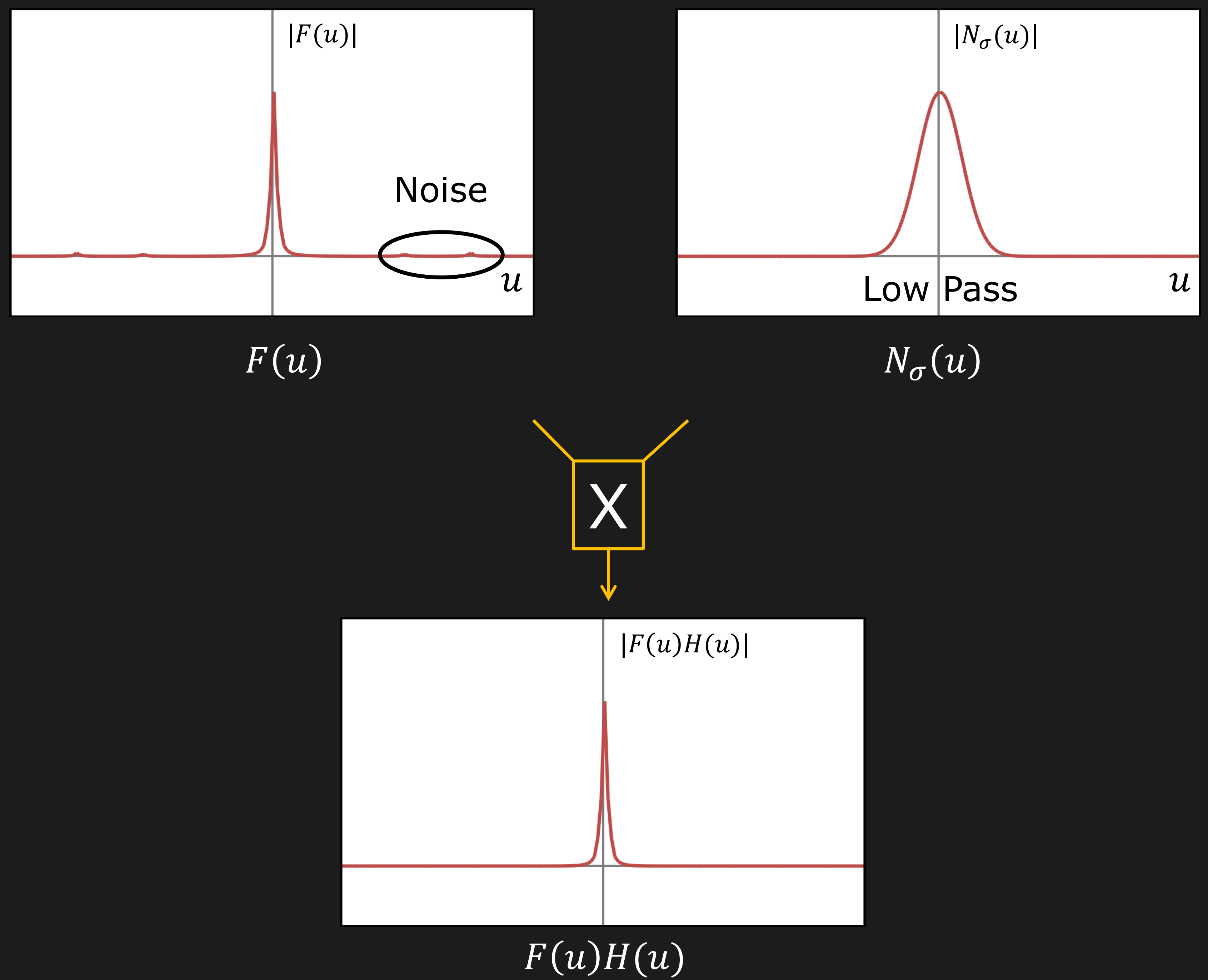
Convolve the Noisy Signal with a Gaussian Kernel



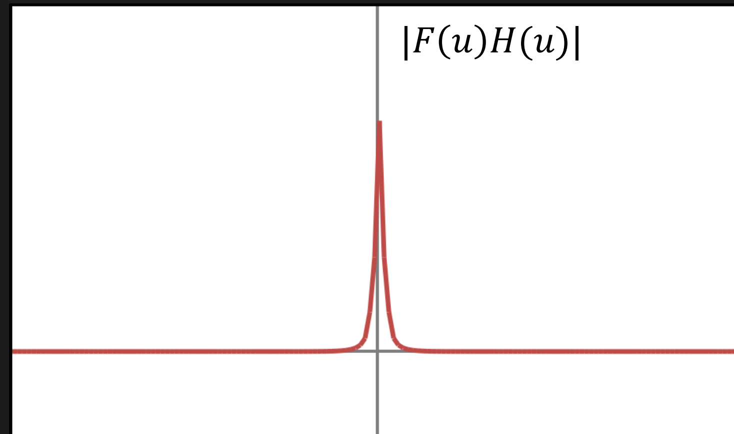
x
*
x
Noisy Signal $f(x)$
Gaussian Kernel $n_\sigma(x)$
FT
FT
$|F(u)|$
Noise
u
$|N_\sigma(u)|$
Low
Pass
u
$F(u)$
$N_\sigma(u)$



x
Noisy Signal $f(x)$
*
x
Gaussian Kernel $n_\sigma(x)$
FT
FT
$|F(u)|$
Noise
u
$F(u)$
$|N_\sigma(u)|$
Low
Pass
u
$N_\sigma(u)$

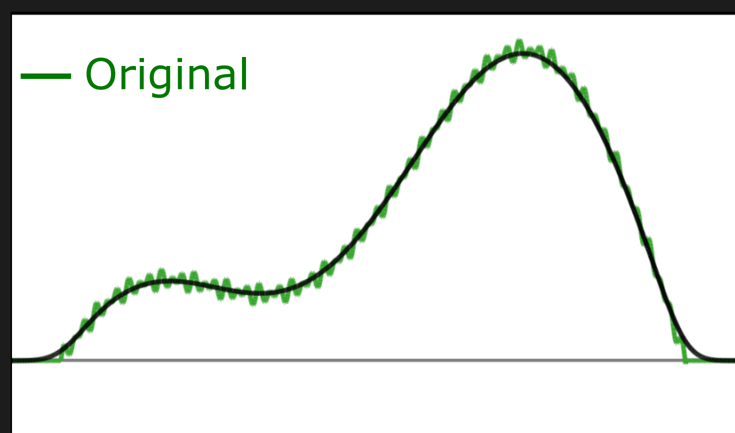


$|F(u)|$
Noise
u
$F(u)$
$|N_\sigma(u)|$
Low Pass
u
$N_\sigma(u)$
X
$|F(u)H(u)|$
$F(u)H(u)$



$F(u)H(u)$

IFT



Gaussian Blurred Signal $g(x)$

2D Fourier Transform

Fourier Transform:

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

u and v are frequencies along x and y, respectively

Inverse Fourier Transform:

$$f(x,y) = \iint_{-\infty}^{\infty} F(u,v) e^{i2\pi(xu+yv)} dudv$$

2D Fourier Transform: Discrete Images

Discrete Fourier Transform (DFT):

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

$p = 0 \ldots M-1$

$q = 0 \ldots N-1$

p and q are frequencies along m and n, respectively

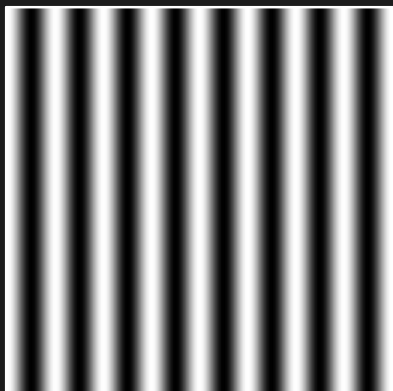
Inverse Discrete Fourier Transform (IDFT):

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

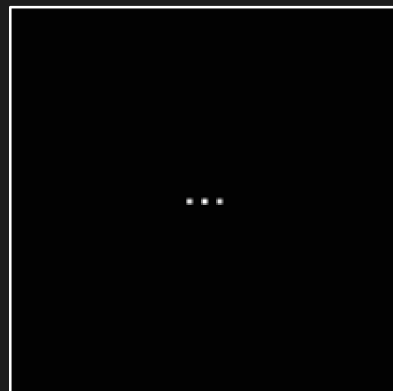
$m = 0 \ldots M-1$

$n = 0 \ldots N-1$

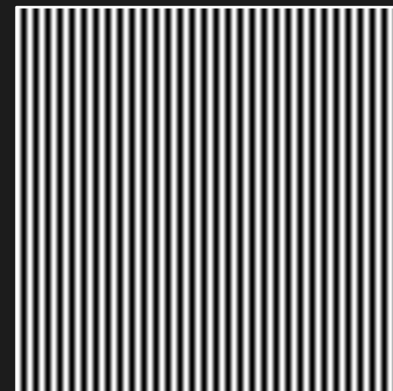
2D Fourier Transform: Example 1



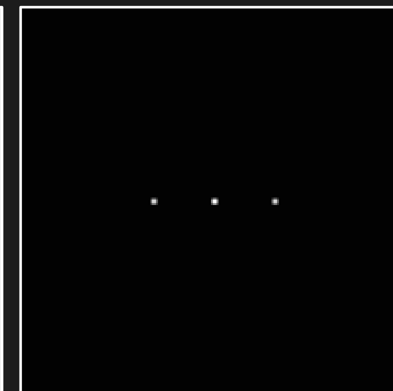
$f(m,n)$



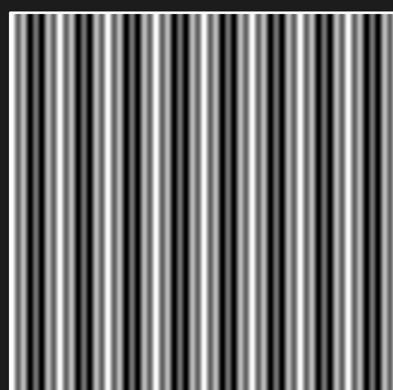
$\log(|F(p,q)|)$



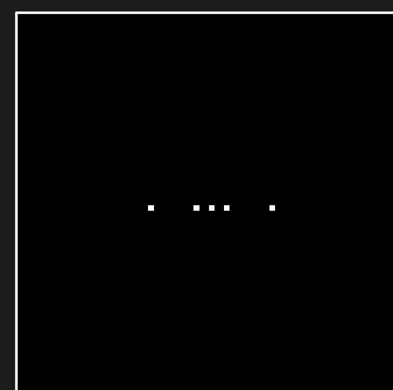
$g(m,n)$



$\log(|G(p,q)|)$



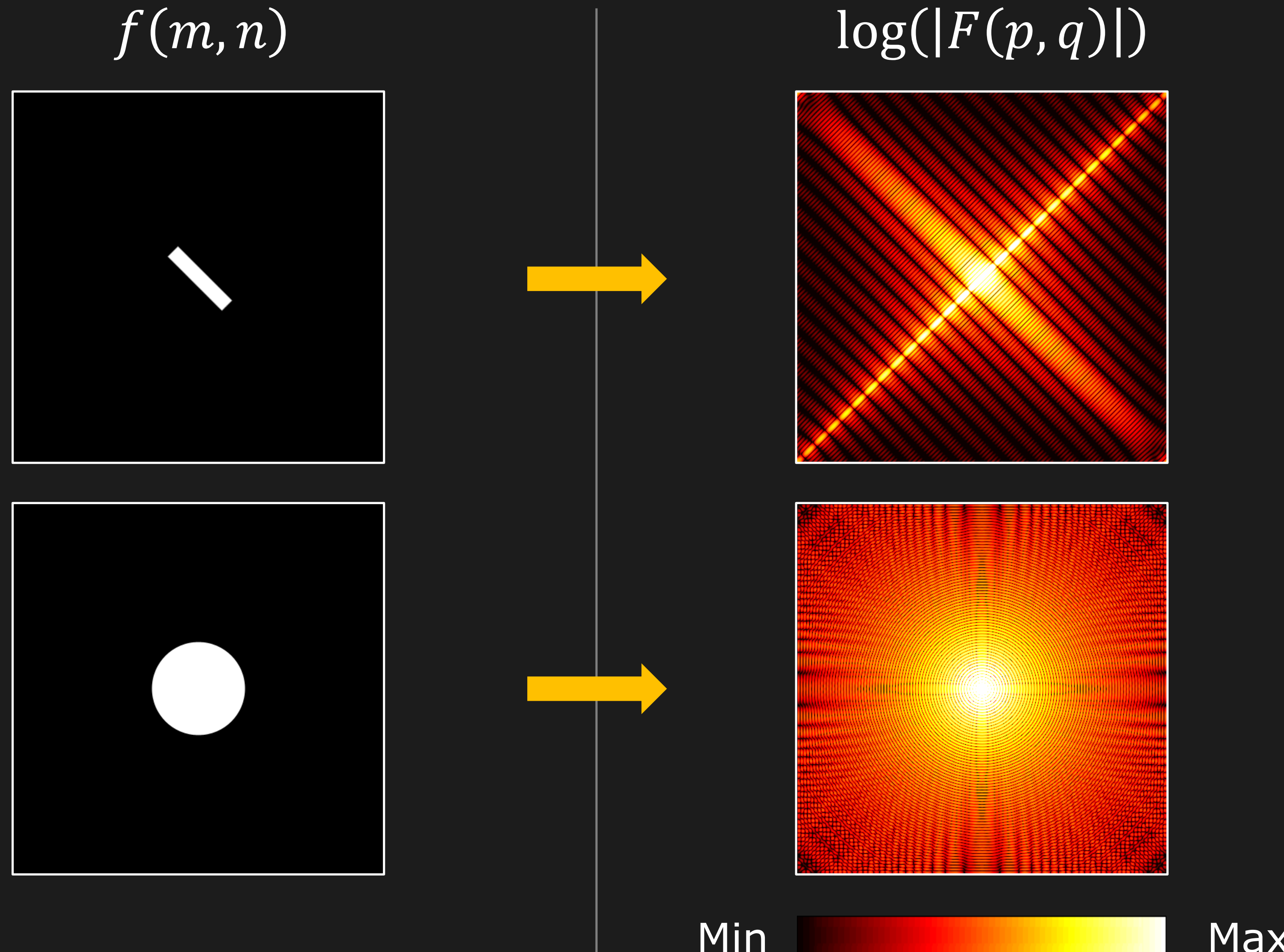
$f(m,n) + g(m,n)$



$\log(|F(p,q) + G(p,q)|)$

Note: $\log(|F|)$ is used just for display

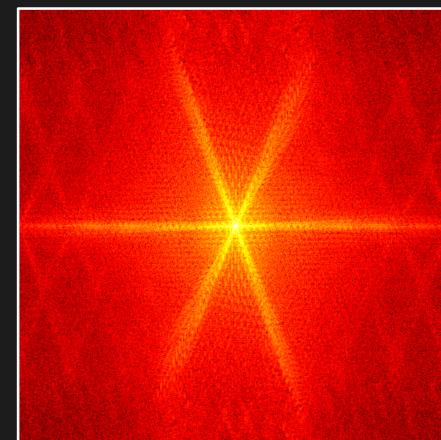
2D Fourier Transform: Example 2

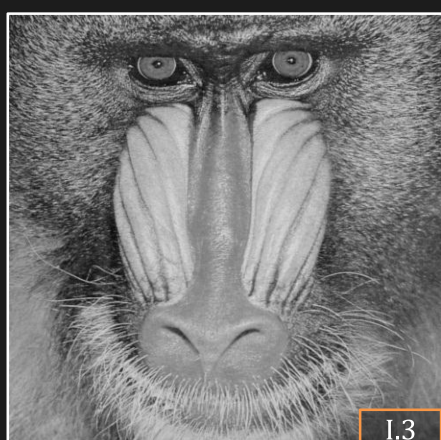


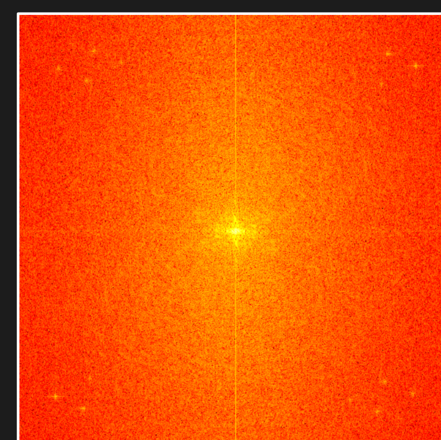
2D Fourier Transform: Example 3

$f(m,n)$

$\log(|F(p,q)|)$





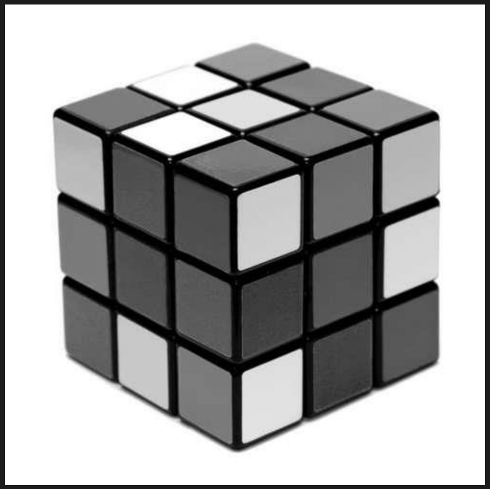


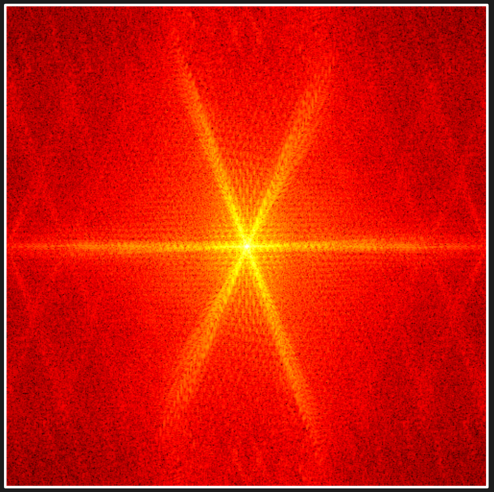
Min  Max

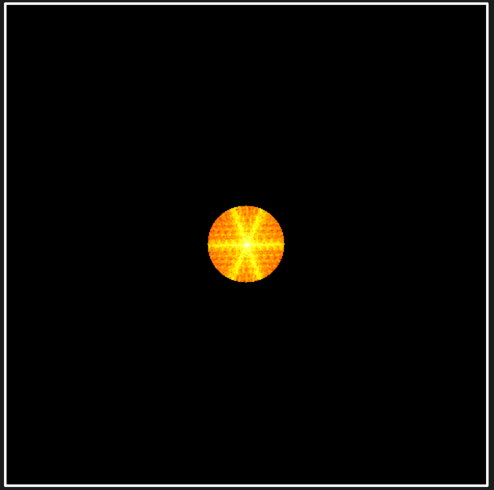
Low Pass Filtering

$f(m, n)$

$\log(|F(p, q)|)$

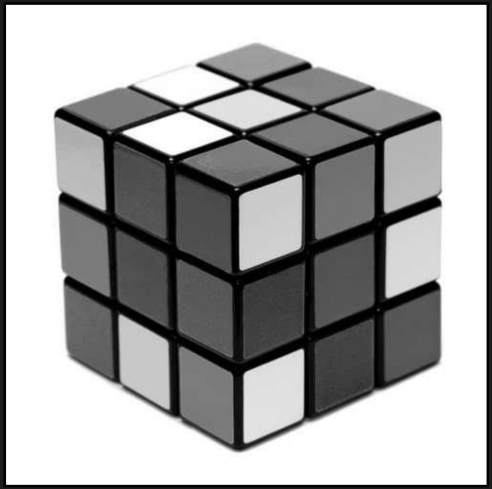




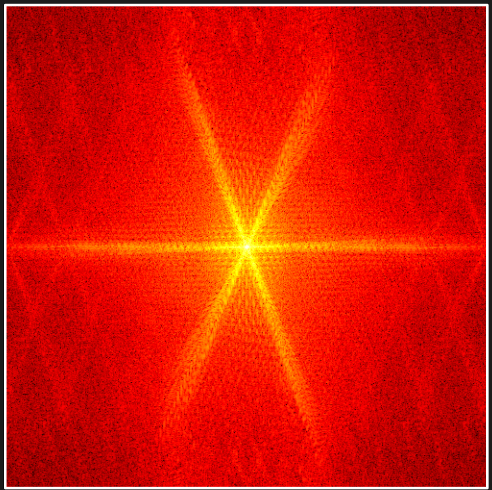


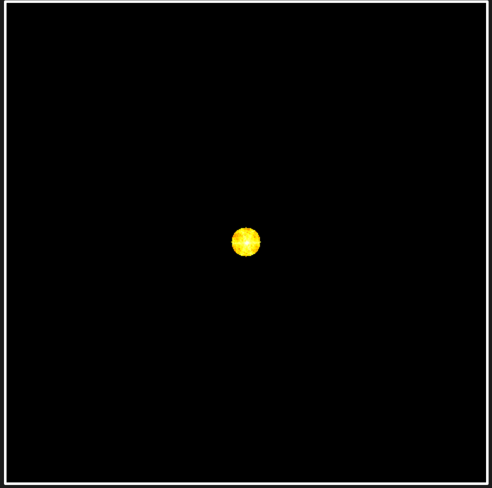
Low Pass Filtering

$f(m,n)$



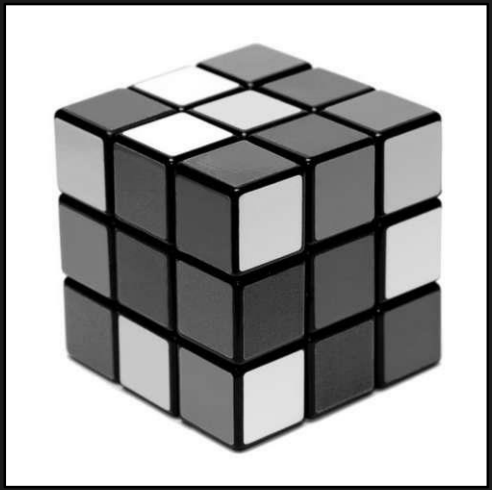
$\log(|F(p,q)|)$



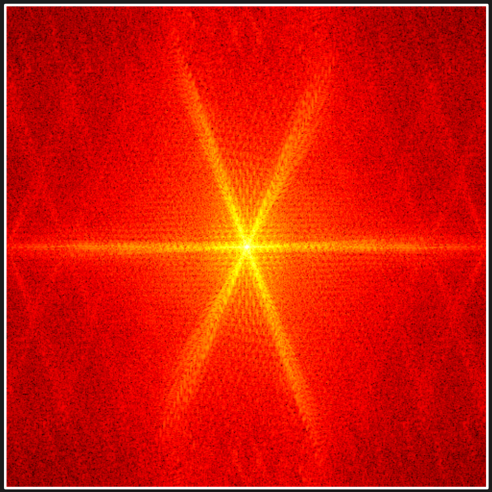


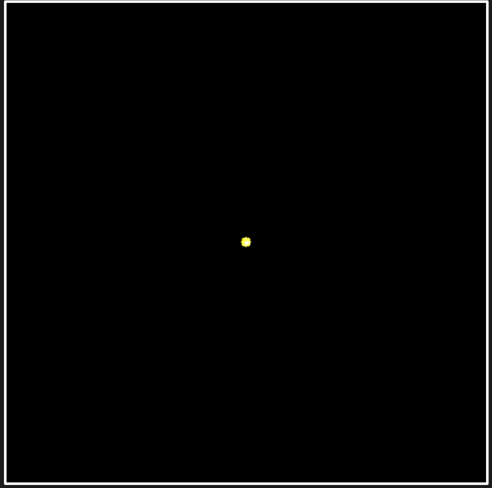
Low Pass Filtering

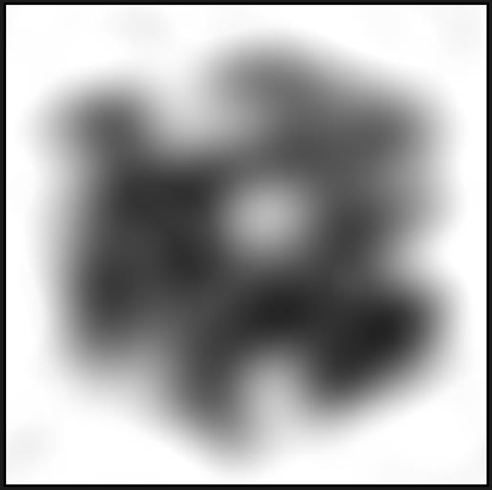
$f(m,n)$



$\log(|F(p,q)|)$

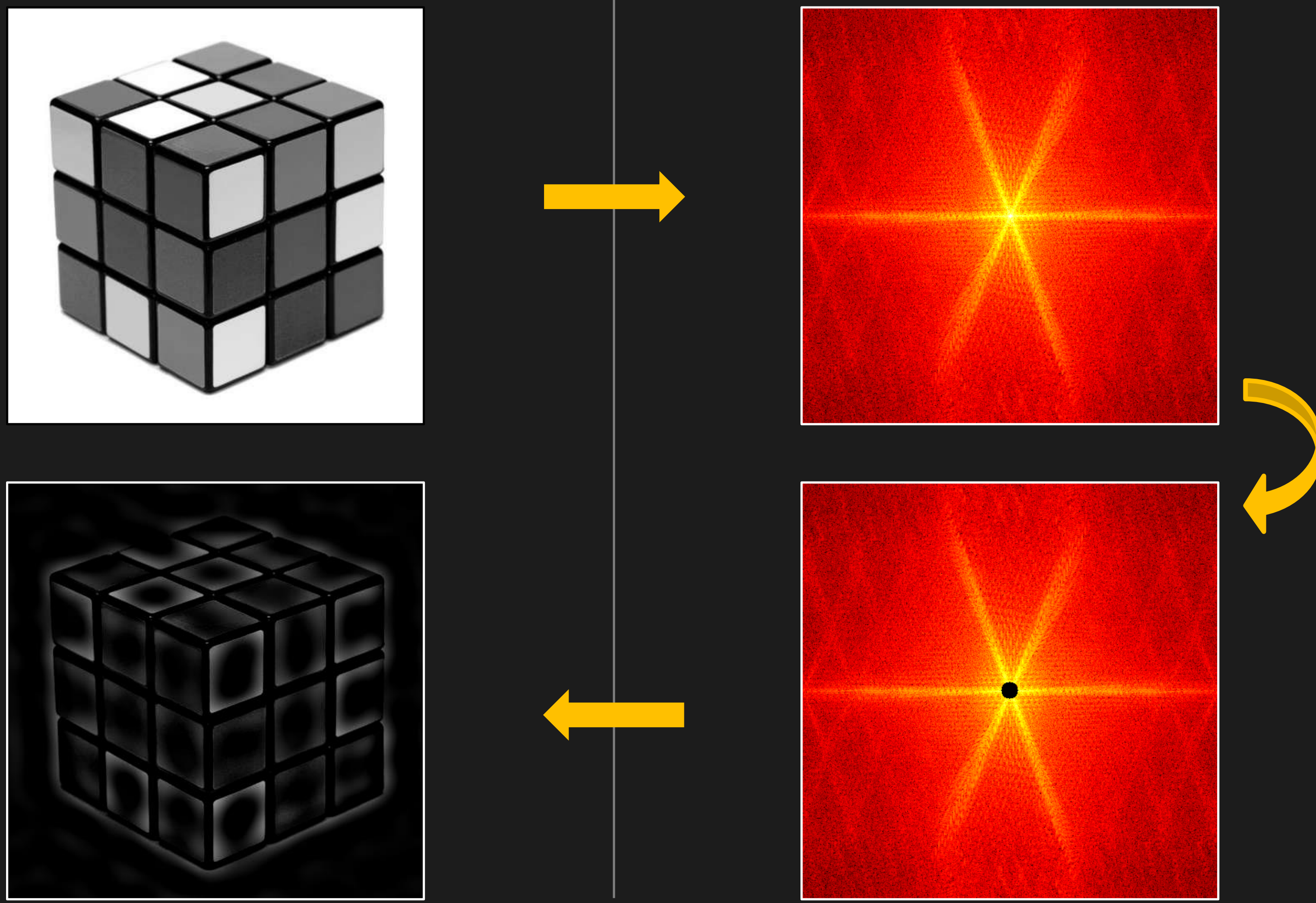






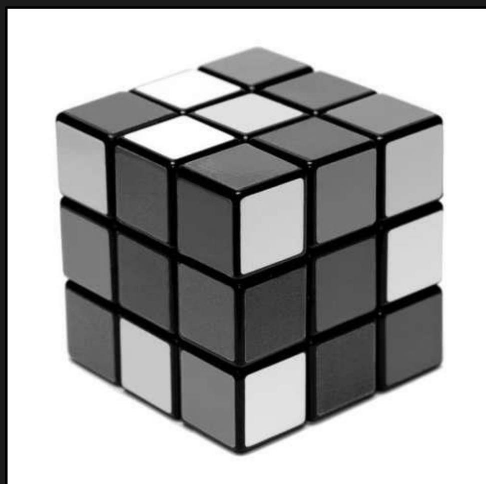
High Pass Filtering

$f(m,n)$ $\log(|F(p,q)|)$

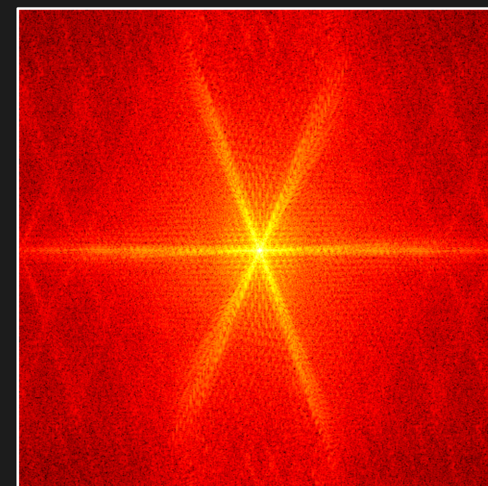


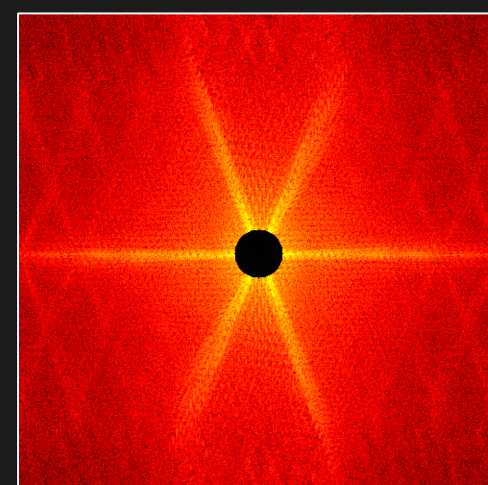
High Pass Filtering

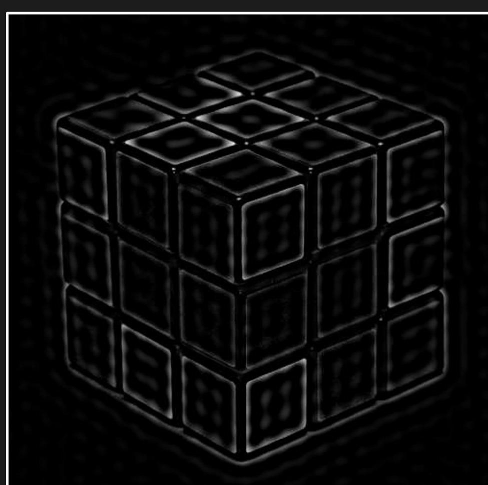
$f(m,n)$



$\log(|F(p,q)|)$

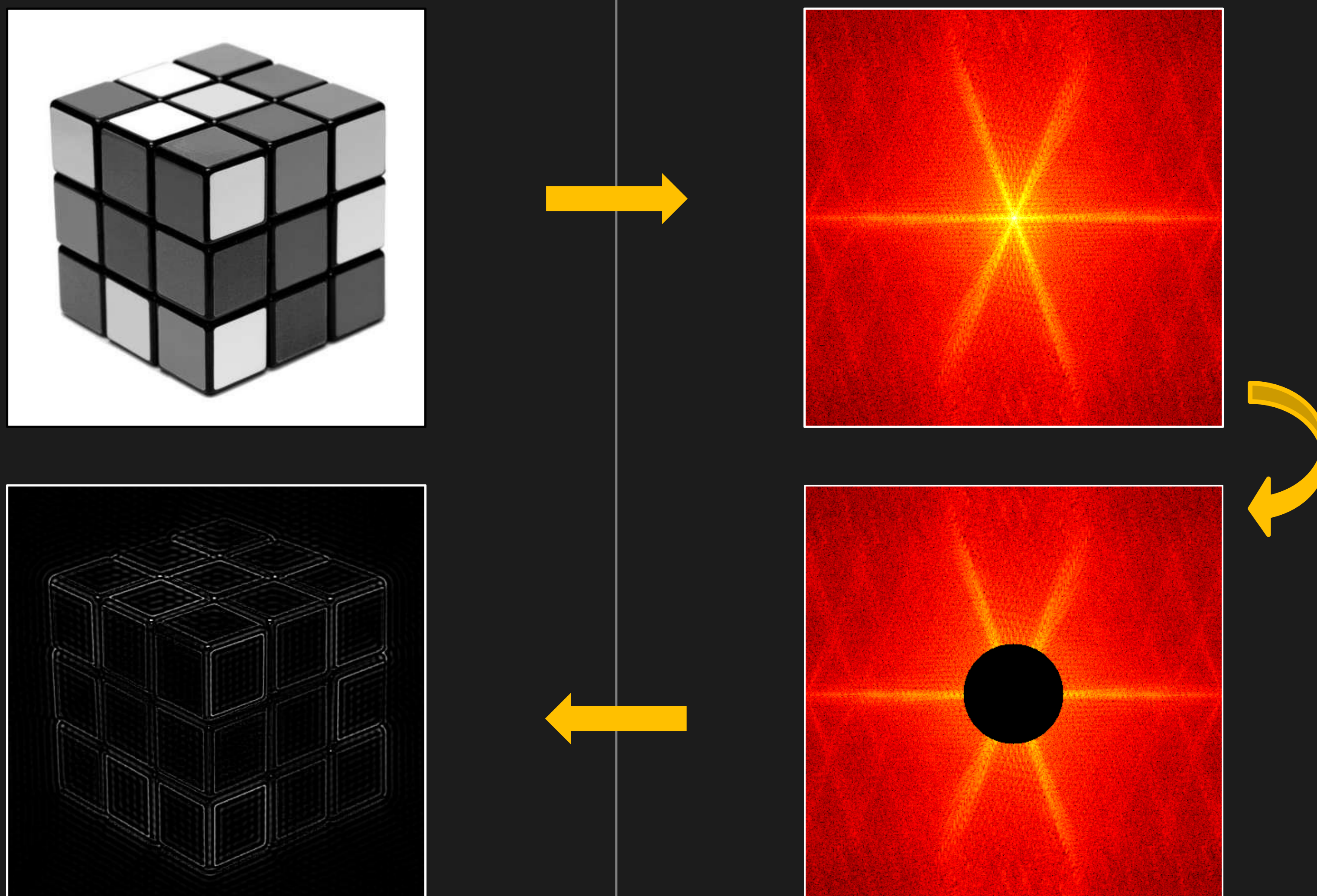




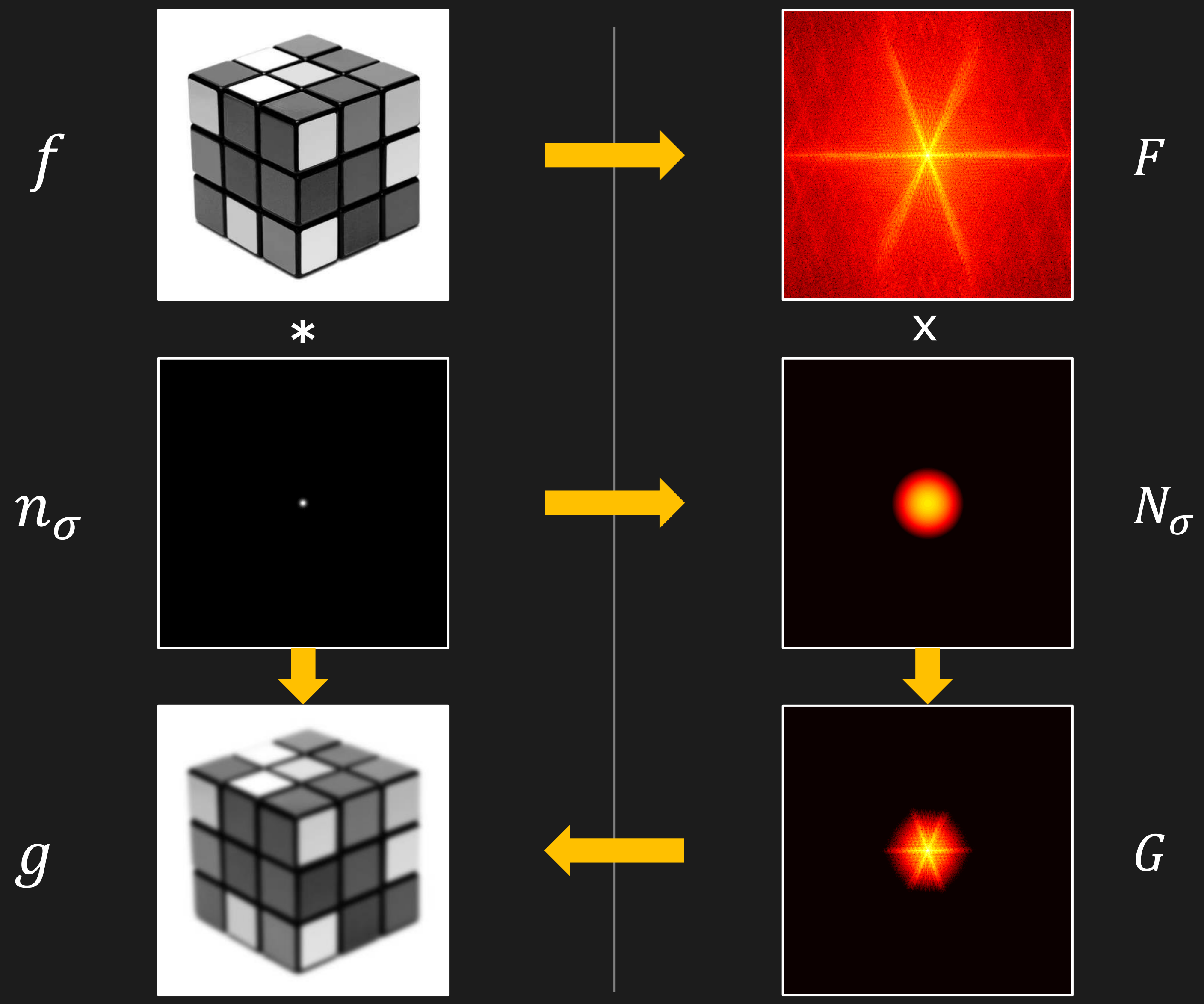


High Pass Filtering

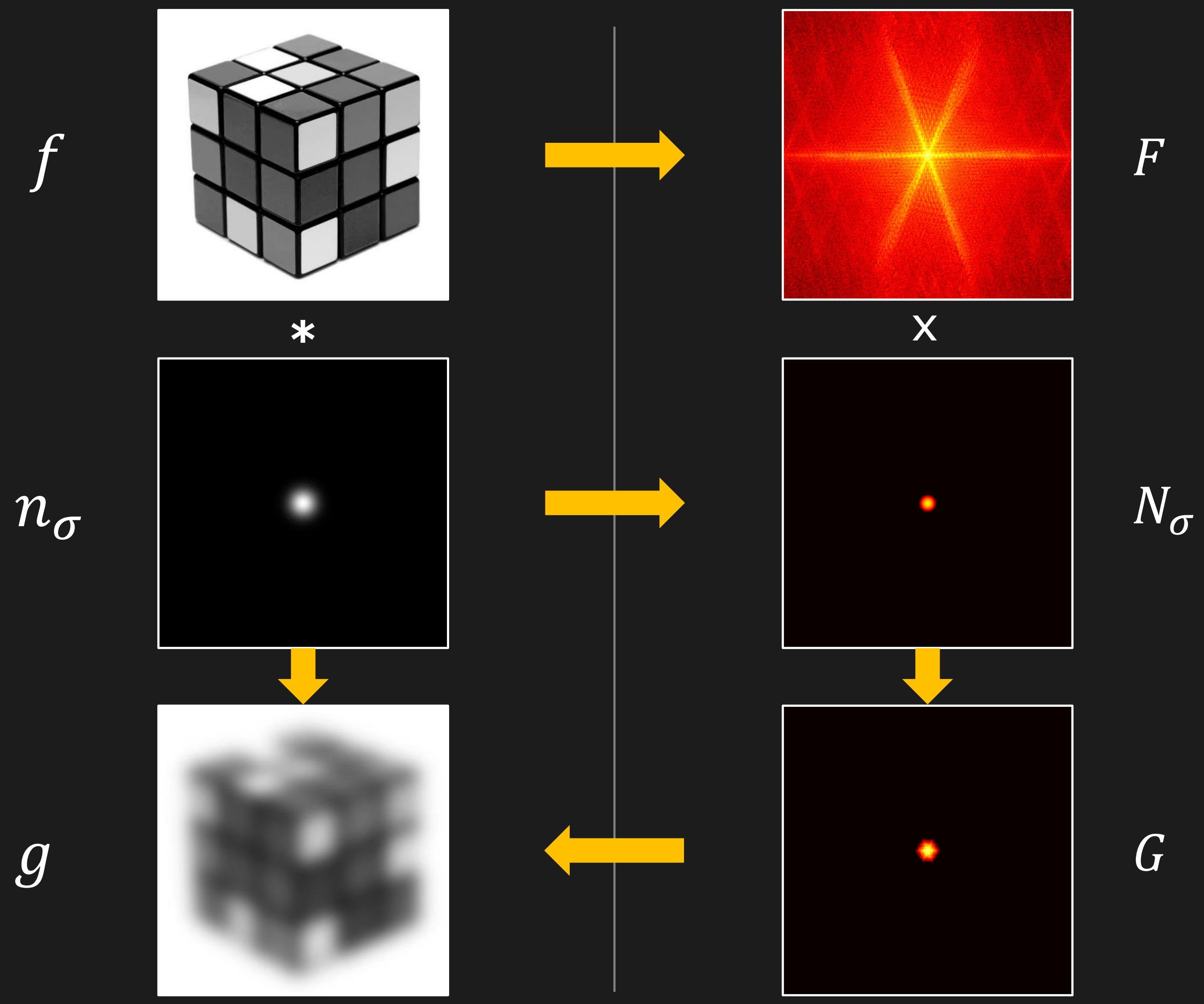
$f(m,n)$ $\qquad$ $\log(|F(p,q)|)$



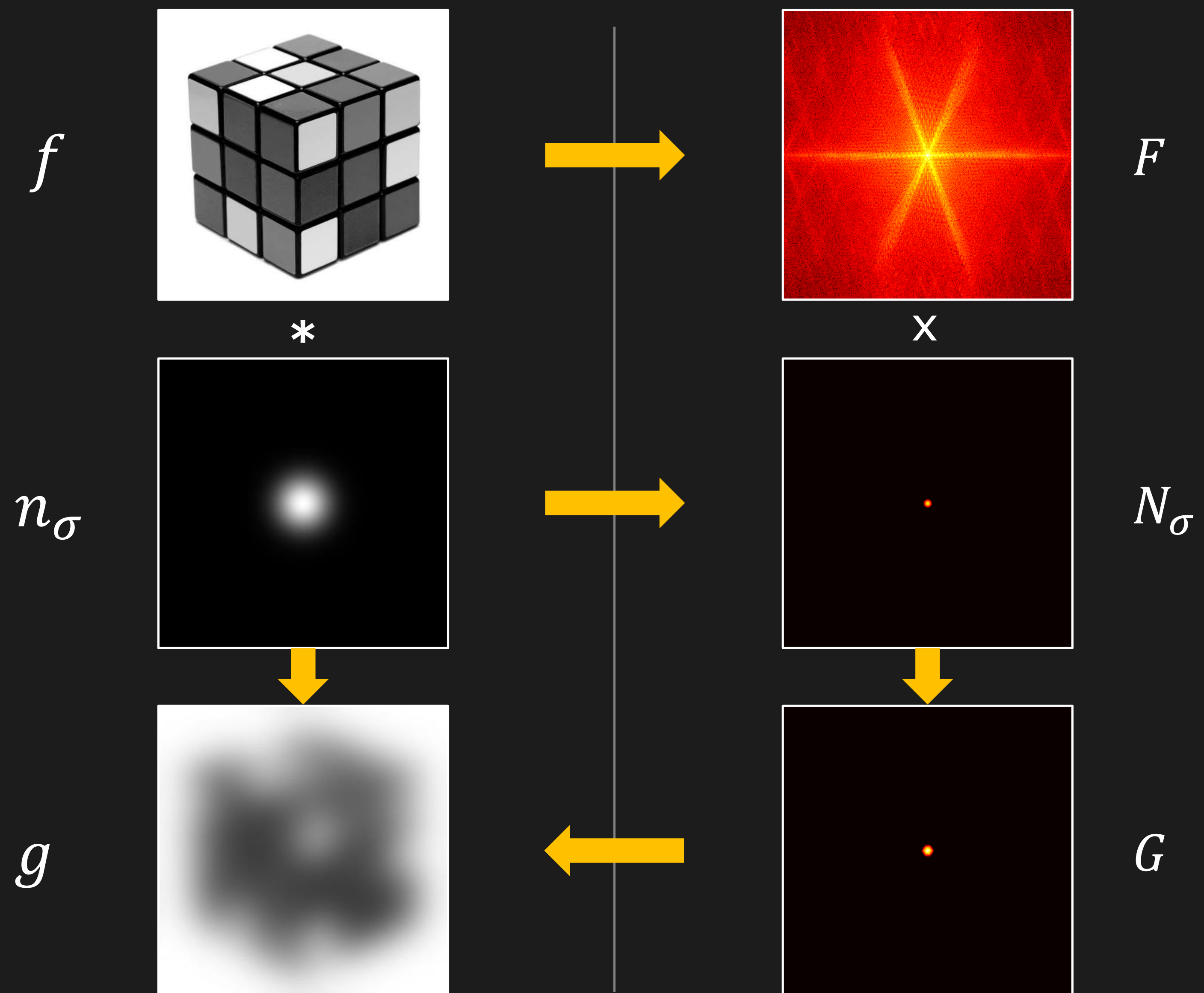
Gaussian Smoothing



Gaussian Smoothing



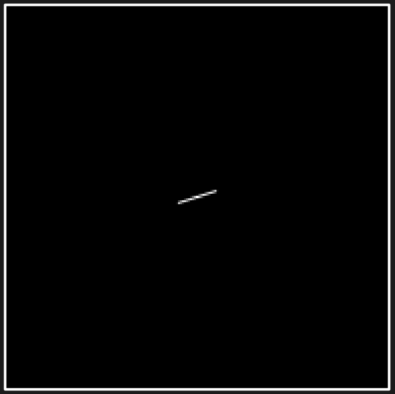
Gaussian Smoothing



Motion Blur

Scene $f(x,y)$

$*$



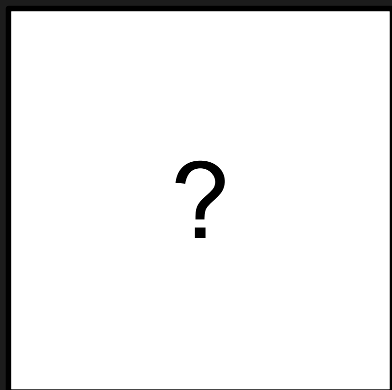
PSF $h(x,y)$
(Camera Shake)

$=$

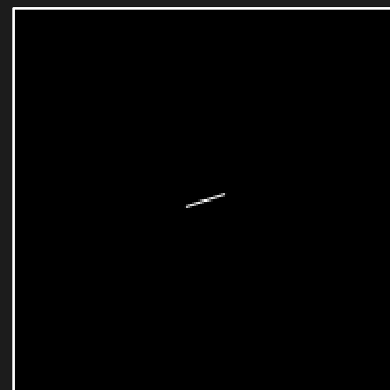
Image $g(x,y)$

$$\boxed{f(x,y) * h(x,y) = g(x,y)}$$

Motion Blur



*



=

Scene $f(x, y)$

PSF $h(x, y)$
(Camera Shake)

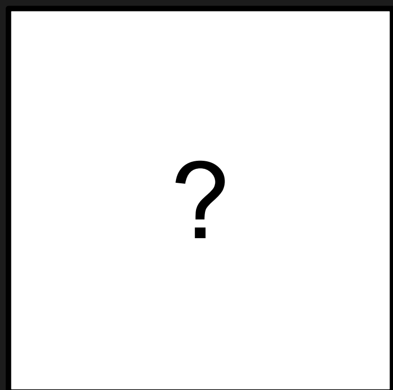
Image $g(x, y)$

$$f(x, y) * h(x, y) = g(x, y)$$

Given captured image $g(x, y)$ and PSF $h(x, y)$,
can we estimate actual scene $f(x, y)$?

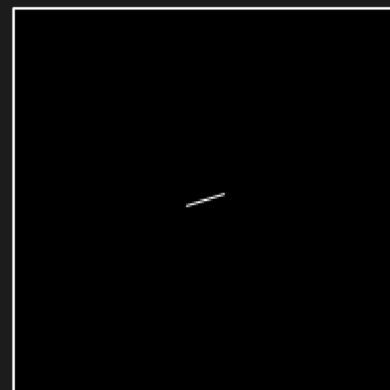
Fourier Transform To the Rescue!

Motion Deblur: Deconvolution



Scene $f(x,y)$

$*$



PSF $h(x,y)$
(Camera Shake)

$=$

Image $g(x,y)$

Let f' be the recovered scene.

$$f'(x,y) * h(x,y) = g(x,y)$$

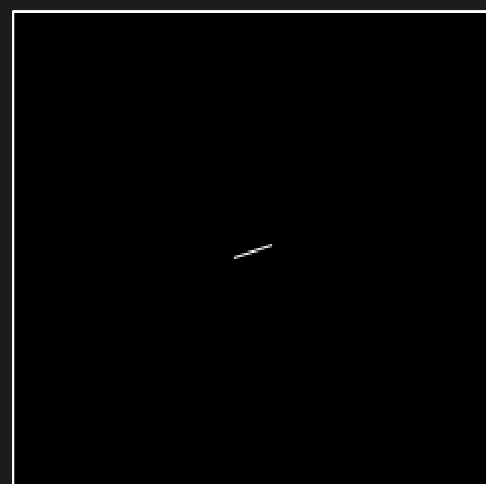
$$F'(u,v)H(u,v) \quad = G(u,v)$$

$$F'(u,v) = \frac{G(u,v)}{H(u,v)} \longrightarrow \boxed{\mathrm{IFT}} \longrightarrow f'(x,y)$$

Motion Deblur: Deconvolution

$$F'(u,v) = \frac{G(u,v)}{H(u,v)} \longrightarrow \boxed{\text{IFT}} \longrightarrow f'(x,y)$$

/



=

?

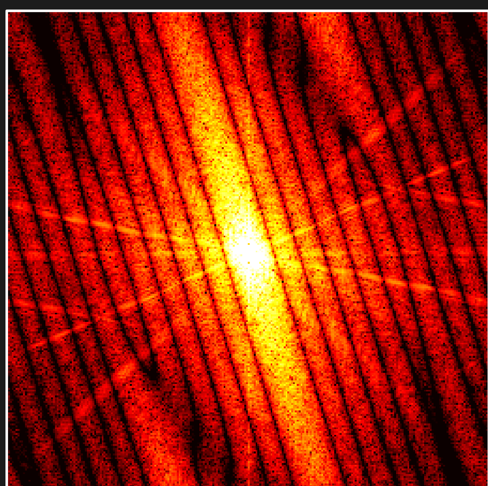
Image $g(x,y)$

PSF $h(x,y)$

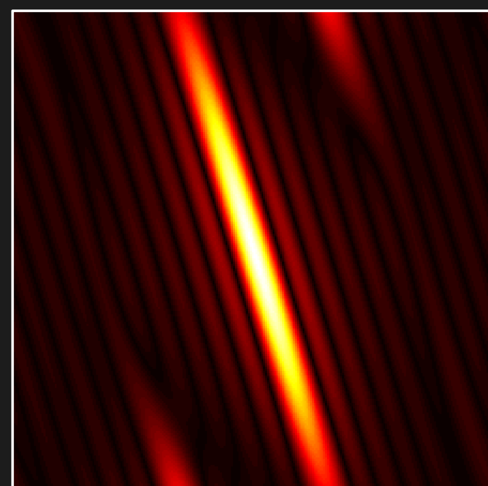
Recovered $f'(x,y)$

Motion Deblur: Deconvolution

$$F'(u,v) = \frac{G(u,v)}{H(u,v)} \longrightarrow \boxed{\text{IFT}} \longrightarrow f'(x,y)$$



/



=

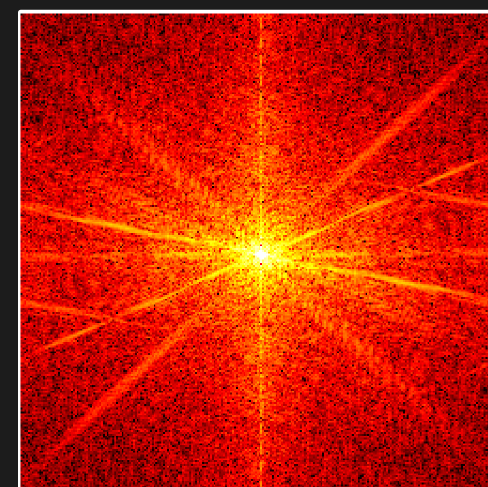


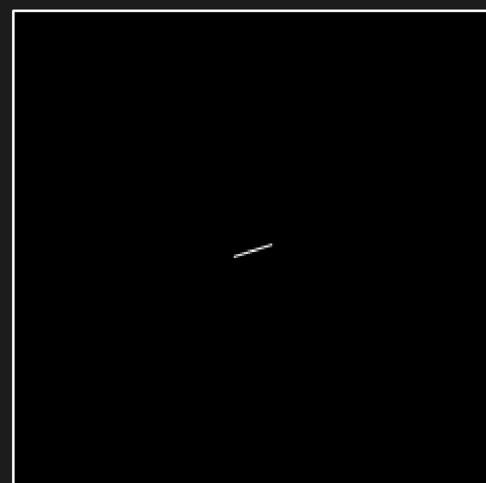
Image $G(u,v)$ | PSF $H(u,v)$ | Recovered $F'(u,v)$

Step 1: Recover $F'(u,v)$ in Fourier Domain

Motion Deblur: Deconvolution

$$F'(u,v) = \frac{G(u,v)}{H(u,v)} \longrightarrow \boxed{\text{IFT}} \longrightarrow f'(x,y)$$

/

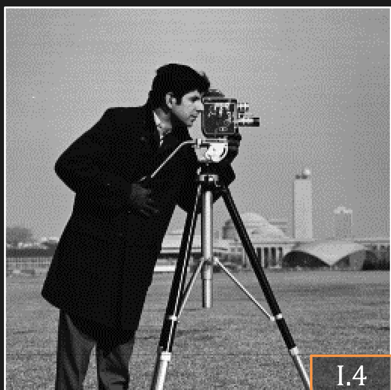


=

Image $g(x,y)$ PSF $h(x,y)$ Recovered $f'(x,y)$

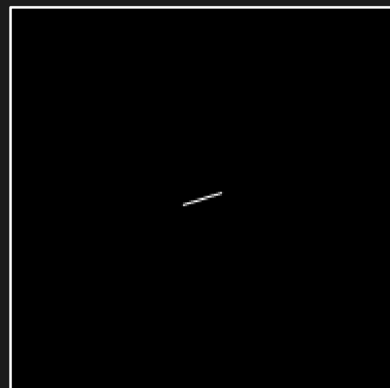
Step 2: Compute IFT of $F'(u,v)$ to recover scene

Adding Noise to the Problem



Scene $f(x, y)$

$*$



PSF $h(x, y)$
(Camera Shake)

$+$

Noise $\eta(x, y)$

$=$

Image $g(x, y)$

$$\boxed{f(x, y) * h(x, y) + \eta(x, y) = g(x, y)}$$

Can we afford to ignore noise?

Motion Deblur: Deconvolution

If we ignore the noise ($\eta(x,y)$):

$$\frac{G(u,v)}{H(u,v)} = F'(u,v) \longrightarrow \boxed{\text{IFT}} \longrightarrow f'(x,y)$$

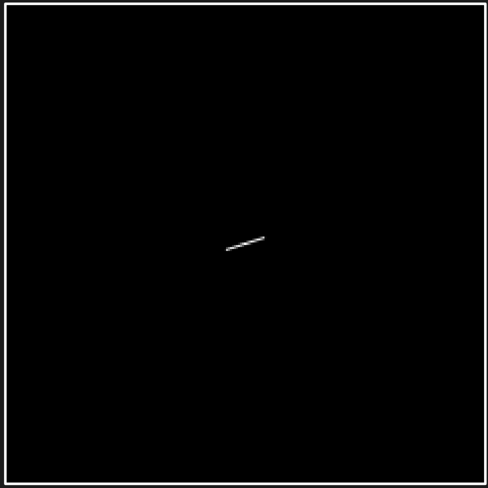
 /  = ?

Image $g(x,y)$
(with noise)

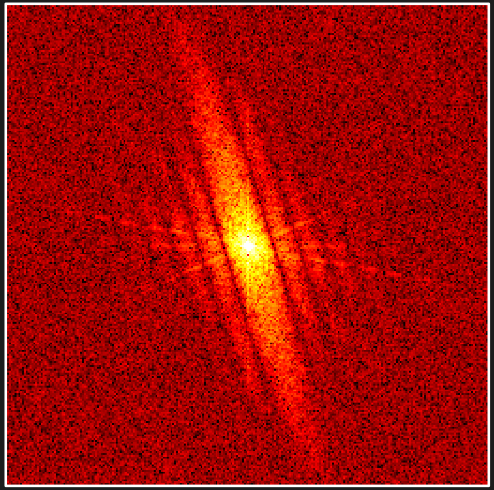
PSF $h(x,y)$

Recovered $f'(x,y)$

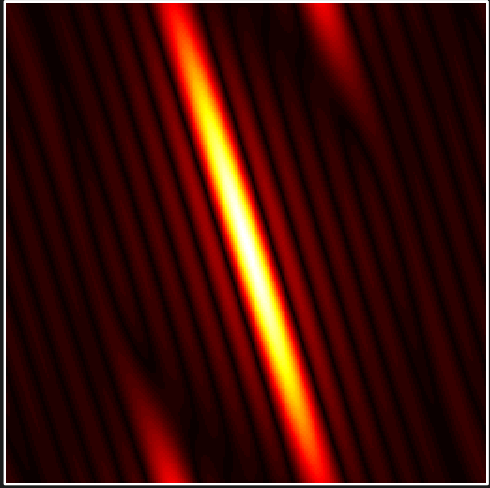
Motion Deblur: Deconvolution

If we ignore the noise $(\eta(x, y))$:

$$\frac{G(u,v)}{H(u,v)} = F'(u,v) \longrightarrow \boxed{\text{IFT}} \longrightarrow f'(x,y)$$



/



=

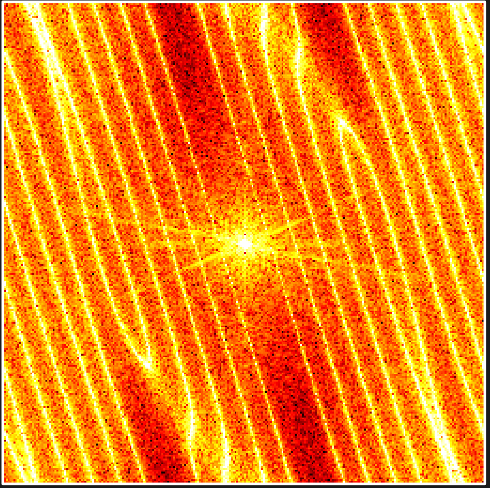


Image $G(u,v)$

PSF $H(u,v)$

Recovered $F'(u,v)$

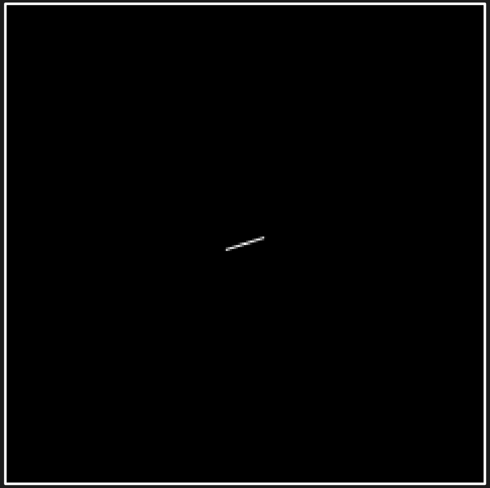
Higher frequencies in $F'(u,v)$ are amplified

Motion Deblur: Deconvolution

If we ignore the noise ($\eta(x, y)$):

$$\frac{G(u,v)}{H(u,v)} = F'(u,v) \longrightarrow \boxed{\mathrm{IFT}} \longrightarrow f'(x,y)$$

/



=

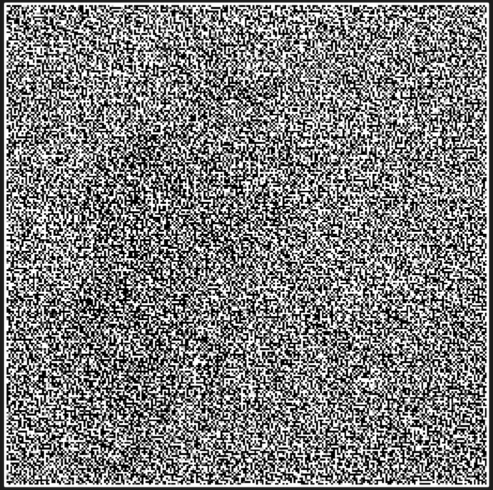


Image $g(x,y)$ (with noise)

PSF $h(x,y)$

Recovered $f'(x,y)$

Noise is significantly amplified

Deconvolution: Issues

$$\frac{G(u,v)}{H(u,v)} = F'(u,v) \longrightarrow \boxed{\text{IFT}} \longrightarrow f'(x,y)$$

- Where $H(u,v) = 0$, $F'(u,v) = \infty$
- Where $H(u,v) \approx 0$, noise in $G(u,v)$ is amplified

We need some kind of Noise Suppression.

Noise Suppression: Weiner Deconvolution

$$\boxed{F'(u,v) = \frac{G(u,v)}{H(u,v)} \left[\frac{1}{1 + \frac{NSR(u,v)}{|H(u,v)|^2}}\right]}$$

Where:

$$\boxed{\left[\frac{1}{1 + \frac{NSR(u,v)}{|H(u,v)|^2}}\right] = W(u,v)}$$

(Weiner Filter)

Noise-to-Signal Ratio, $NSR(u,v) = \dfrac{\text{Power of Noise at } (u,v)}{\text{Power of Signal (Scene) at } (u,v)}$

Noise Suppression: Weiner Deconvolution

$$\boxed{F'(u,v) = \frac{G(u,v)}{H(u,v)} \left[\frac{1}{1 + \frac{NSR(u,v)}{|H(u,v)|^2}} \right]}$$

- When there is no noise, $NSR \to 0$; $F(u,v) = \frac{G(u,v)}{H(u,v)}$
- When there is noise, $NSR > 0$; $W(u,v) < 1$ and acts as an attenuator.

Noise Suppression: Weiner Deconvolution

$$\boxed{F'(u,v) = \frac{G(u,v)}{H(u,v)} \left[\frac{1}{1 + \frac{NSR(u,v)}{|H(u,v)|^2}} \right]}$$

- Determination of NSR requires that we have prior knowledge of the noise “pattern” and the scene (or that of a similar scene).

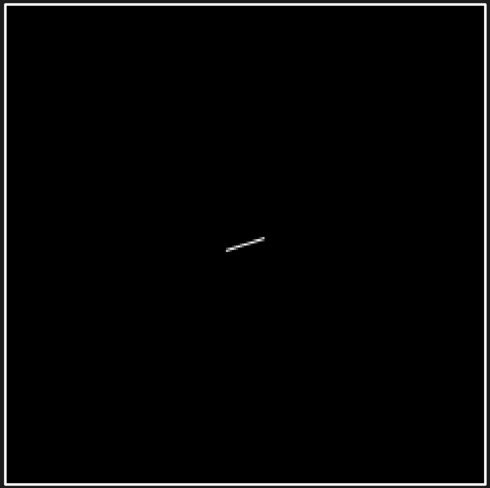
- Often NSR is set to a single suitable constant λ.

$$\boxed{NSR(u,v) = \lambda}$$

Noise Suppression: Weiner Deconvolution

Noisy, Blurred Image $g(x, y)$

/



PSF $h(x, y)$

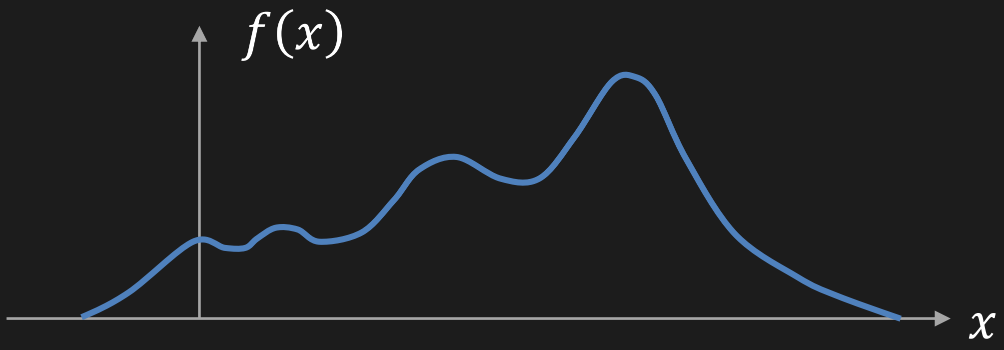
=

Recovered $f'(x, y)$

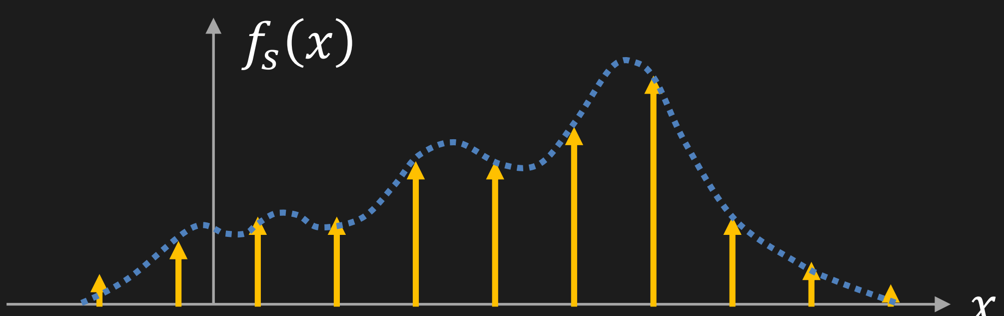
$NSR(u, v) = \lambda = 0.002$ was used to recover image

From Continuous to Digital Image

Continuous Signal:

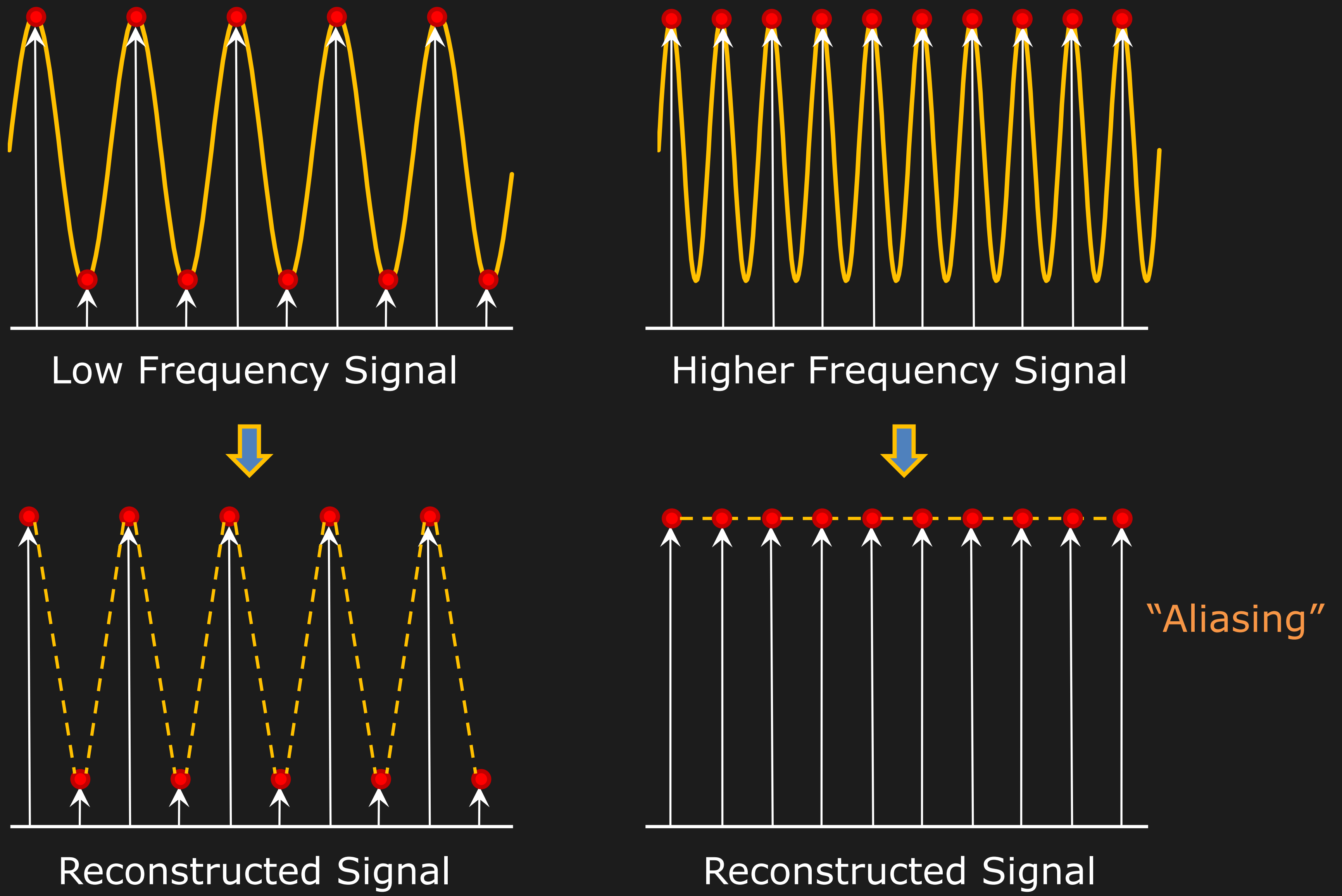


Digital Signal:



How “dense” should the samples be?

Sampling Problem



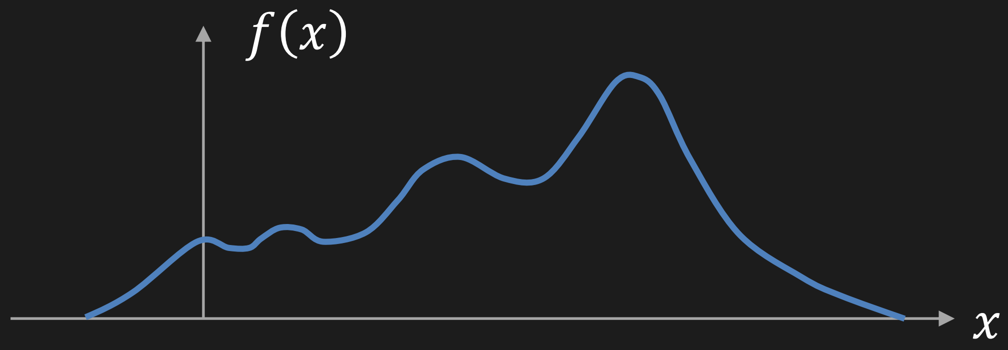
Sampling Problem

“Well sampled” image

“Under sampled” image
(visible aliasing artifacts)

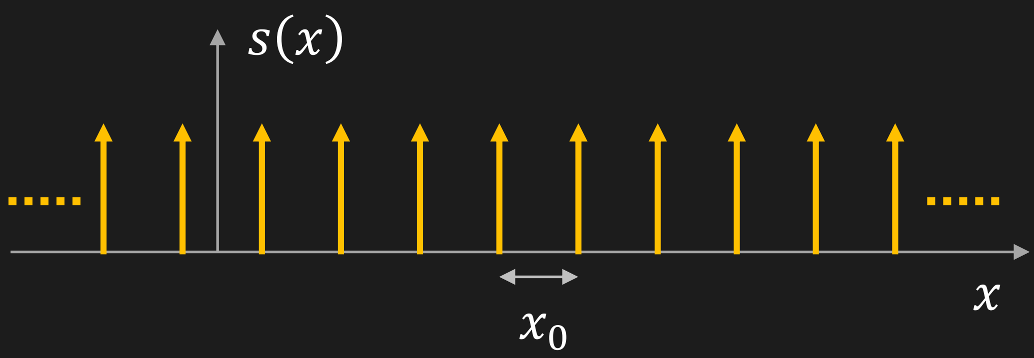
Sampling Theory

Continuous Signal:



Shah Function (Impulse Train):

$$s(x) = \sum_{n=-\infty}^{\infty} \delta(x - nx_0)$$

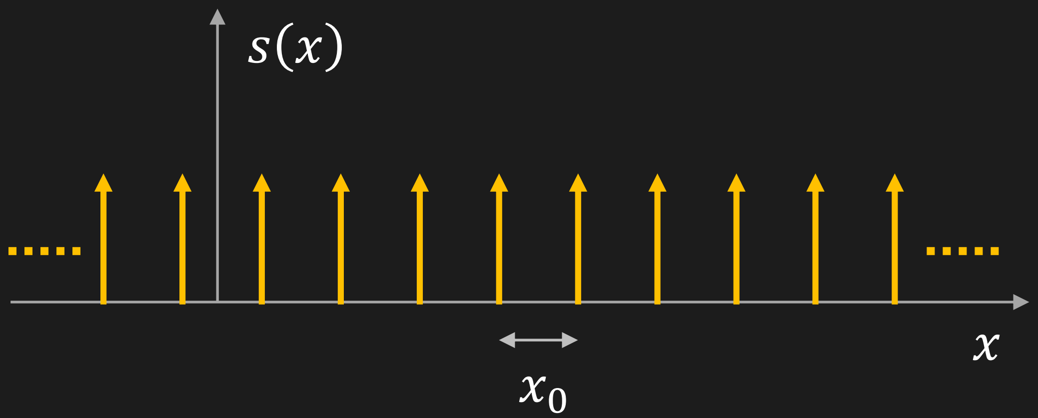


Sampled Function: $\boxed{f_s(x) = f(x)s(x)}$

Shah Function (Impulse Train)

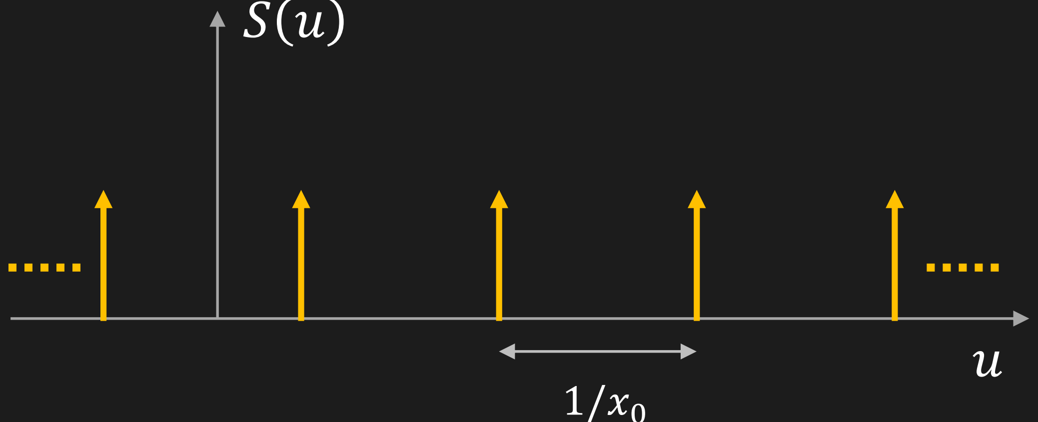
Shah Function (Spatial Domain):

$$s(x) = \sum_{n=-\infty}^{\infty} \delta(x - nx_0)$$



Shah Function (Fourier Domain):

$$S(u) = \frac{1}{x_0} \sum_{n=-\infty}^{\infty} \delta\left(x - \frac{n}{x_0}\right)$$



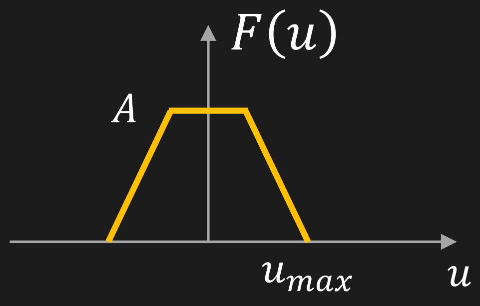
Fourier Analysis of Sampled Signal

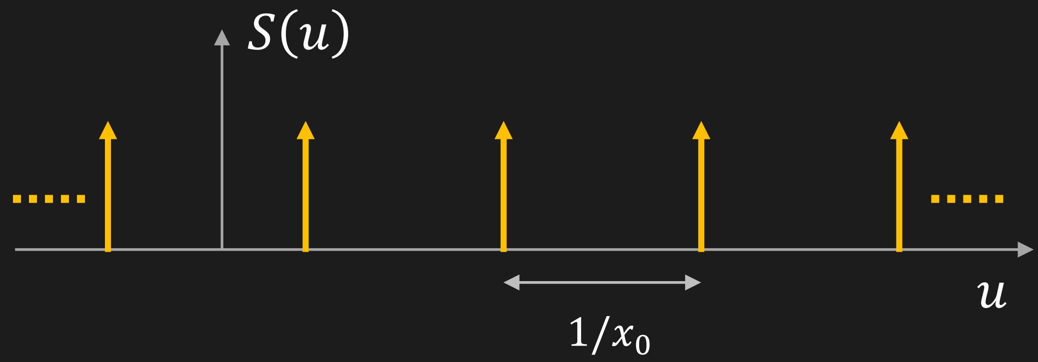
Sampled Signal:

$$f_s(x) = f(x)s(x) = f(x)\sum \delta(x - nx_0)$$

$$F_s(u) = F(u) * S(u) = F(u) * \frac{1}{x_0}\sum \delta(u - n/x_0)$$

For example:





Fourier Analysis of Sampled Signal

Sampled Signal:

$$f_s(x) = f(x)s(x) = f(x)\sum \delta(x - nx_0)$$

$$F_s(u) = F(u) * S(u) = F(u) * \frac{1}{x_0}\sum \delta(u - n/x_0)$$

If $u_{max} \leq \frac{1}{2x_0}$

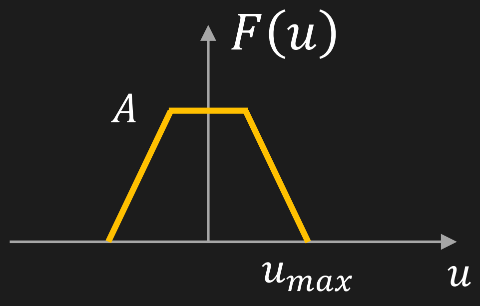
Aliasing

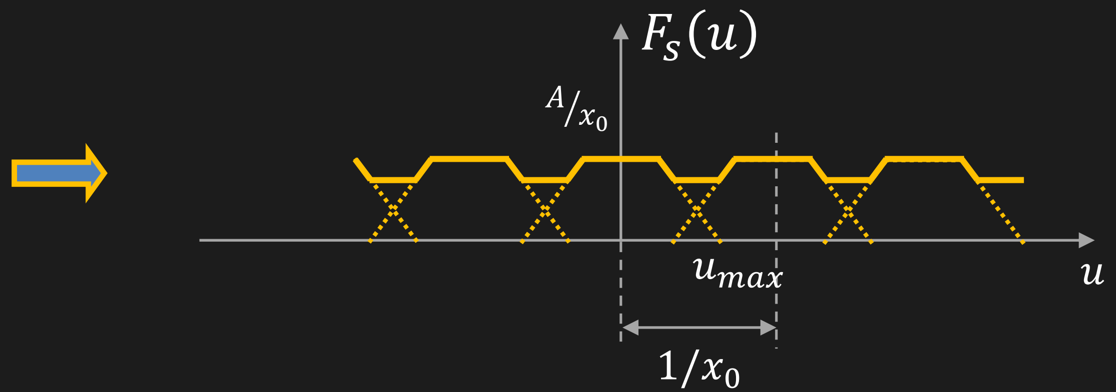
Sampled Signal:

$$f_s(x) = f(x)s(x) = f(x)\sum \delta(x - nx_0)$$

$$F_s(u) = F(u) * S(u) = F(u) * \frac{1}{x_0}\sum \delta(u - n/x_0)$$

If $u_{max} > \dfrac{1}{2x_0}$ Aliasing

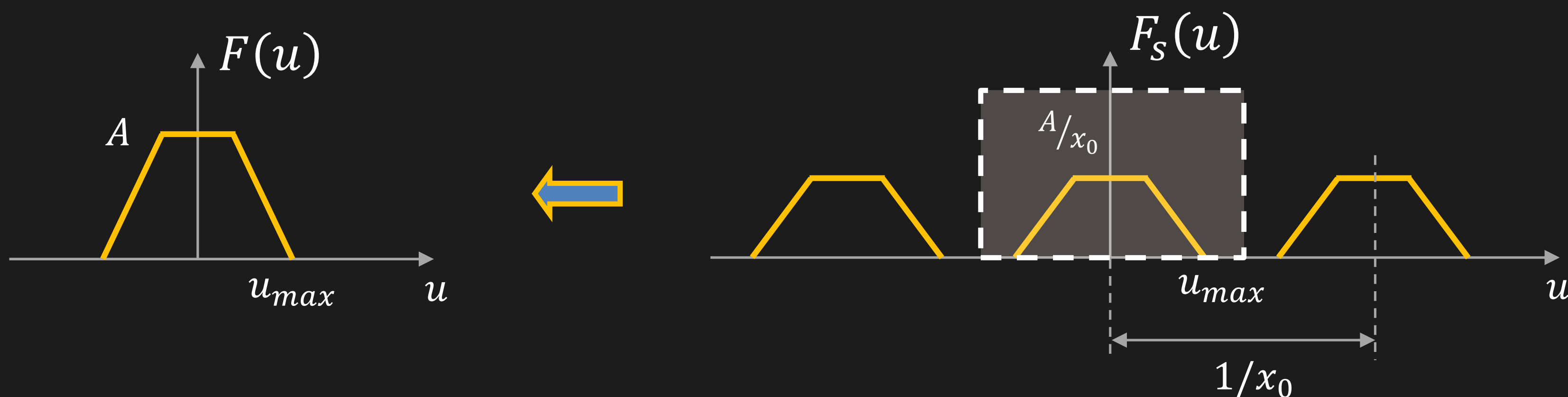




Nyquist Theorem

Can we recover $f(x)$ from $f_s(x)$? In other words, can we recover $F(u)$ from $F_s(u)$?

Only if $u_{max} \leq \frac{1}{2x_0}$ (Nyquist Frequency)



$$F(u) = F_s(u)C(u)$$

$$C(u) = \begin{cases} x_0, & |u| < 1/2x_0 \\ 0, & Otherwise \end{cases}$$

$$\boxed{f(x) = IFT\big(F(u)\big)}$$

References: Textbooks

Digital Image Processing (Chapter 3 and 4)
González, R and Woods, R., Prentice Hall

Computer Vision: A Modern Approach (Chapter 7)
Forsyth, D and Ponce, J., Prentice Hall

Robot Vision (Chapter 6 and 7)
Horn, B. K. P., MIT Press

Image Credits

I.1 http://en.wikipedia.org/wiki/File:Fourier2.jpg

I.2 http://www.instructables.com/image/FY1T8VKG79F1MO7/Rubiks-cube-pranks.jpg

I.3 Matlab Demo Image

I.4 Matlab Demo Image