Image Filtering in Fourier (Frequency) Domain

Computer Vision: CS 566

Computer Science

University of Wisconsin-Madison

Image Filtering in Fourier Domain

Transform image to new one that is easier to manipulate/analyze.

Topics:

(1) Frequency Representation of Signals

(2) Fourier Transform

(3) Convolution and Fourier Transform

(4) Deconvolution in Frequency Domain

(5) Sampling Theory

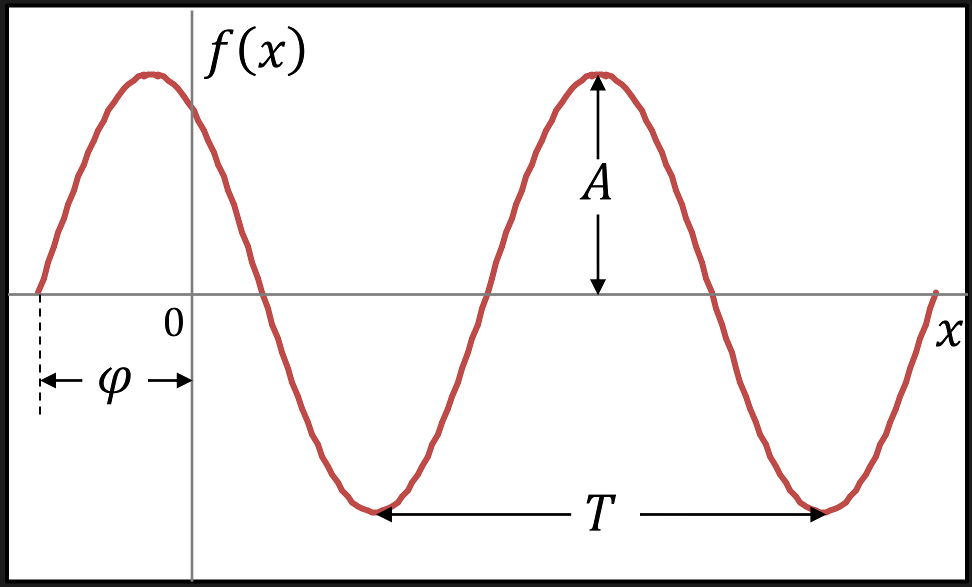
Jean Baptiste Joseph Fourier

(1768-1830)

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

Sinusoid

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



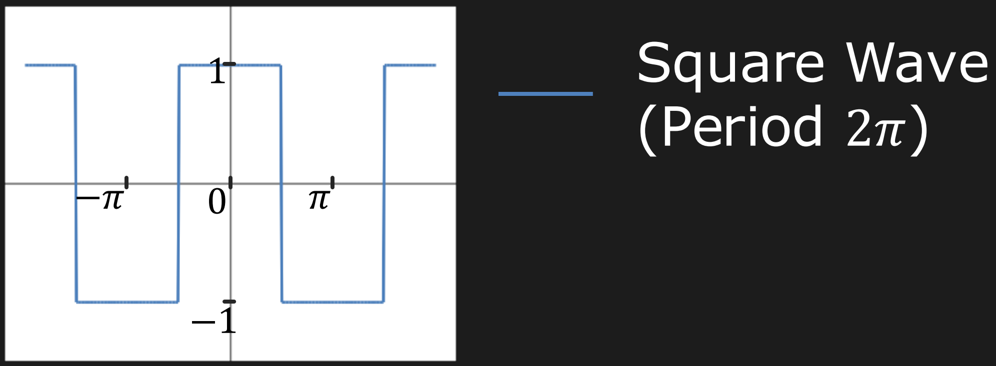
A: Amplitude

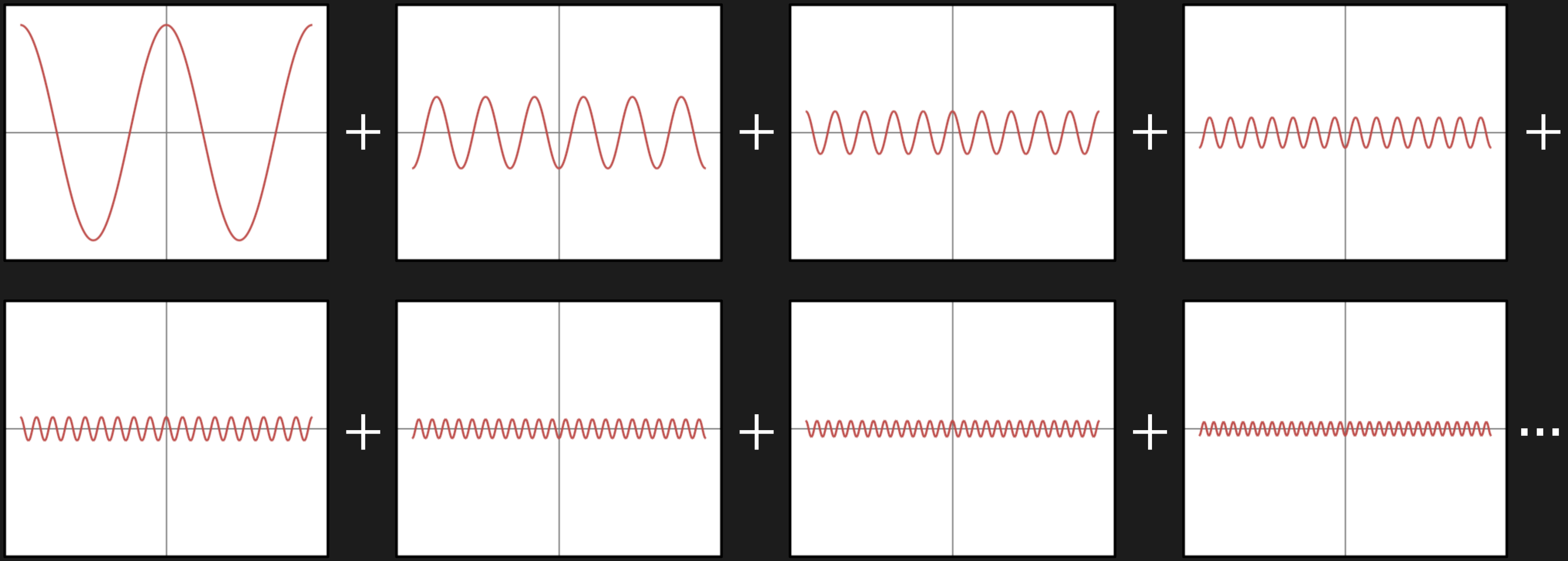
T: Period

φ: Phase

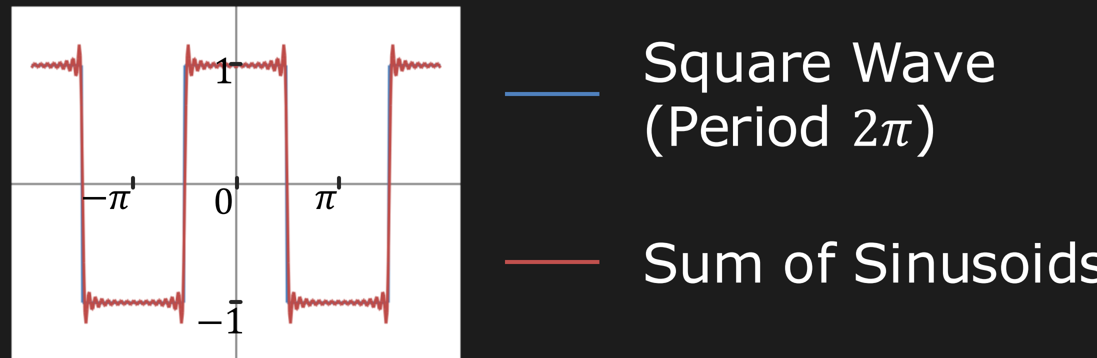
u: Frequency $(1/T)$

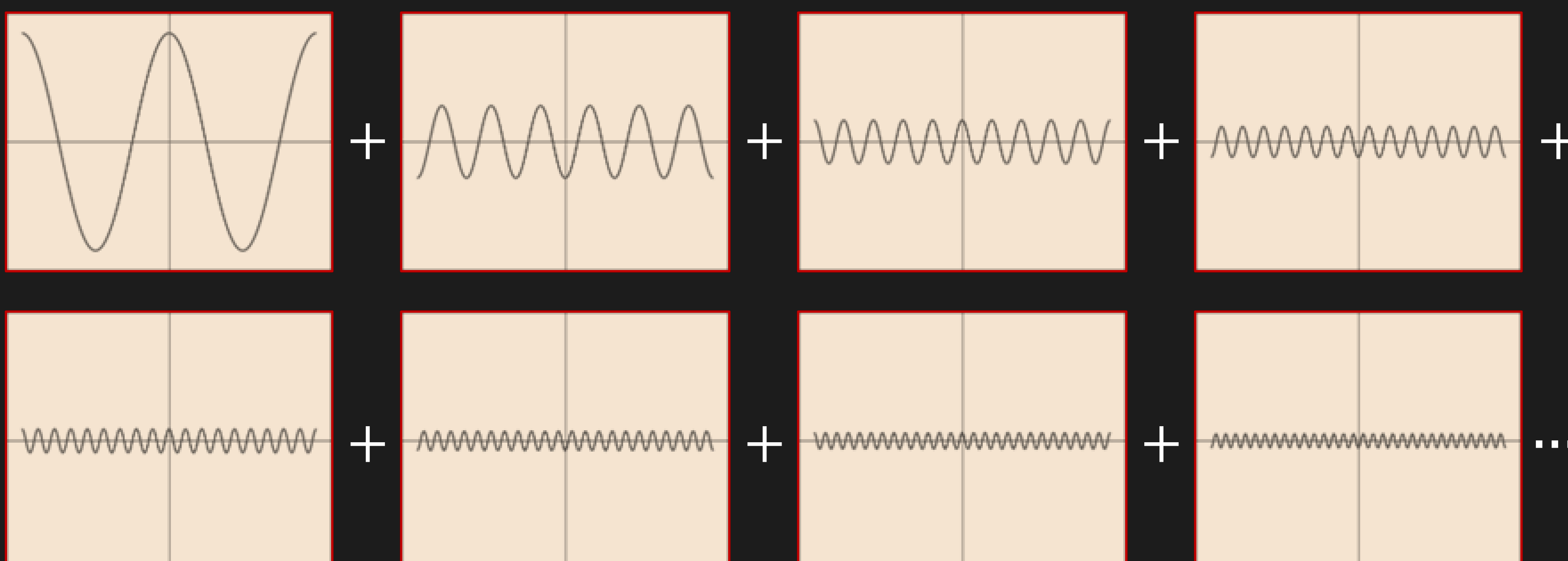
Fourier Series



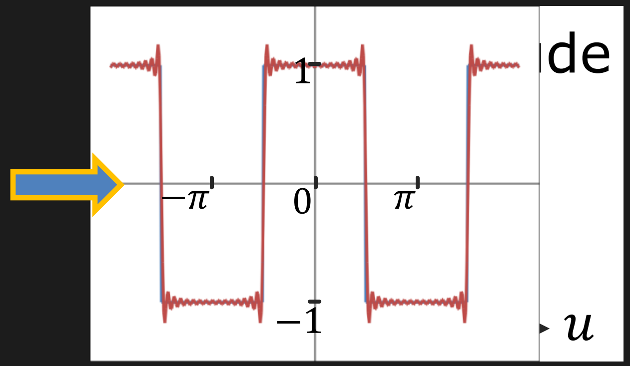


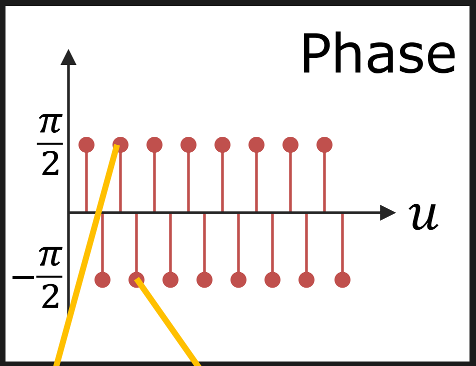
Fourier Series

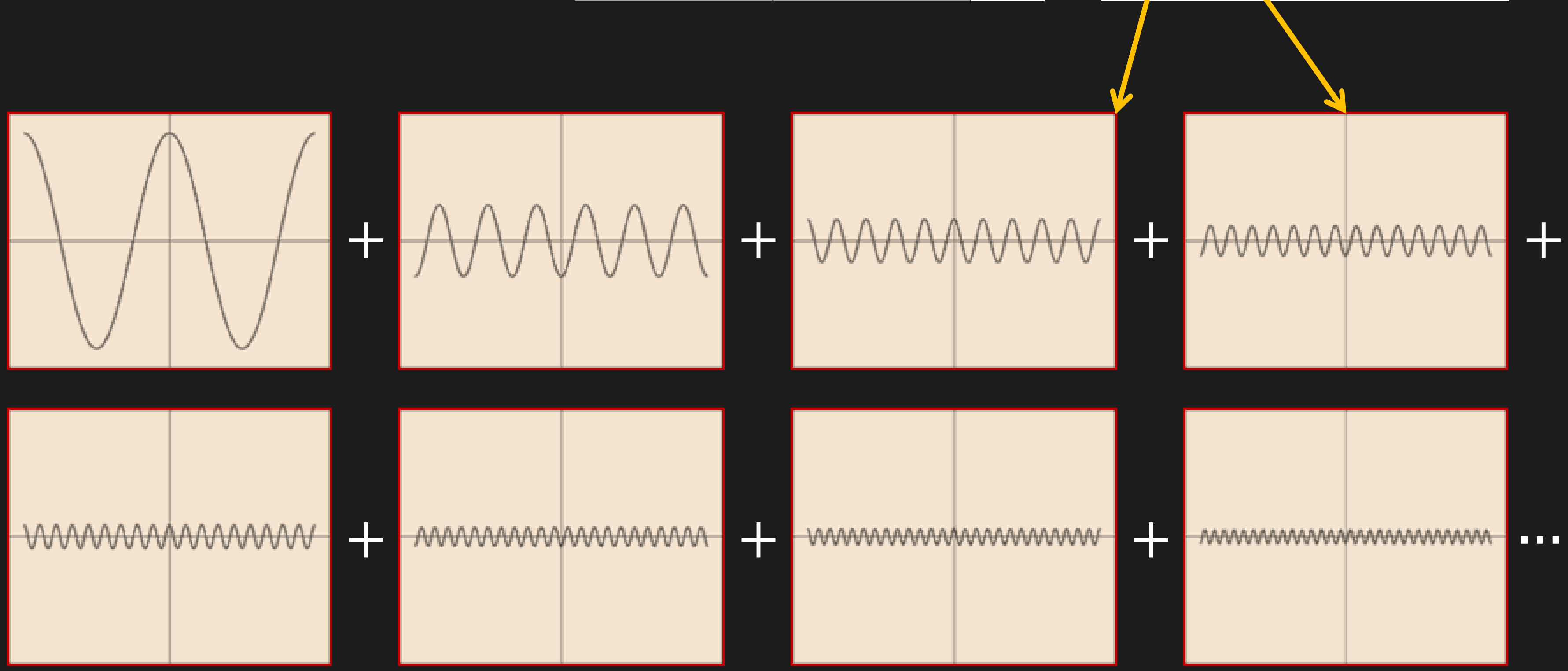




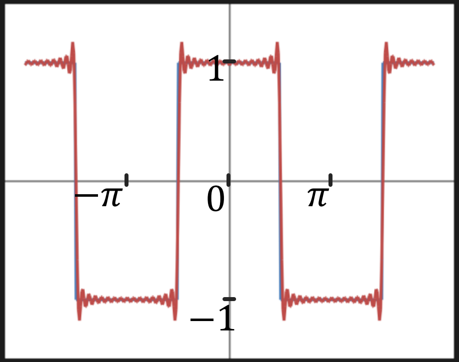
An Alternate Representation of Signal

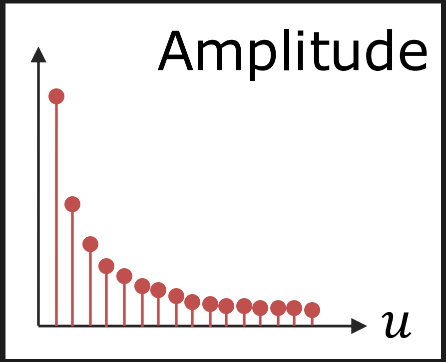


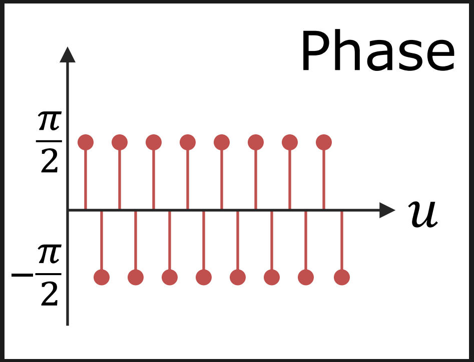




Fourier Transform (FT)



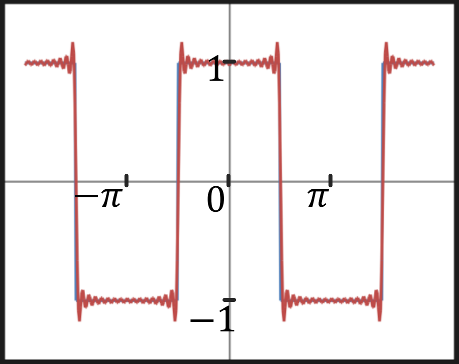


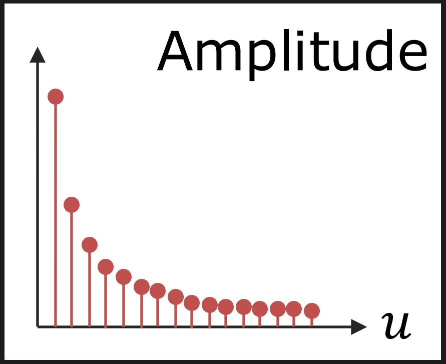


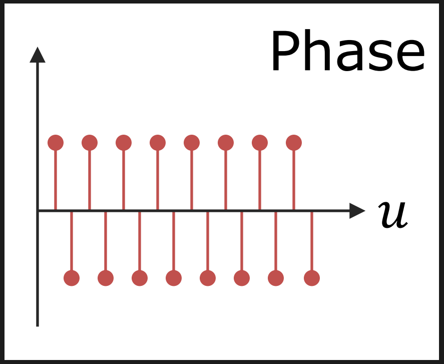
Represents a signal $f(x)$ in terms of Amplitudes and Phases of its Constituent Sinusoids.

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

Inverse Fourier Transform (IFT)







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

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

Fourier Transform is Complex!

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

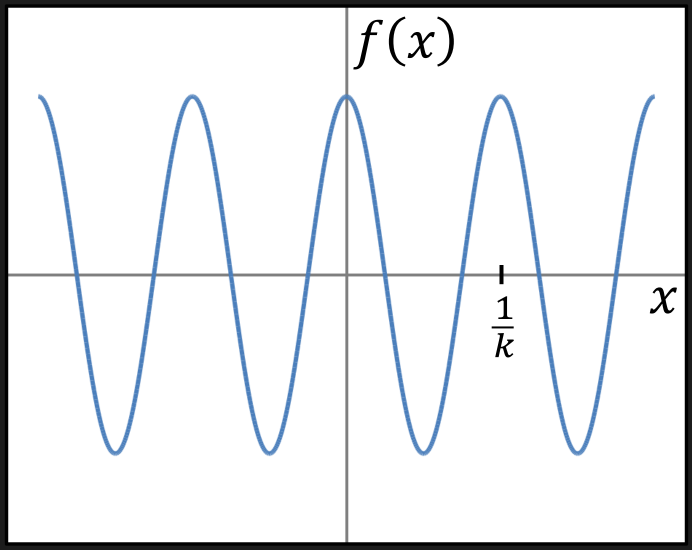
$$F(u) = \mathfrak{Re}\{F(u)\} + i\,\mathfrak{Im}\{F(u)\}$$

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

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

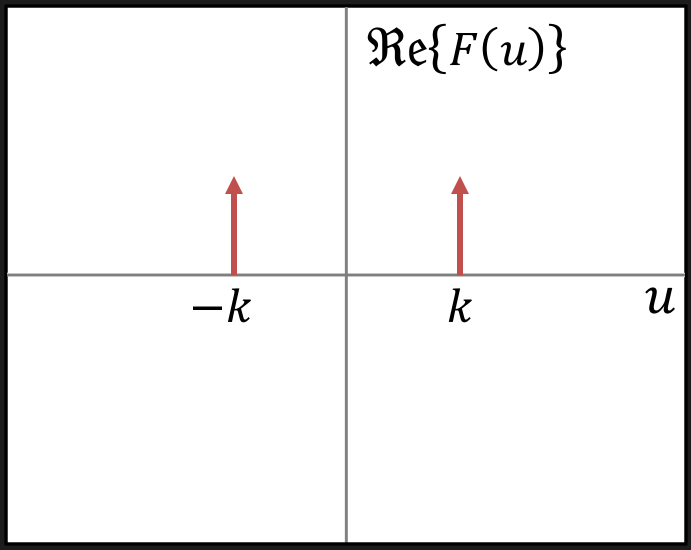
Fourier Transform Examples

Signal $f(x)$



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

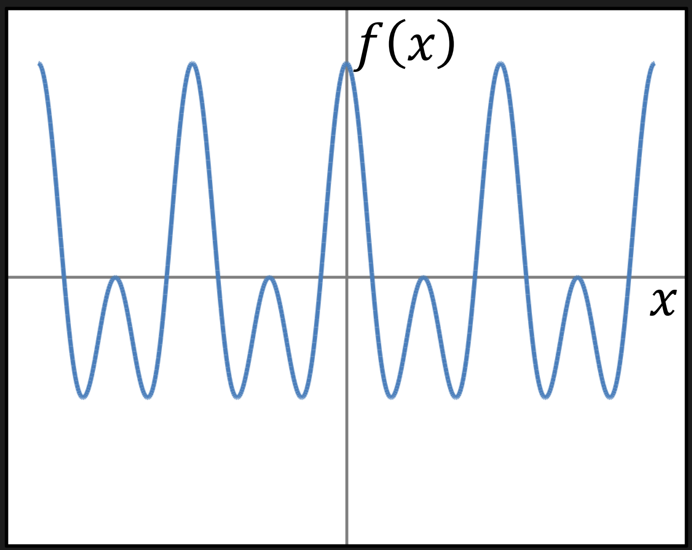
Fourier Transform $F(u)$



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

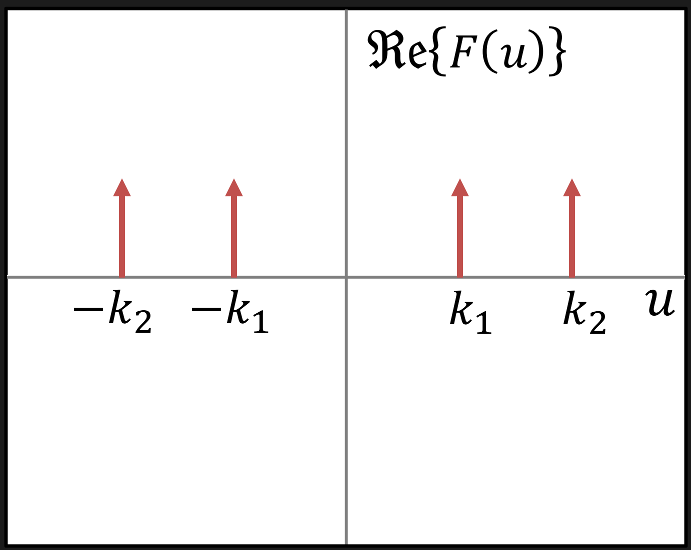
Fourier Transform Examples

Signal $f(x)$



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

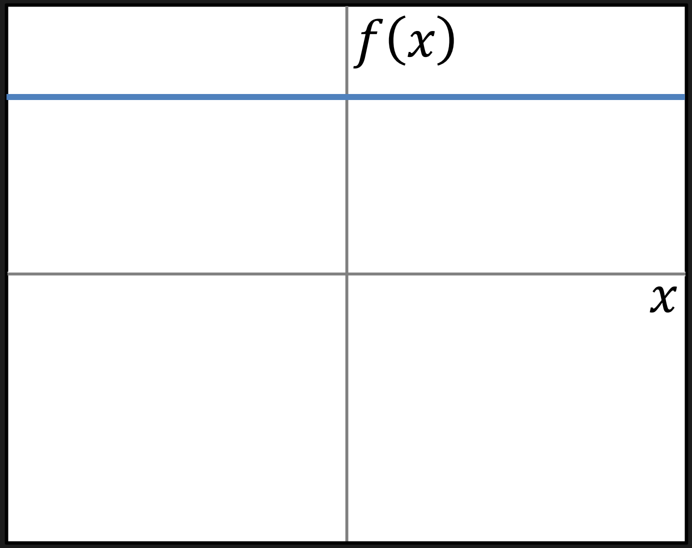
Fourier Transform $F(u)$



$$F(u) = \frac{1}{2}[\delta(u + k_1) + \delta(u - k_1) + \delta(u + k_2) + \delta(u - k_2)]$$

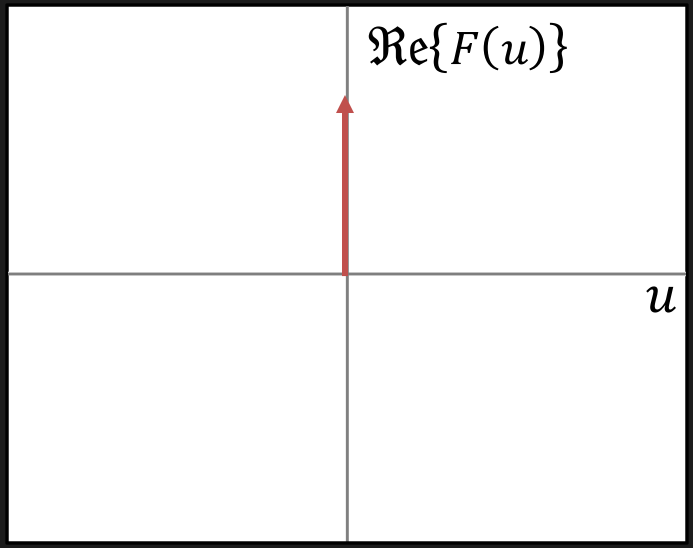
Fourier Transform Examples

Signal $f(x)$



$$f(x) = 1$$

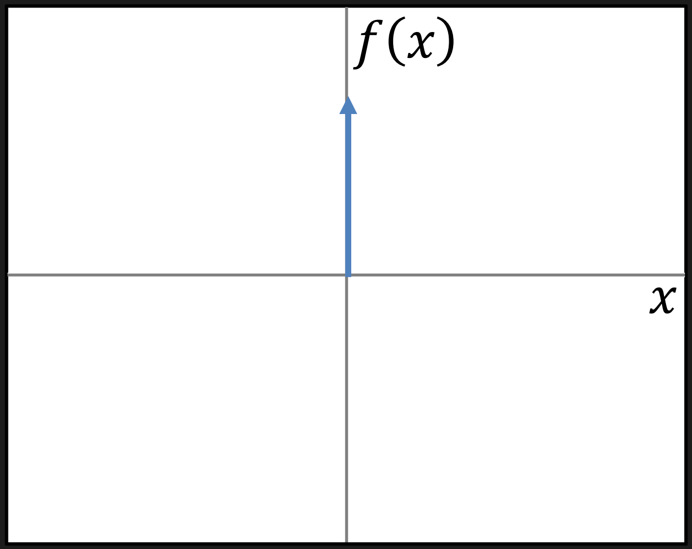
Fourier Transform $F(u)$



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

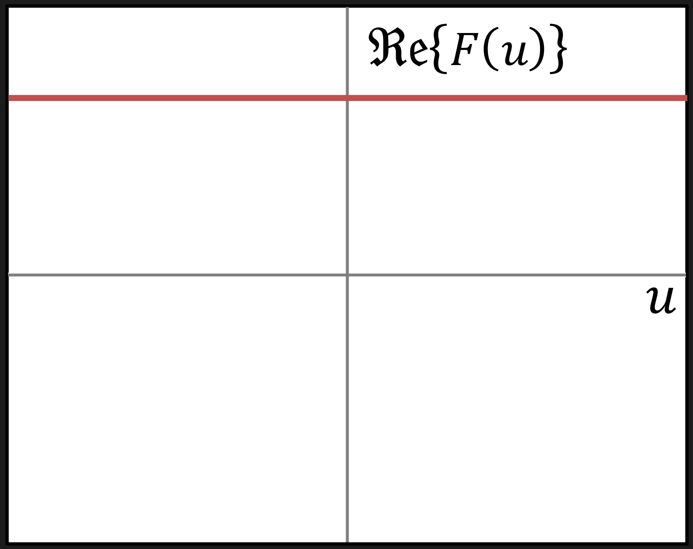
Fourier Transform Examples

Signal $f(x)$



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

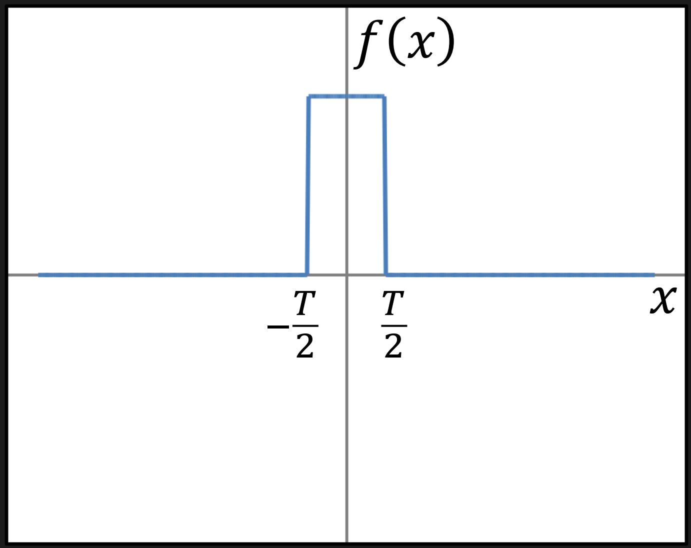
Fourier Transform $F(u)$



$$F(u) = 1$$

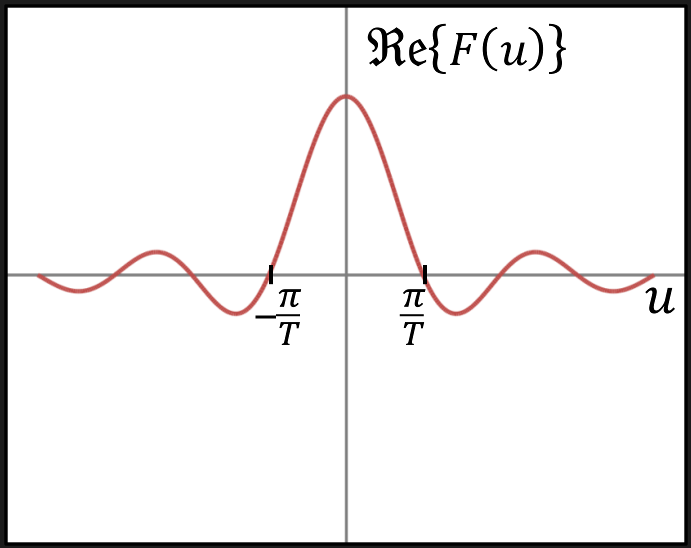
Fourier Transform Examples

Signal $f(x)$



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

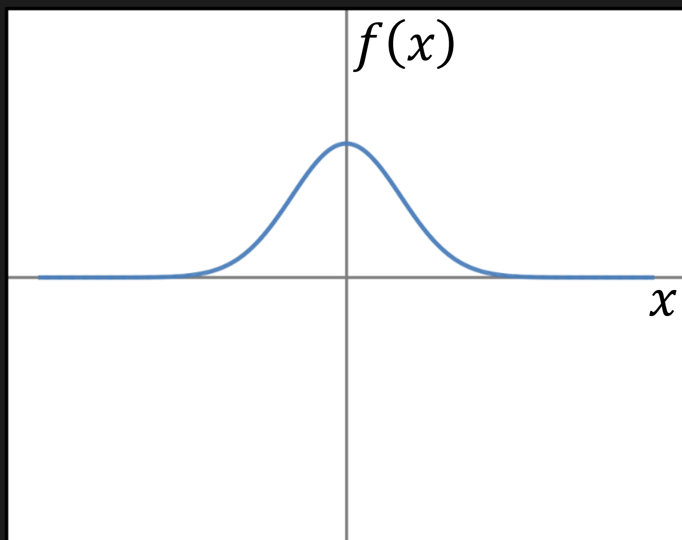
Fourier Transform $F(u)$



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

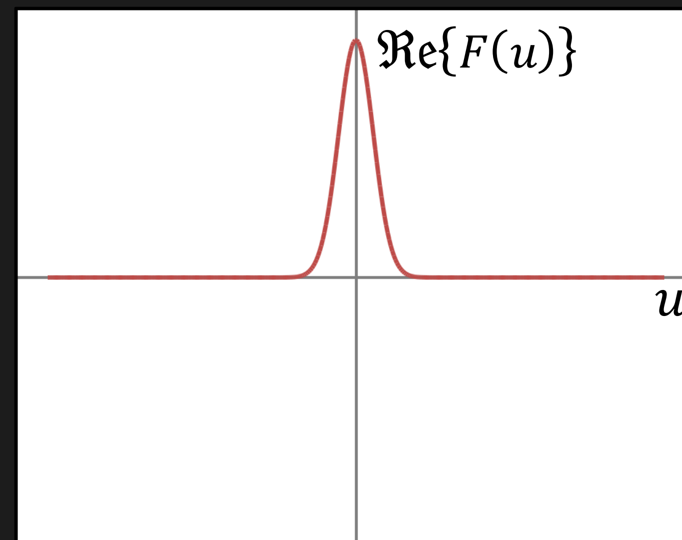
Fourier Transform Examples

Signal $f(x)$



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

Fourier Transform $F(u)$



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

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	$\frac{d^n}{dx^n}\big(f(x)\big)$	$(i2\pi u)^n F(u)$

Correlation vs. Convolution

Correlation:

$$R_{tf}[i,j] = \sum_m \sum_n f[m,n]\underline{t[m-i,n-j]} = t \otimes f$$

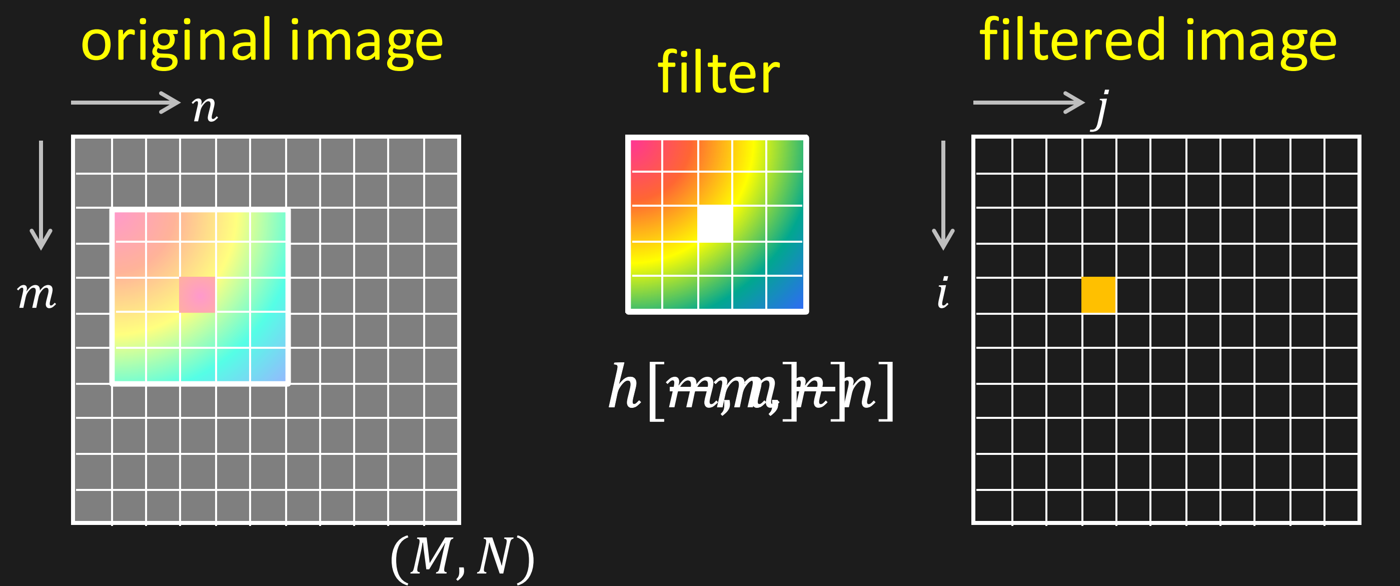
Convolution:

$$g[i,j] = \sum_m \sum_n f[m,n]\underline{t[i-m,j-n]} = t * f$$

Flipping in Convolution

Convolution with Discrete Images

$$f[m,n] \longrightarrow \boxed{h[m,n]} \longrightarrow g[i,j]$$



$$g[i,j] = \sum_{m=1}^{M} \sum_{n=1}^{N} f[m,n]\,h[i-m,j-n]$$

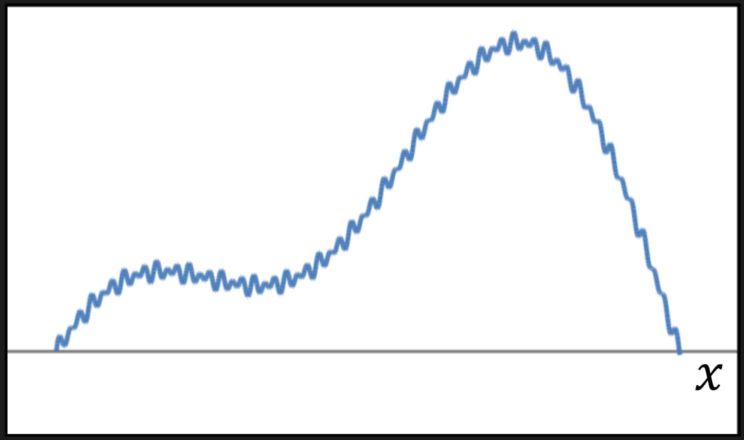
Convolution and Fourier Transform

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

Convolution Using Fourier Transform

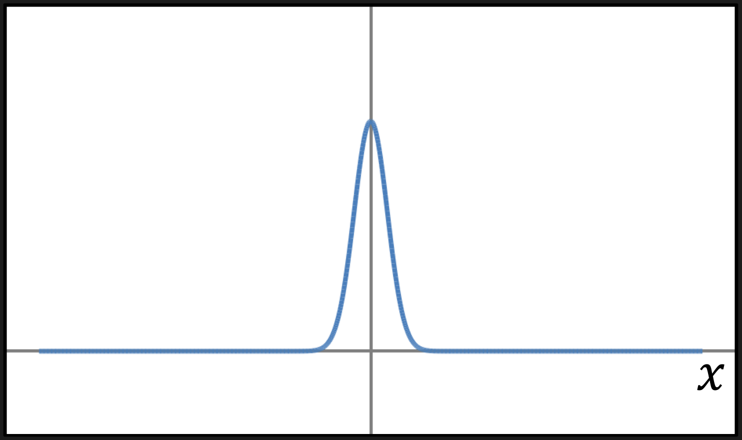
$$
\begin{array}{ccccc}
g(x) & = & f(x) & * & h(x) \\
\uparrow \boxed{\mathrm{IFT}} & & \downarrow \boxed{\mathrm{FT}} & & \downarrow \boxed{\mathrm{FT}} \\
G(u) & = & F(u) & \times & H(u)
\end{array}
$$

Gaussian Smoothing in Fourier Domain



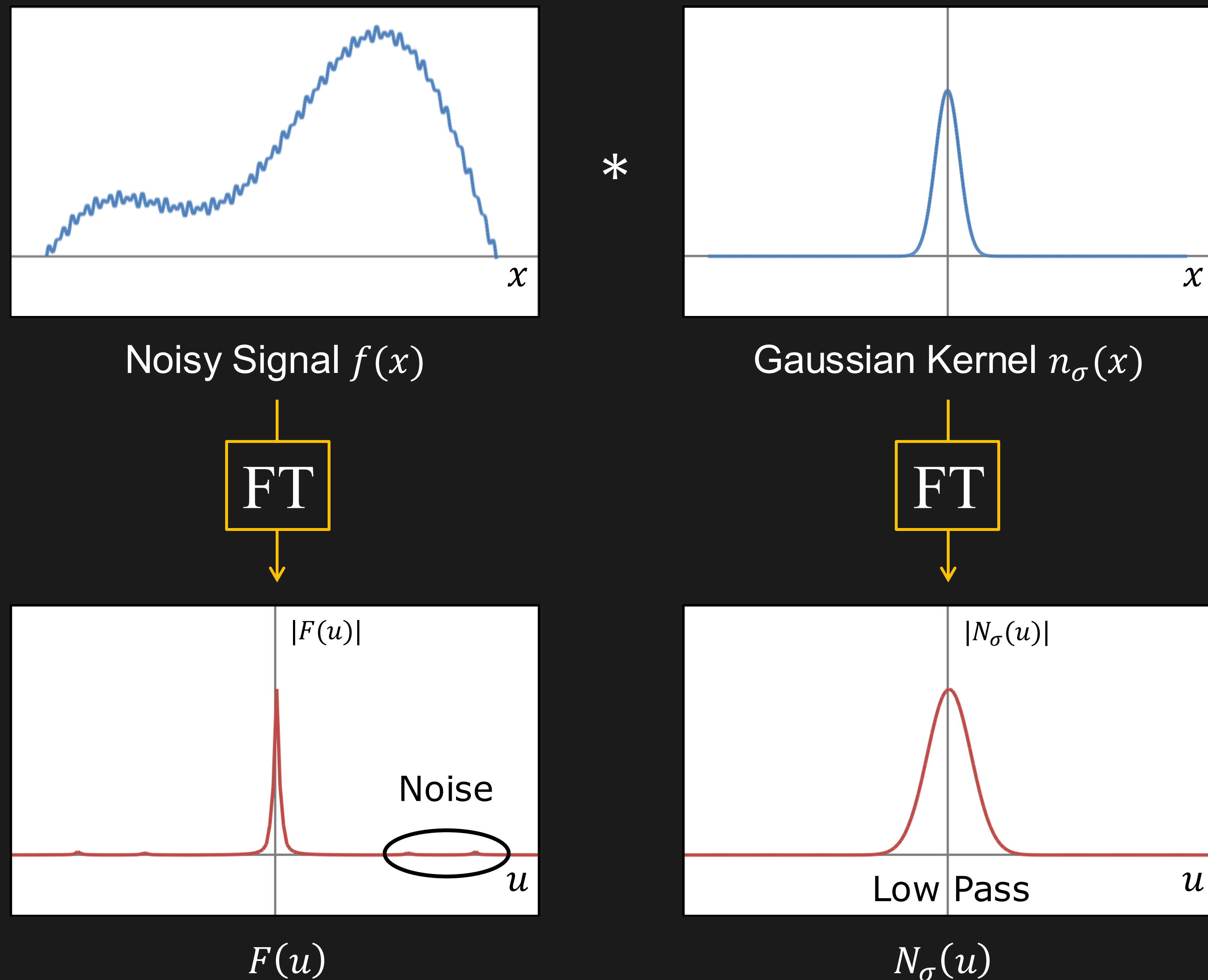
Noisy Signal $f(x)$

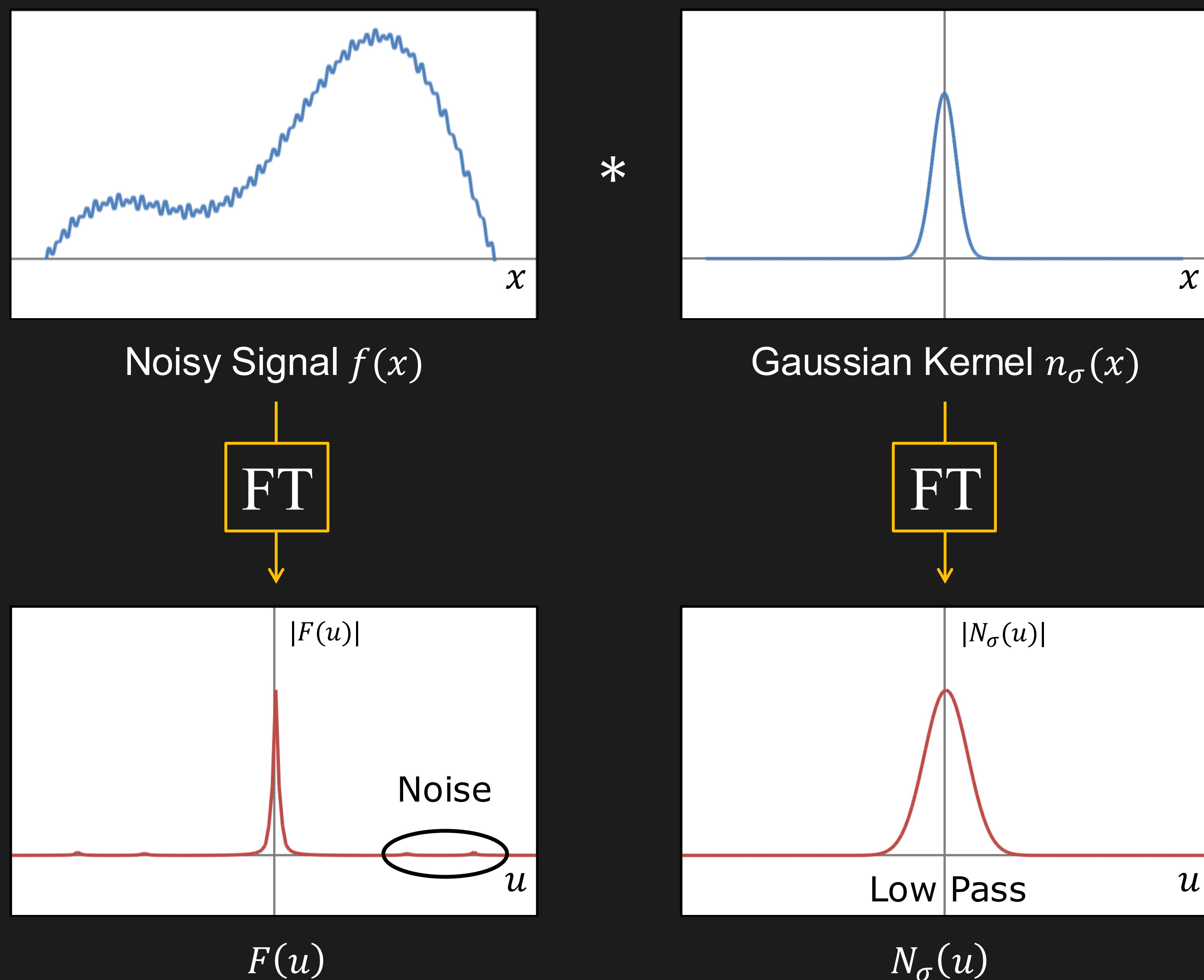
$*$



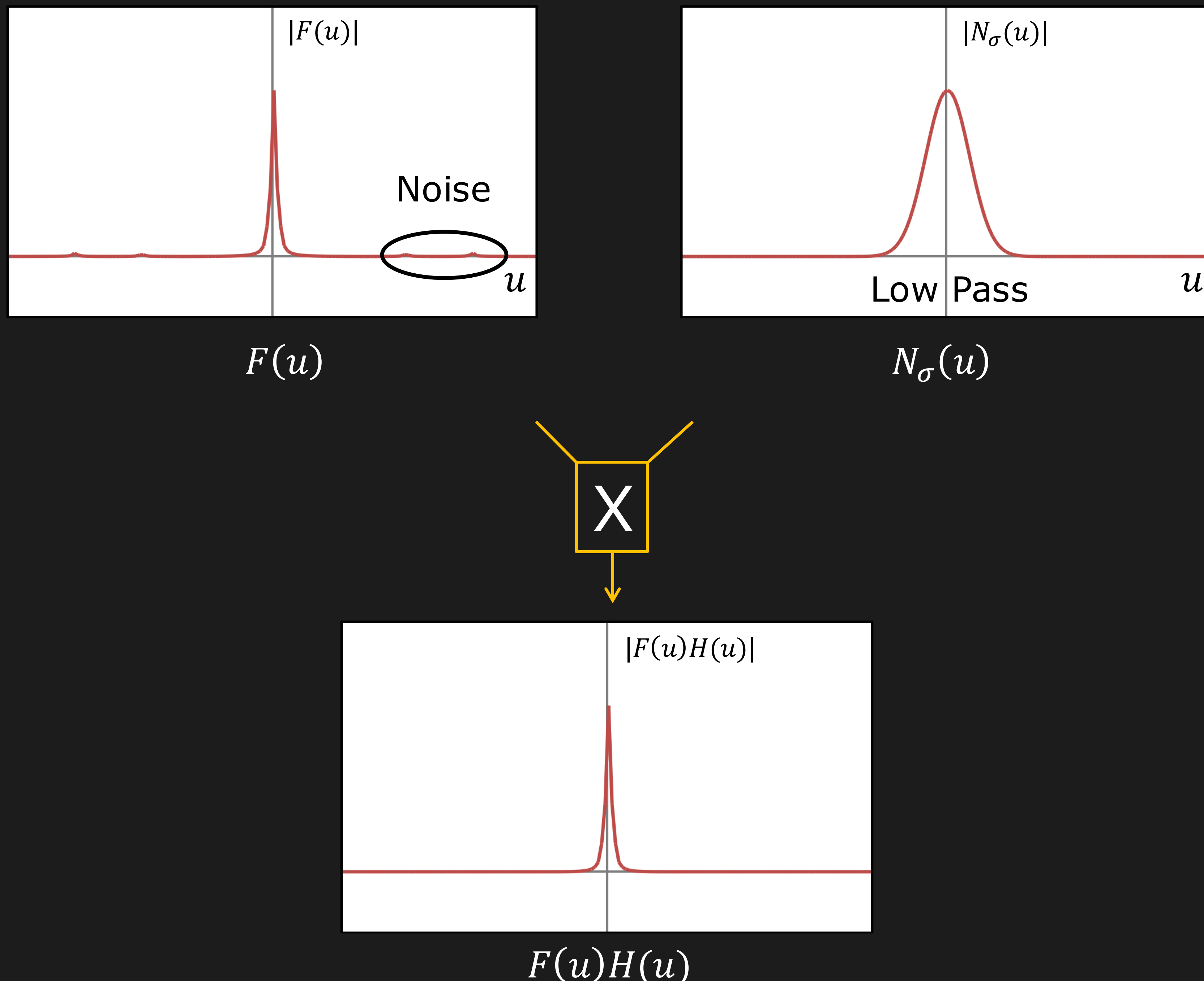
Gaussian Kernel $n_\sigma(x)$

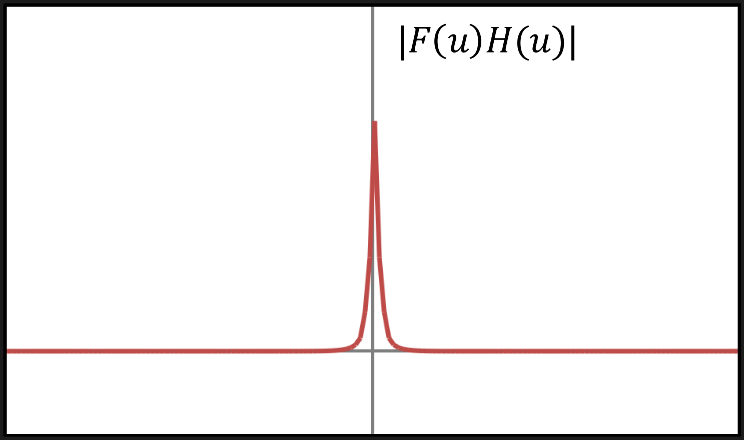
Convolve the Noisy Signal with a Gaussian Kernel


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
$N_\sigma(u)$

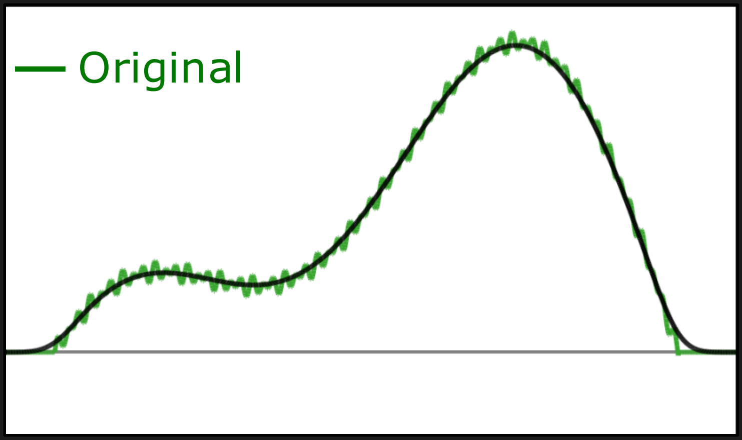


x
Noisy Signal $f(x)$
*
x
Gaussian Kernel $n_\sigma(x)$
FT
FT
$|F(u)|$
Noise
u
$F(u)$
$|N_\sigma(u)|$
Low
Pass
u
$N_\sigma(u)$


|F(u)|
Noise
u
F(u)
|N_σ(u)|
Low Pass
u
N_σ(u)
X
|F(u)H(u)|
F(u)H(u)



$F(u)H(u)$



Gaussian Blurred Signal $g(x)$

Finding FT and IFT

Fourier Transform:

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

x: space

u: frequency

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

$i = \sqrt{-1}$

Inverse Fourier Transform:

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

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 frequencies along x and y, respectively

Inverse Fourier Transform:

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

2D Fourier Transform: Discrete Images

Discrete Fourier Transform (DFT):

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

$p = 0 \ldots M-1$

$q = 0 \ldots N-1$

p and q are frequencies along m and n, respectively

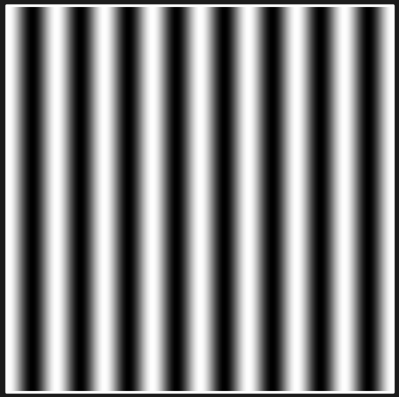
Inverse Discrete Fourier Transform (IDFT):

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

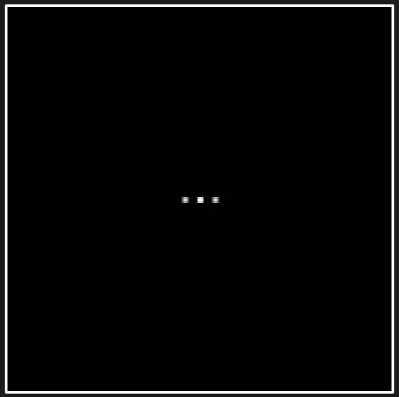
$m = 0 \ldots M-1$

$n = 0 \ldots N-1$

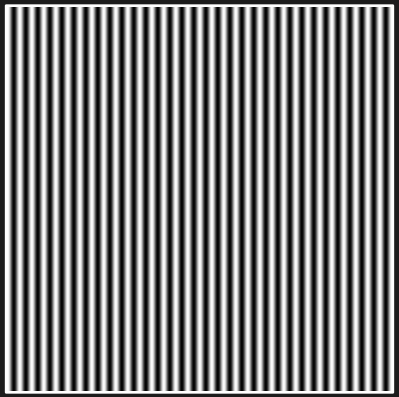
2D Fourier Transform: Example 1



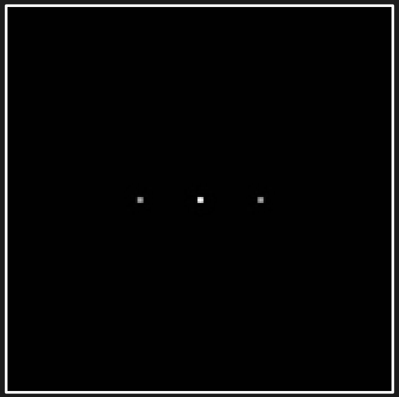
$f(m,n)$



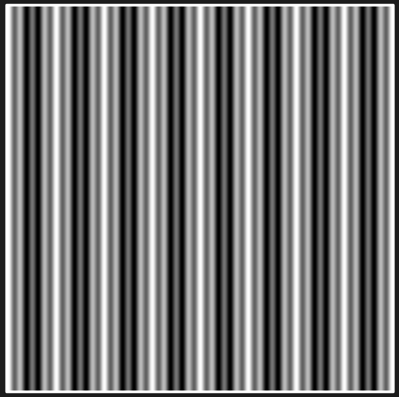
$\log(|F(p,q)|)$



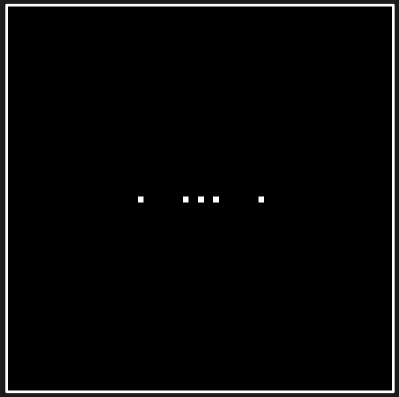
$g(m,n)$



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



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

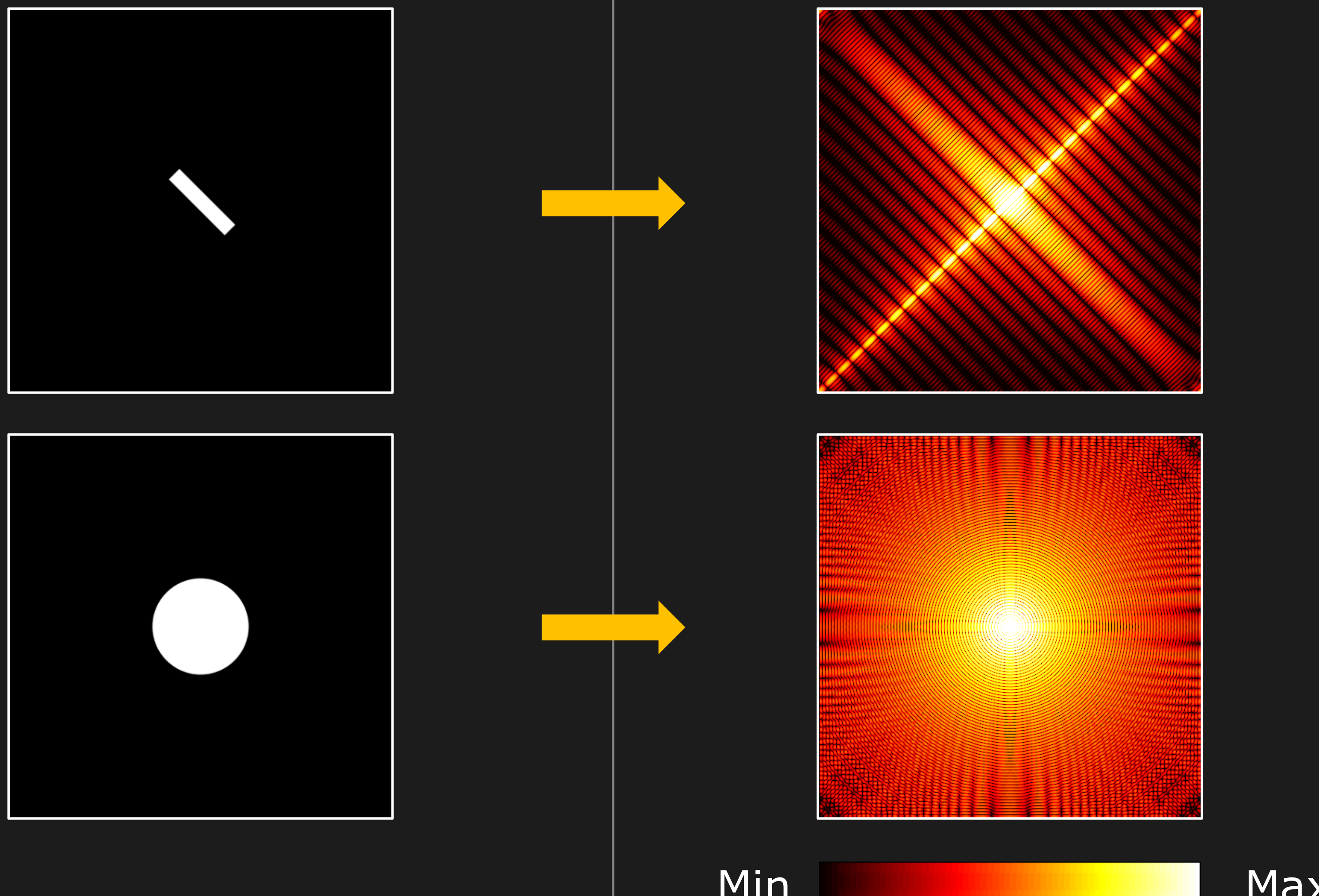


$\log(|F(p,q) + G(p,q)|)$

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

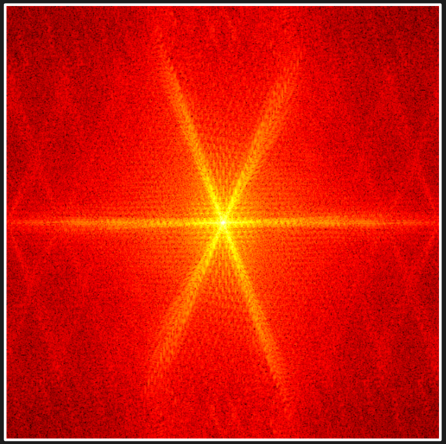
2D Fourier Transform: Example 2

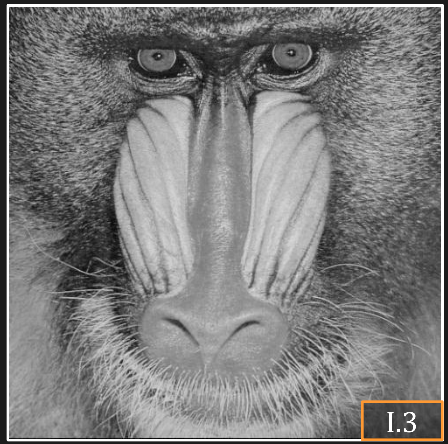
$f(m,n)$ $\log(|F(p,q)|)$

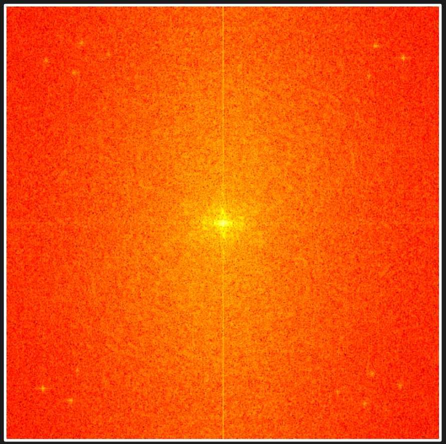


2D Fourier Transform: Example 3

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





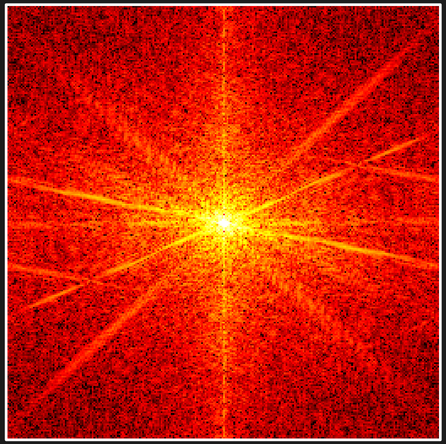


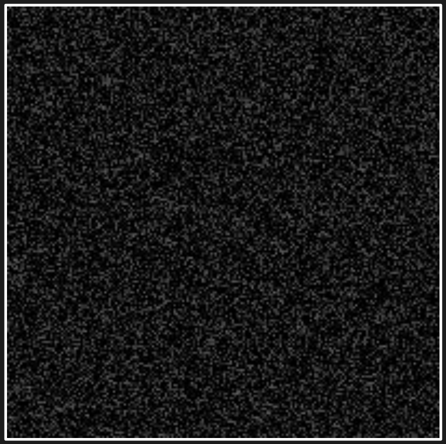
Min  Max

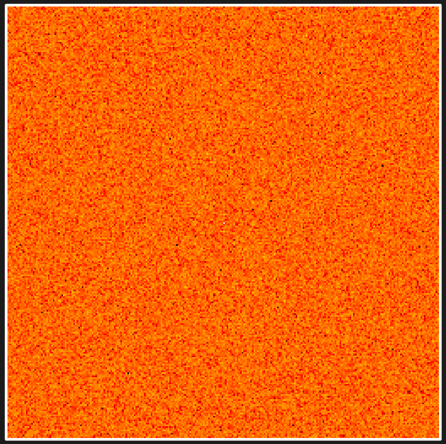
2D Fourier Transform: Example 4

$f(m,n)$

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



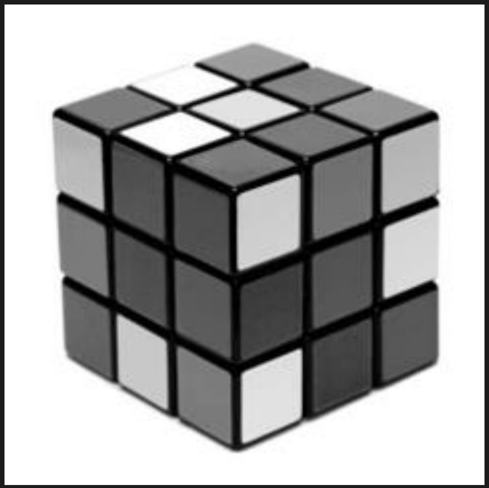




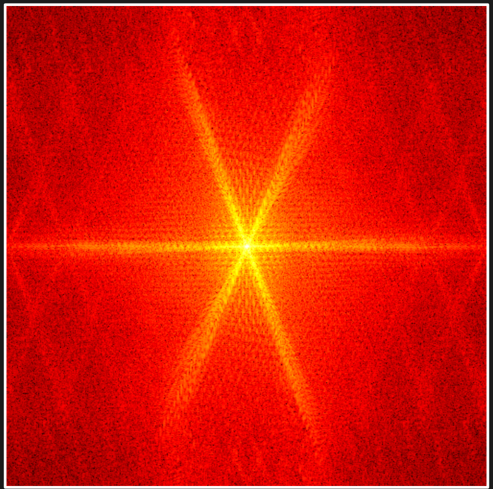
Min  Max

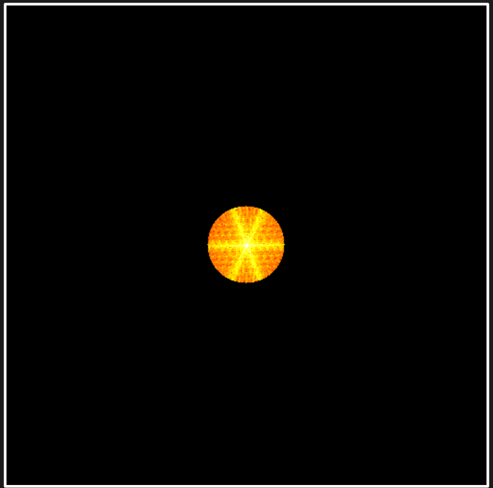
Low Pass Filtering

$f(m,n)$



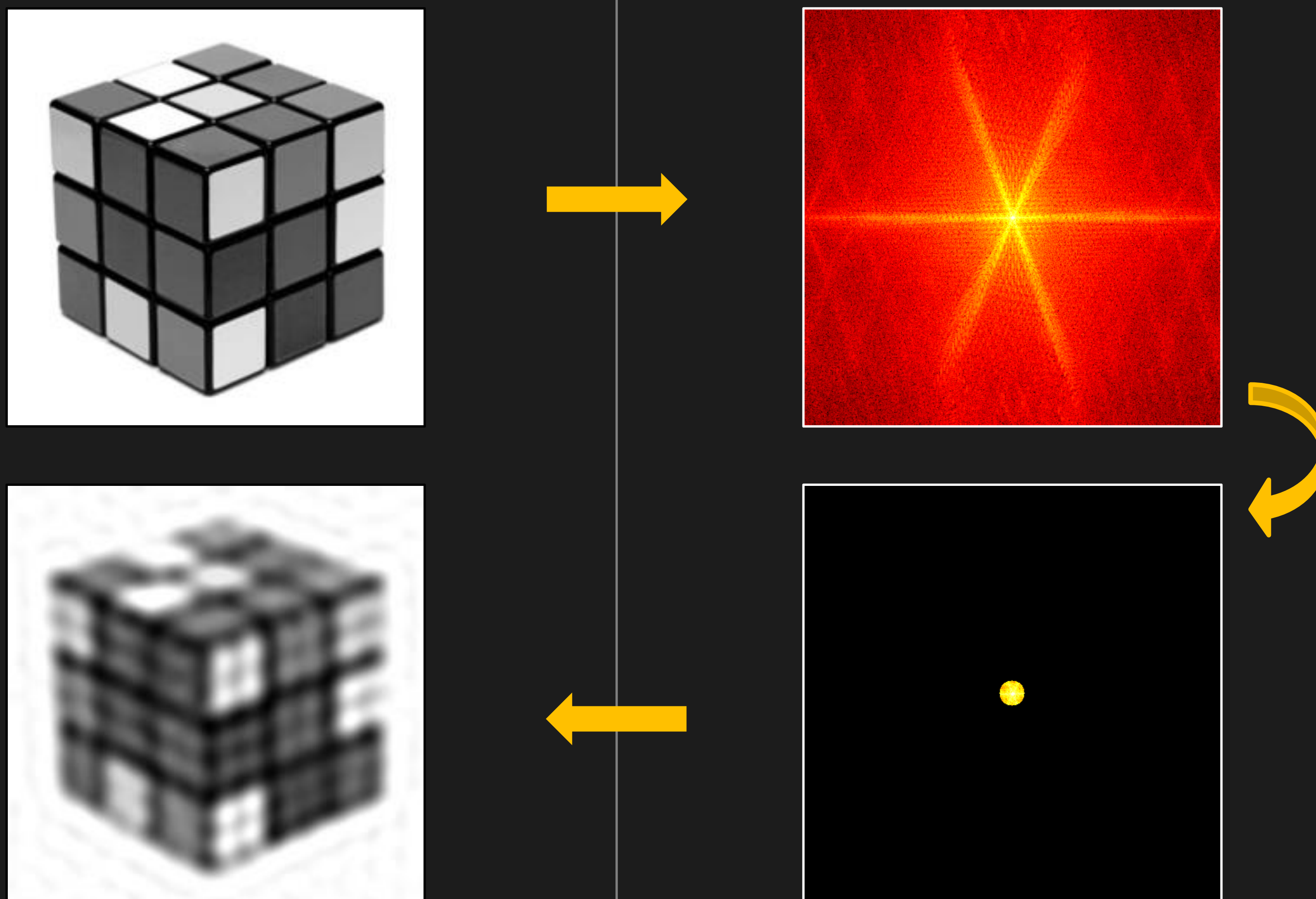
$\log(|F(p,q)|)$





Low Pass Filtering

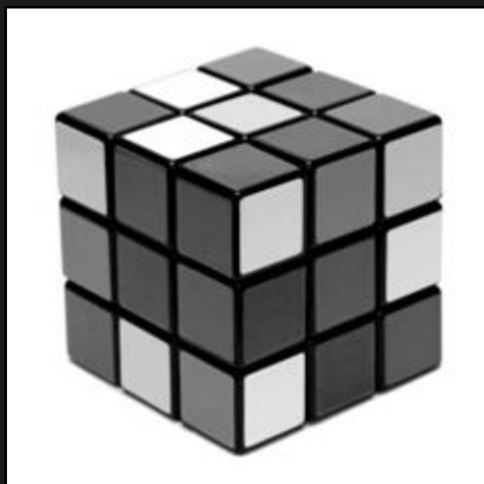
$f(m,n)$ $\log(|F(p,q)|)$

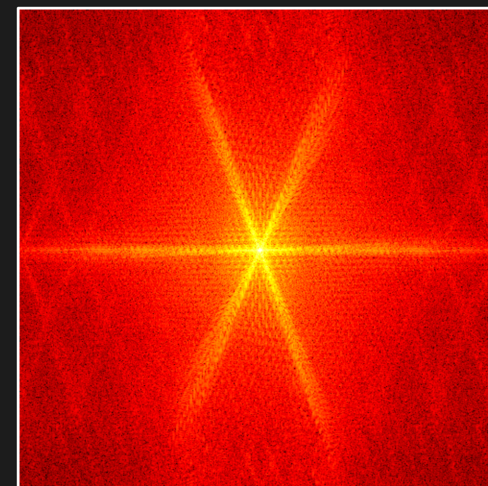


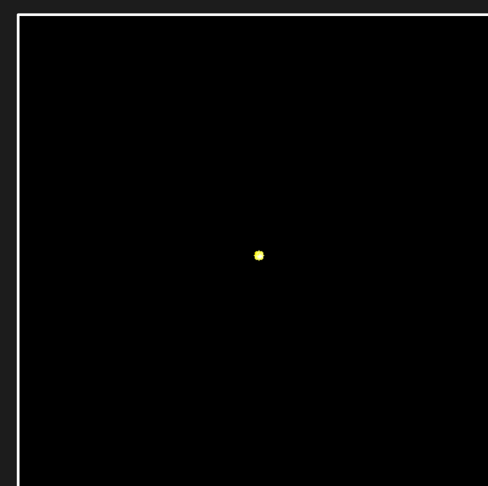
Low Pass Filtering

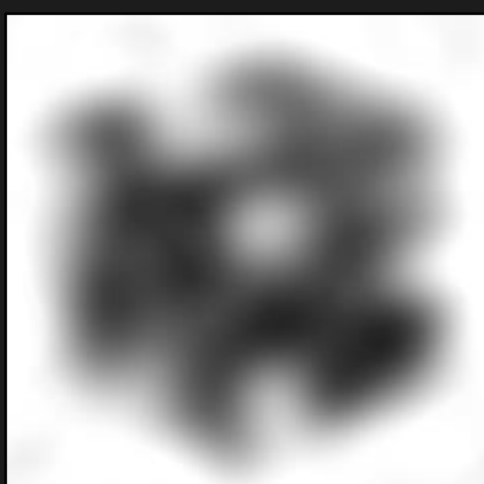
$f(m,n)$

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

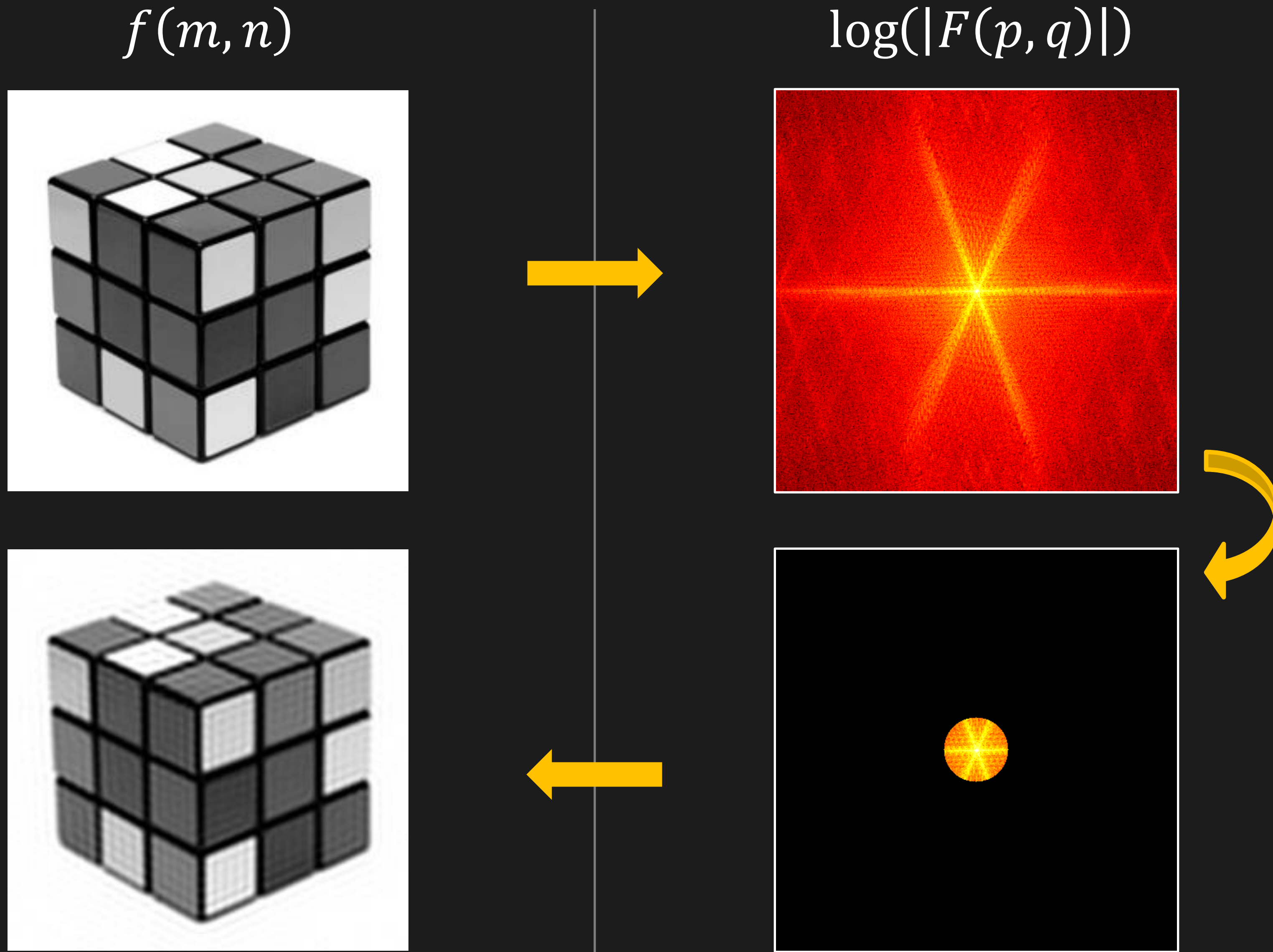




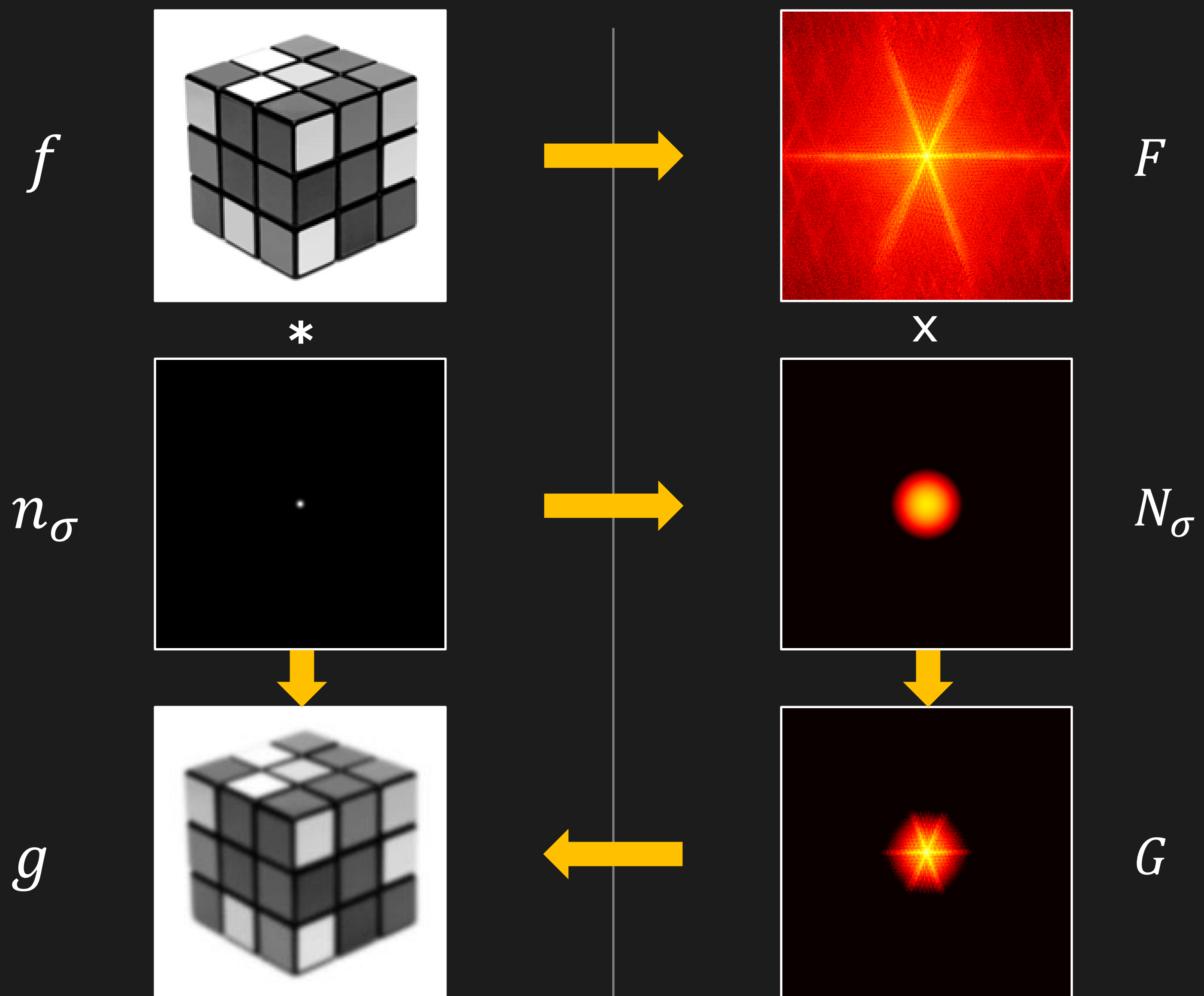




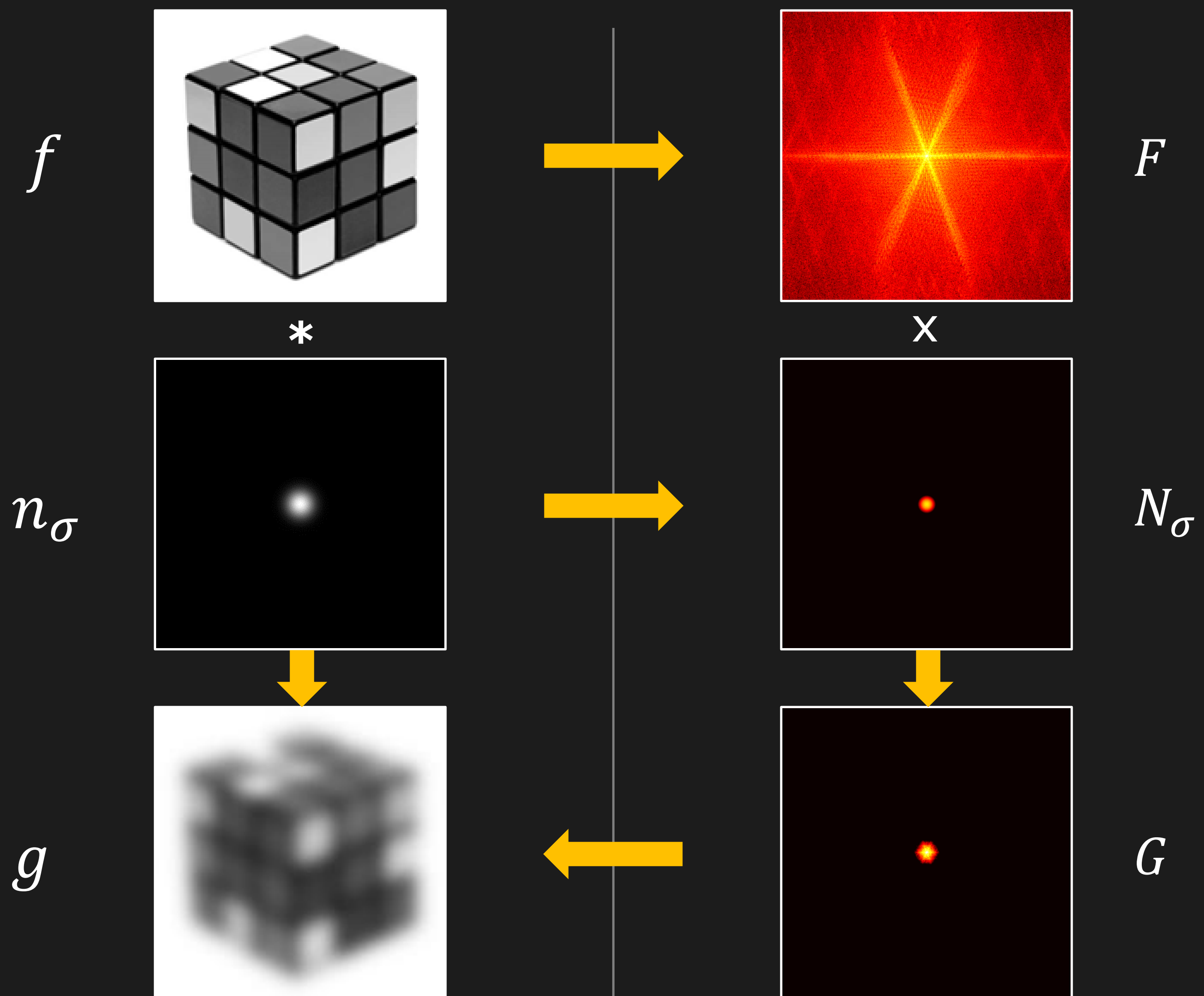
Low Pass Filtering



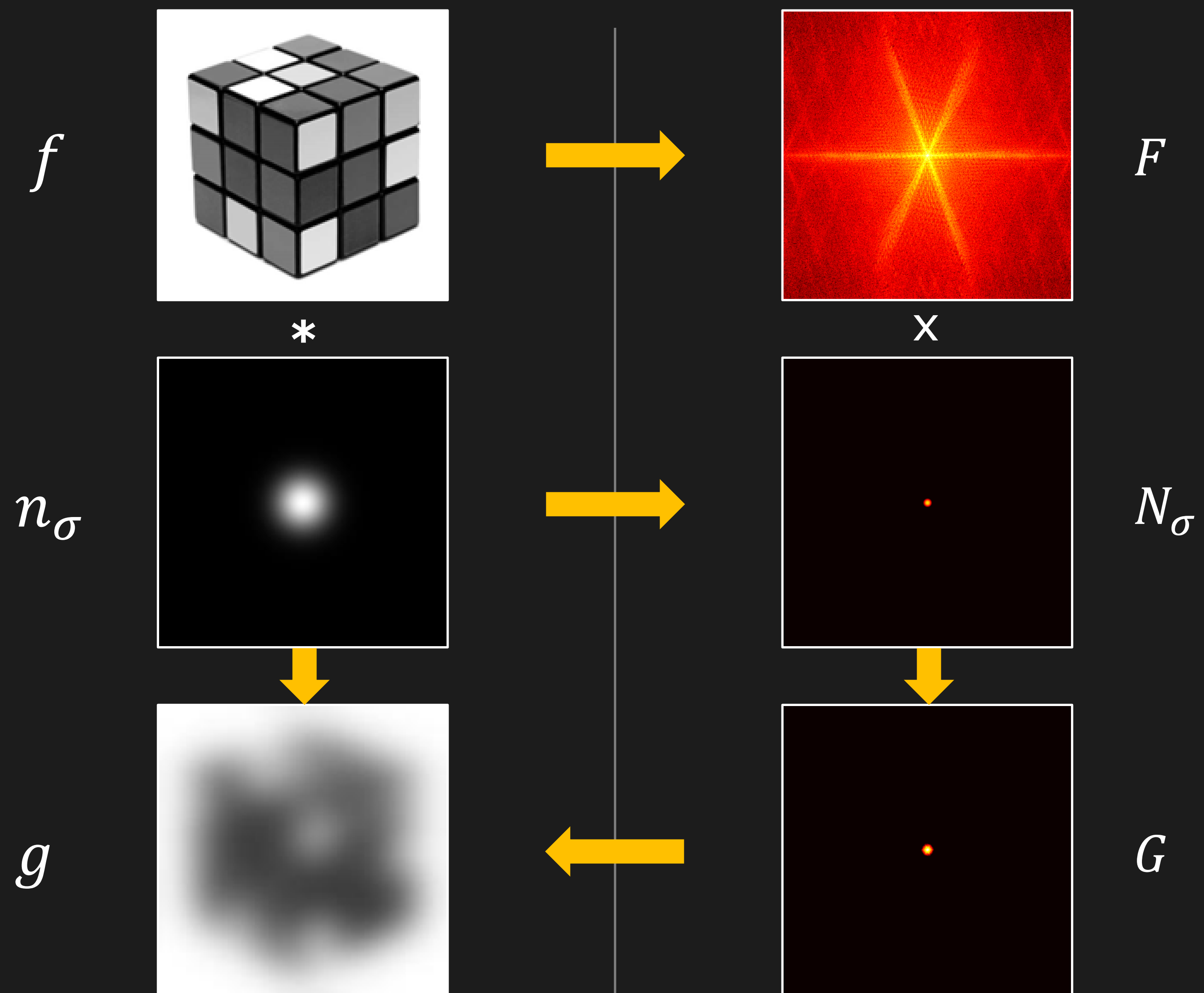
Gaussian Smoothing



Gaussian Smoothing



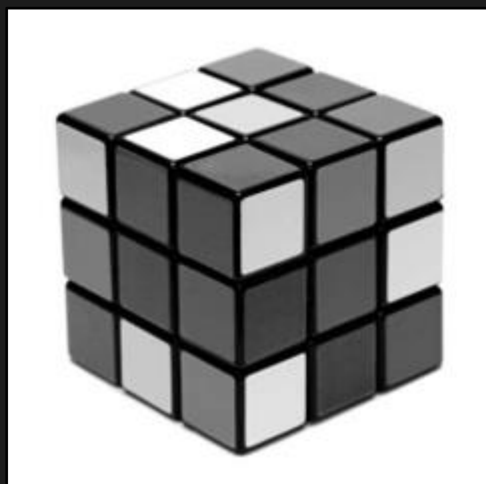
Gaussian Smoothing

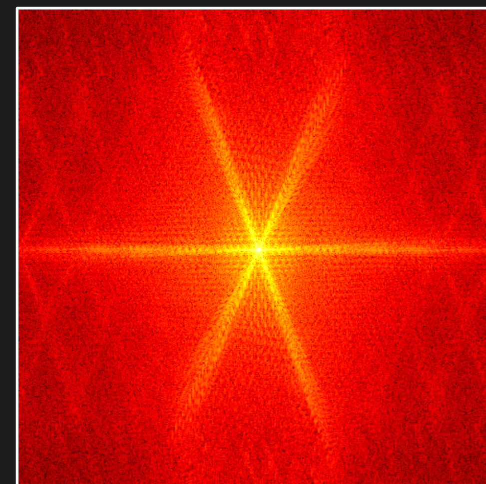


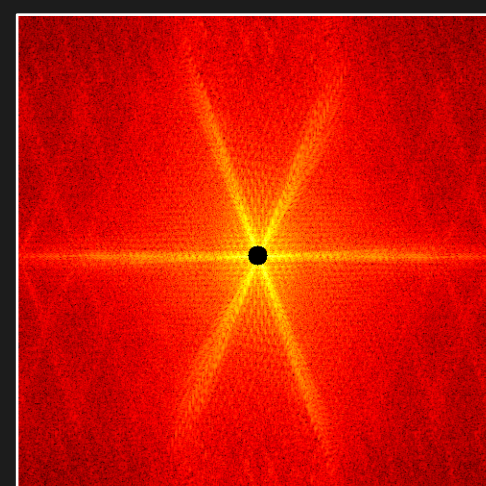
High Pass Filtering

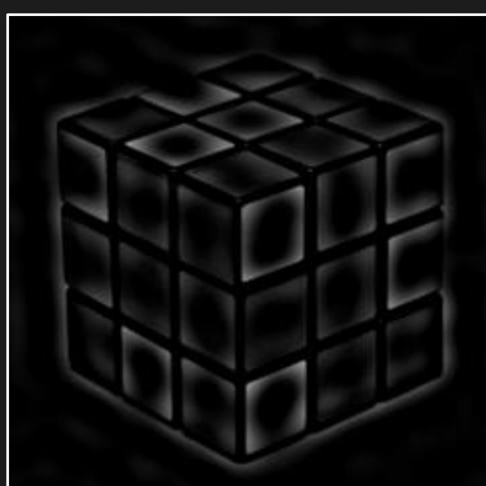
$f(m,n)$

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



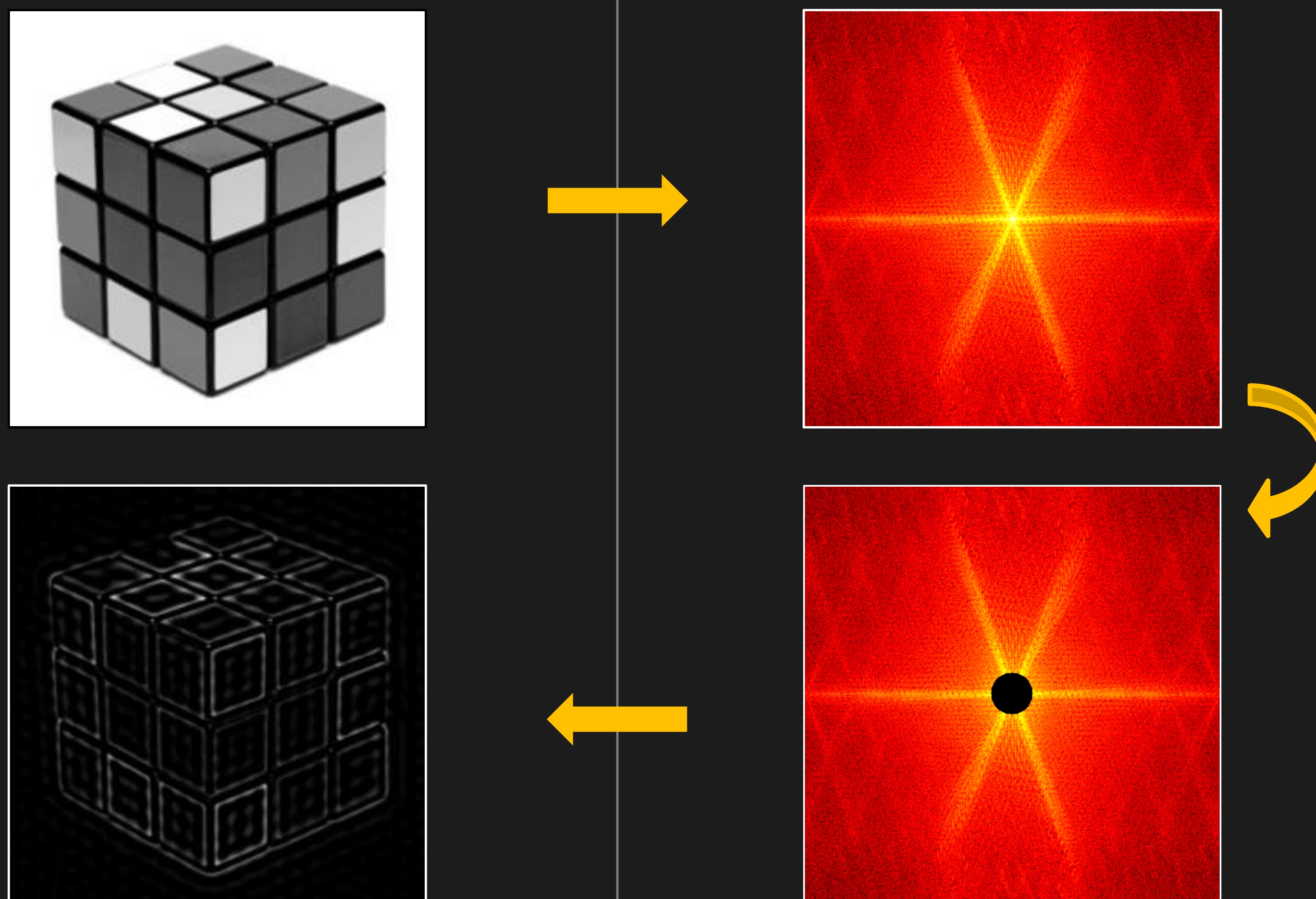






High Pass Filtering

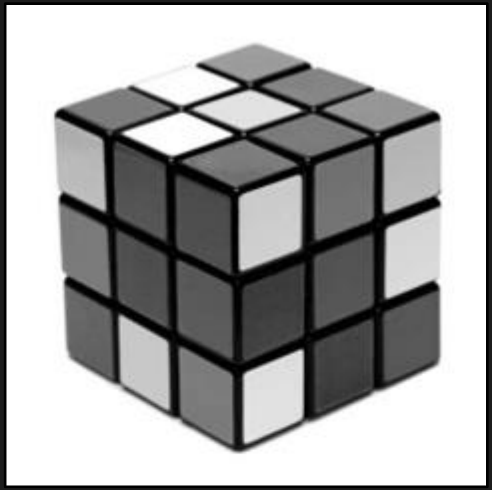
$f(m,n)$ $\log(|F(p,q)|)$

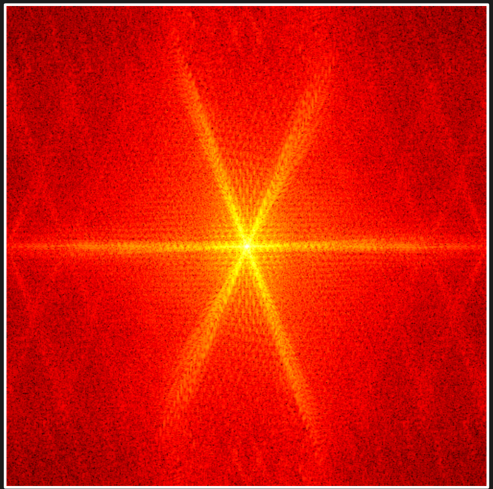


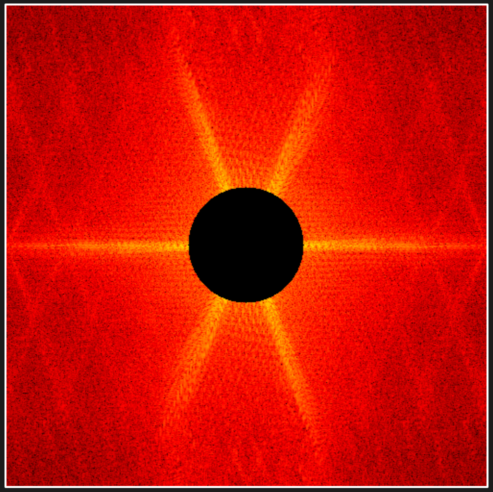
High Pass Filtering

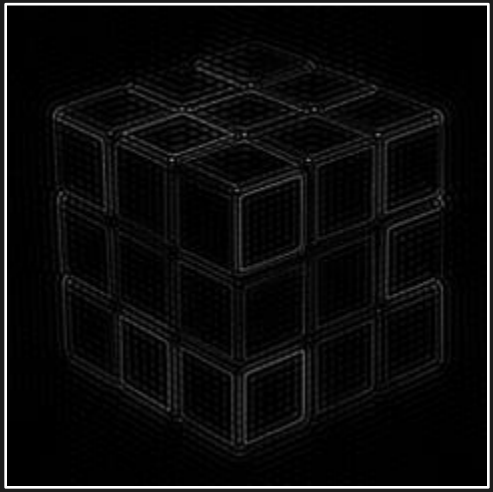
$f(m,n)$

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





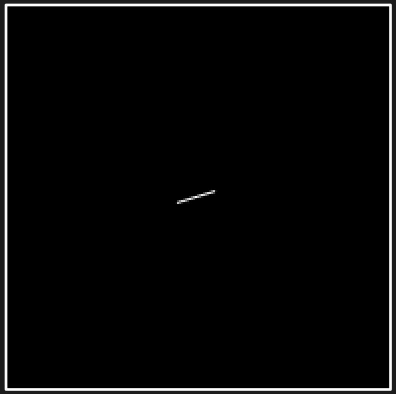




Motion Blur

Scene $f(x, y)$

$*$



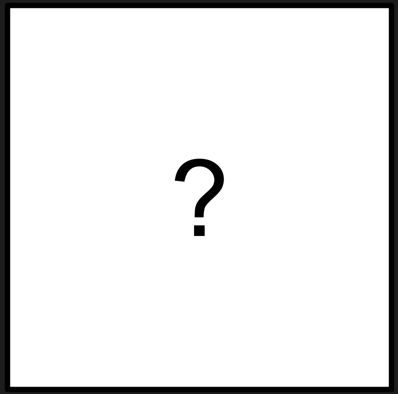
PSF $h(x, y)$
(Camera Shake)

$=$

Image $g(x, y)$

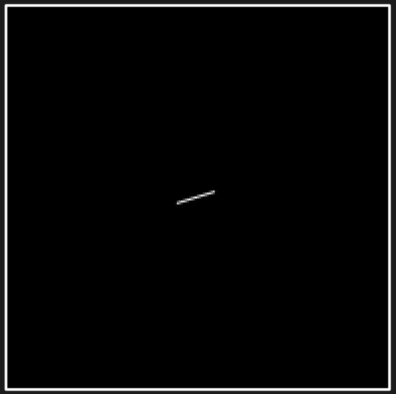
$$\boxed{f(x, y) * h(x, y) = g(x, y)}$$

Motion Blur



Scene $f(x,y)$

$*$



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

$=$

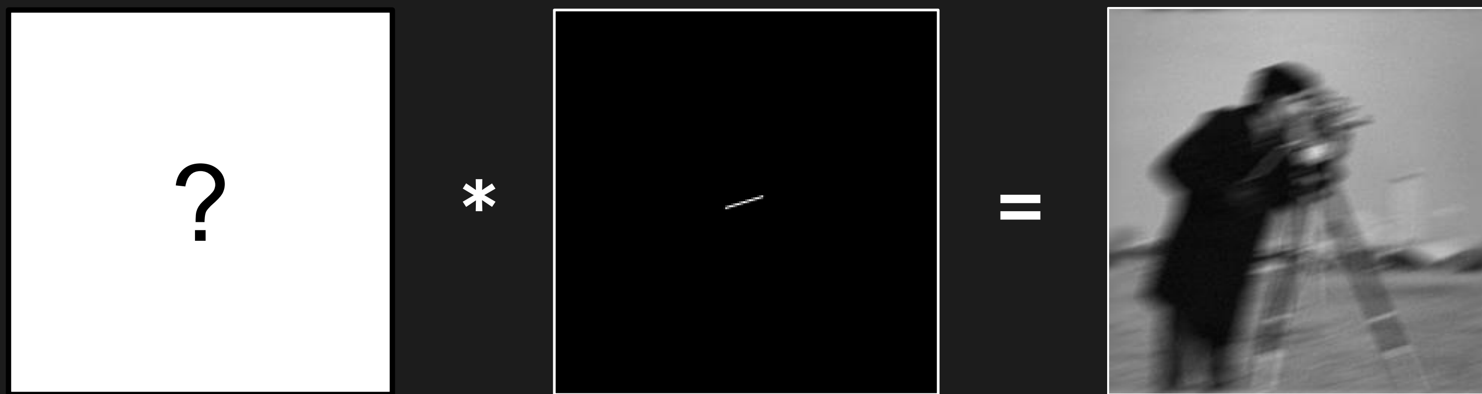
Image $g(x,y)$

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

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.

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

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

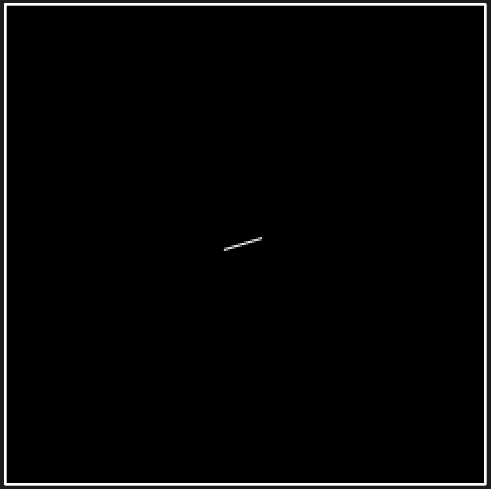
$$F'(u, v) = \frac{G(u, v)}{H(u, v)} \longrightarrow \boxed{\text{IFT}} \longrightarrow f'(x, y)$$

Motion Deblur: Deconvolution

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

Image $g(x,y)$

deconvolve



PSF $h(x,y)$

=

?

Recovered $f'(x,y)$

Motion Deblur: Deconvolution

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

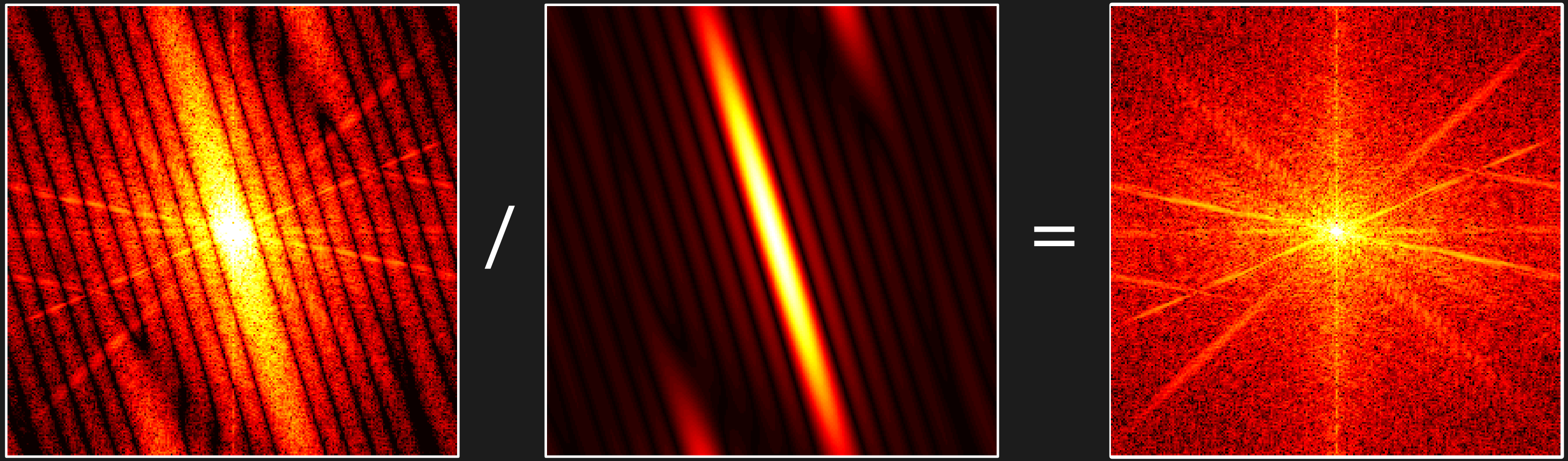


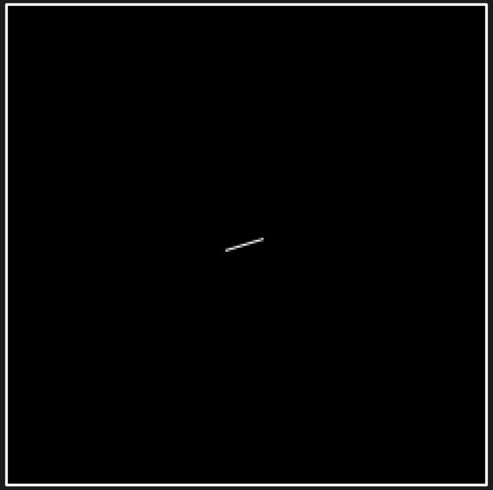
Image $G(u,v)$ PSF $H(u,v)$ Recovered $F'(u,v)$

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

Motion Deblur: Deconvolution

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

deconvolve

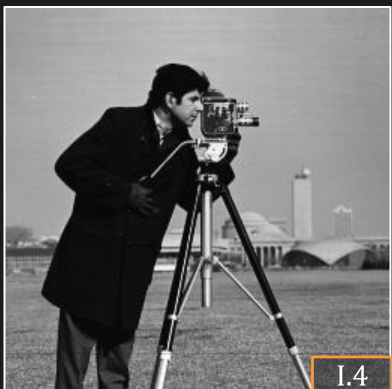


=

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

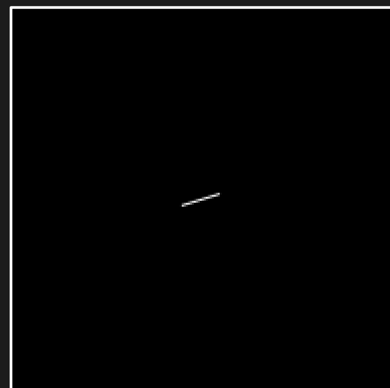
Step 2: Compute IFT of $F'(u,v)$ to recover scene

Adding Noise to the Problem



Scene $f(x,y)$

$*$



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

$+$

Noise $\eta(x,y)$

$=$

Image $g(x,y)$

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

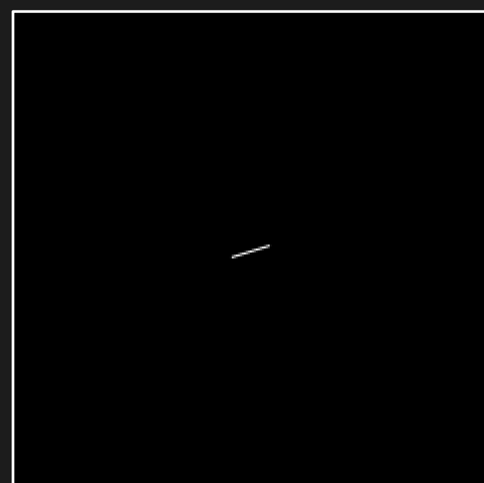
Can we afford to ignore noise?

Motion Deblur: Deconvolution

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

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

deconvolve



=

?

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

PSF $h(x, y)$

Recovered $f'(x, y)$

Motion Deblur: Deconvolution

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

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

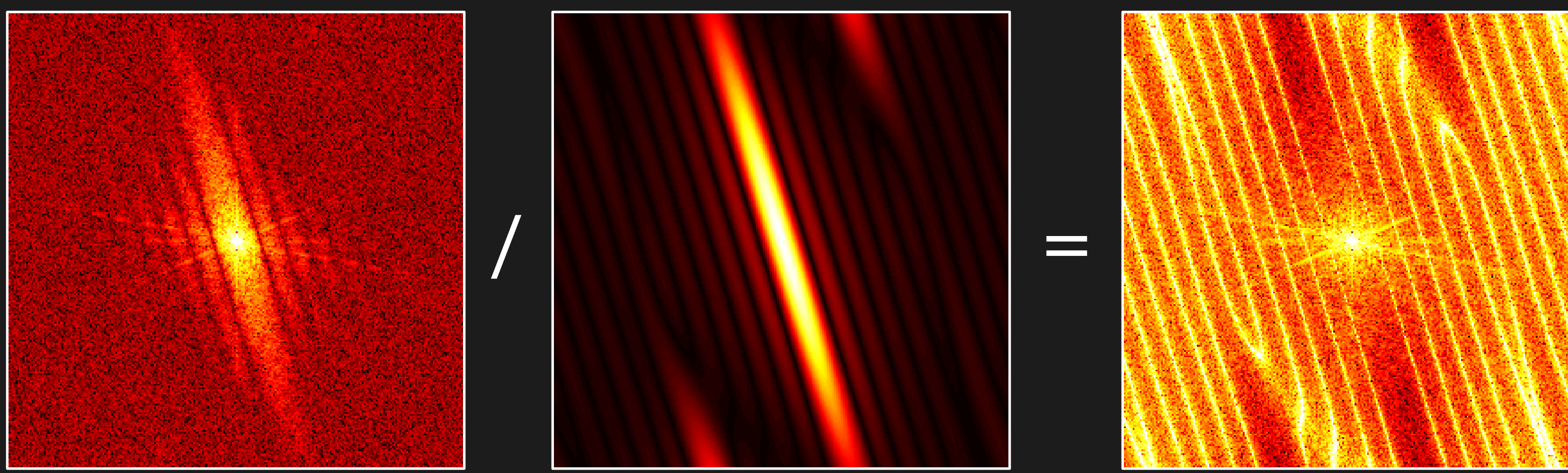


Image $G(u,v)$ PSF $H(u,v)$ Recovered $F'(u,v)$

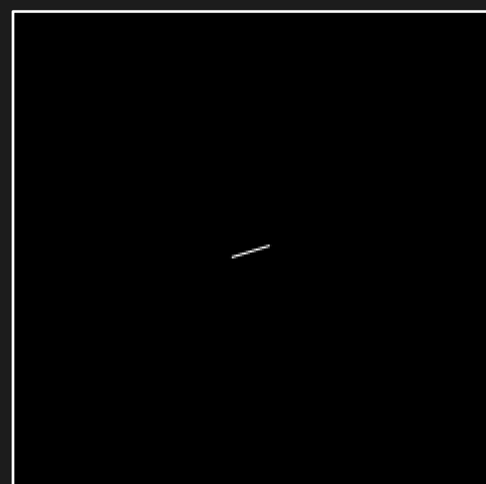
Higher frequencies in $F'(u,v)$ are amplified

Motion Deblur: Deconvolution

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

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

deconvolve



=

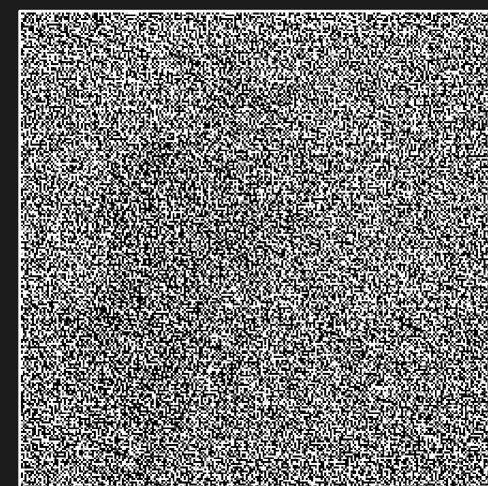


Image $g(x, y)$
(with noise)

PSF $h(x, y)$

Recovered $f'(x, y)$

Noise is significantly amplified

Deconvolution: Issues

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

1. Where $H(u,v) = 0$, $F'(u,v) = \infty \rightarrow$ Not recoverable

2. Motion blur filter $H(u,v)$ is a low pass filter.

   For high frequencies (u,v):
   - Noise $N(u,v)$ in $G(u,v)$ is high
   - Filter $H(u,v) \approx 0$

   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:

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

Noise-to-Signal Ratio, $NSR(u,v)$

$$NSR(u,v) = \frac{\text{Power of Noise at } (u,v)}{\text{Power of Signal (Scene) at } (u,v)} = \frac{|N(u,v)|^2}{|F(u,v)|^2}$$

Noise Suppression: Weiner Deconvolution

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

- Determining NSR requires us to have prior knowledge of the noise “pattern” and the scene (or of a similar scene).

$$NSR(u,v) = \frac{|N(u,v)|^2}{|F(u,v)|^2}$$

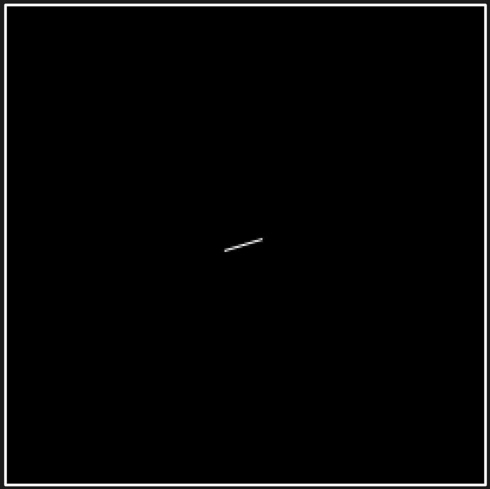
- Often NSR is set to a single suitable constant λ.

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

Noise Suppression: Weiner Deconvolution

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

/



PSF $h(x, y)$

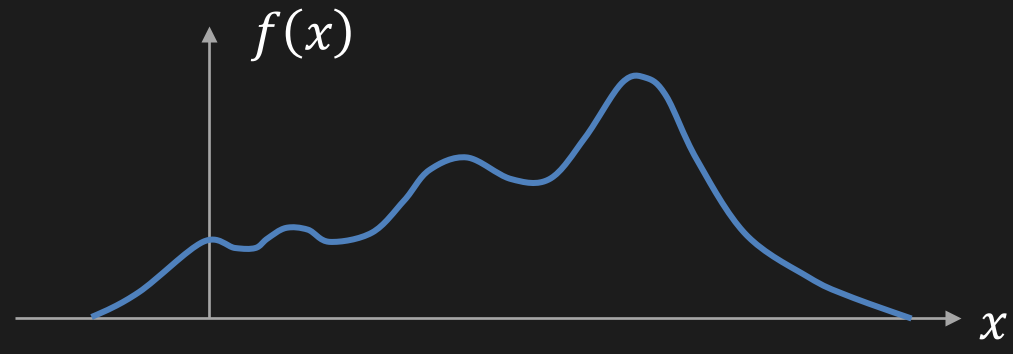
=

Recovered $f'(x, y)$

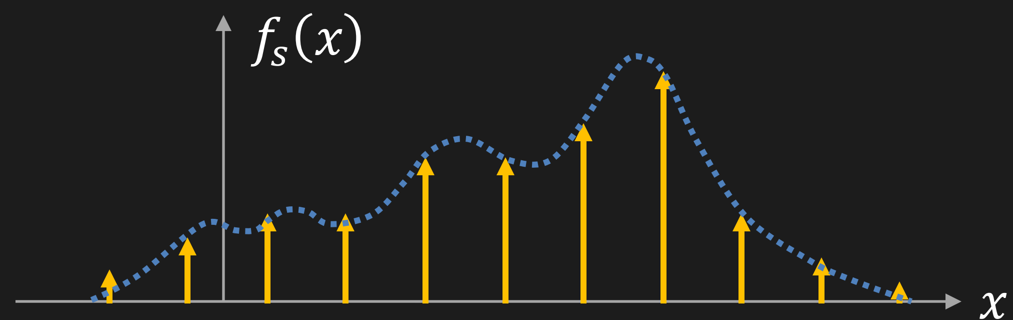
$NSR(u, v) = \lambda = 0.002$ was used to recover image

From Continuous to Digital Image

Continuous Signal:

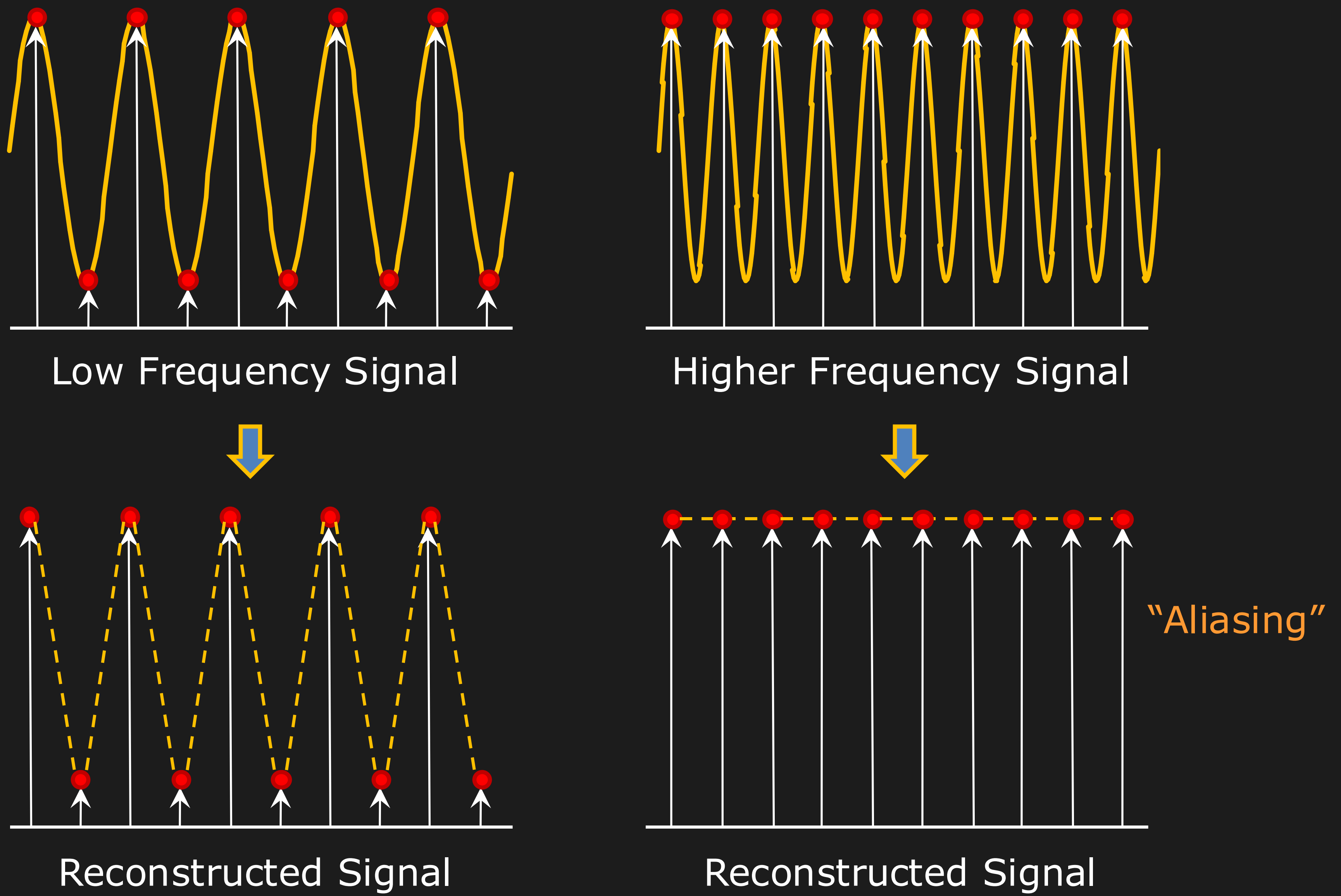


Digital Signal:



How “dense” should the samples be?

Sampling Problem



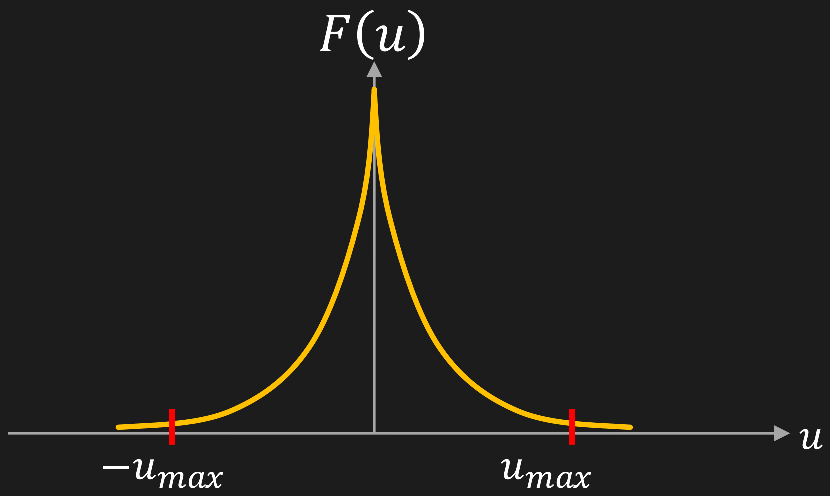
Sampling Problem

"Well sampled" image

"Under sampled" image
(visible aliasing artifacts)

Aliasing in Digital Imaging

Aliasing occurs when imaging a scene (signal) that has frequencies above the image sensor's Nyquist Frequency



Typical Power Spectrum of Natural Scenes

Aliasing artifacts usually occur in the form of Moiré patterns

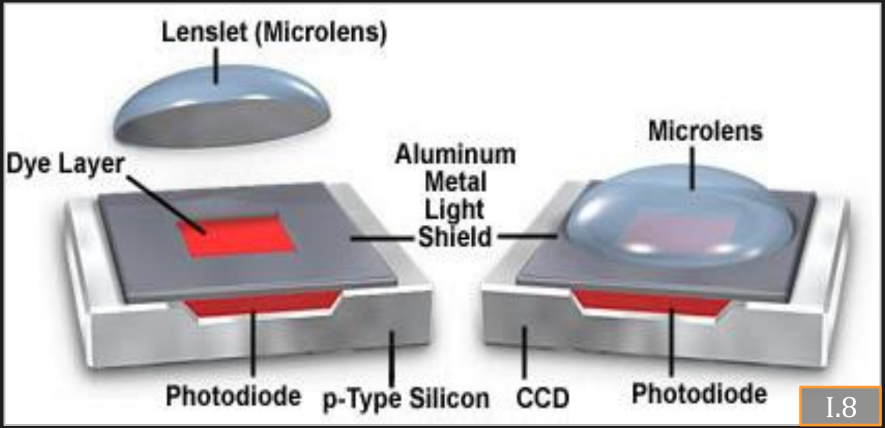
How do sensors combat aliasing?

Minimizing the Effects of Aliasing

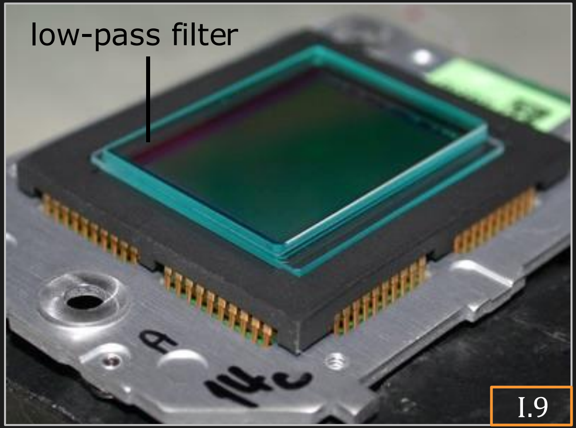
Band Limit: Clip the signal above the Nyquist frequency.

Effectively, "blur" the scene before sampling.

Sensors use two strategies.



Pixels are area-samplers (box-averaging filter)



Use optical low-pass filter (anti-aliasing filter)

References: Textbooks

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

Computer Vision: Algorithms and Applications (Chapter 3)
Szeliski, R., Springer

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

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

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

I.5 http://en.wikipedia.org/wiki/File:Moire_pattern_of_bricks.jpg

I.6 http://www.todayandtomorrow.net/wp-content/uploads/2010/06/shirt_video.jpg

I.7 http://www.svi.nl/wikiimg/StFargeaux_kasteel_buiten1_aliased.jpg

I.8 http://learn.hamamatsu.com/articles/images/lenslet.jpg

I.9 http://www.astrosurf.com/luxorion/Physique/nikon-d200-low-pass-ir.jpg