Image Processing II

Computer Vision: CS 600.461/661

Image Processing I

Transform image to new one that is easier to manipulate.

Topics:

(1) Pixel Processing

(2) Convolution

(3) Linear Filtering

} Minilectures 1 and 2

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) Nonlinear Image Processing

Minilectures 3, 4, and 5

Computer Vision: Algorithms and Appications (Chapter 3.3,3.4)
Szelinski, 2011 (available online)

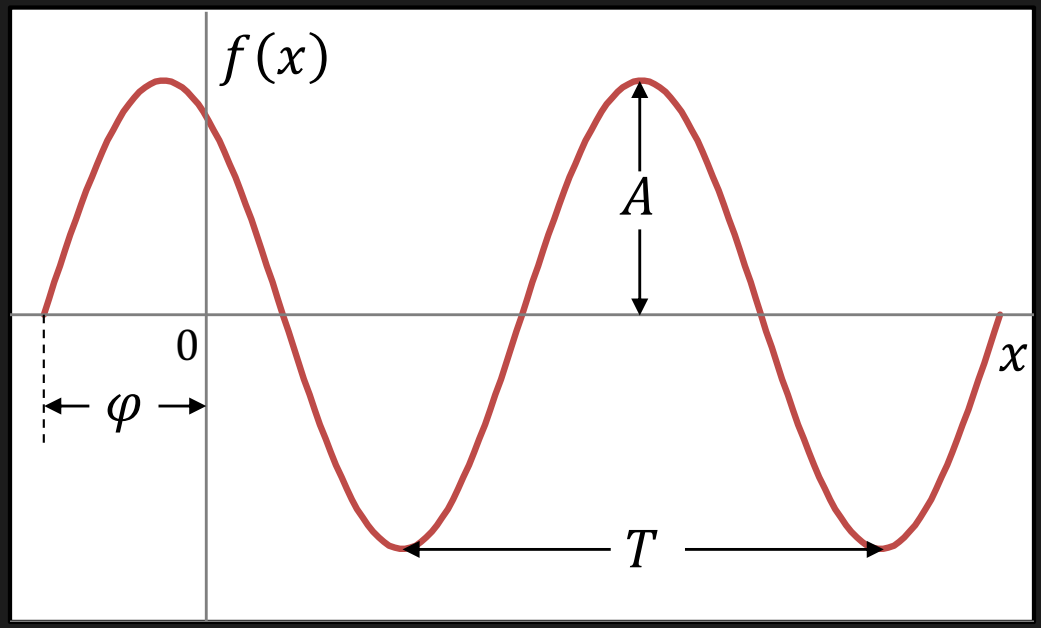
Jean Baptiste Joseph Fourier

(1768-1830)

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

Sinusoid

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



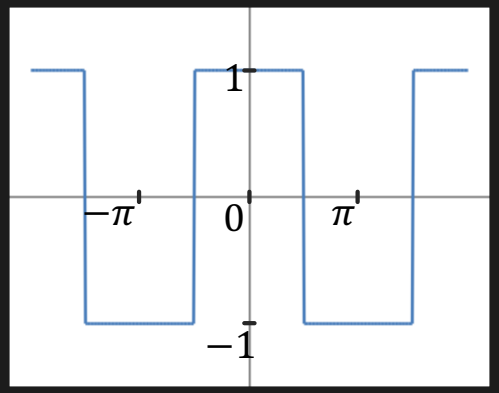
A: Amplitude

T: Period

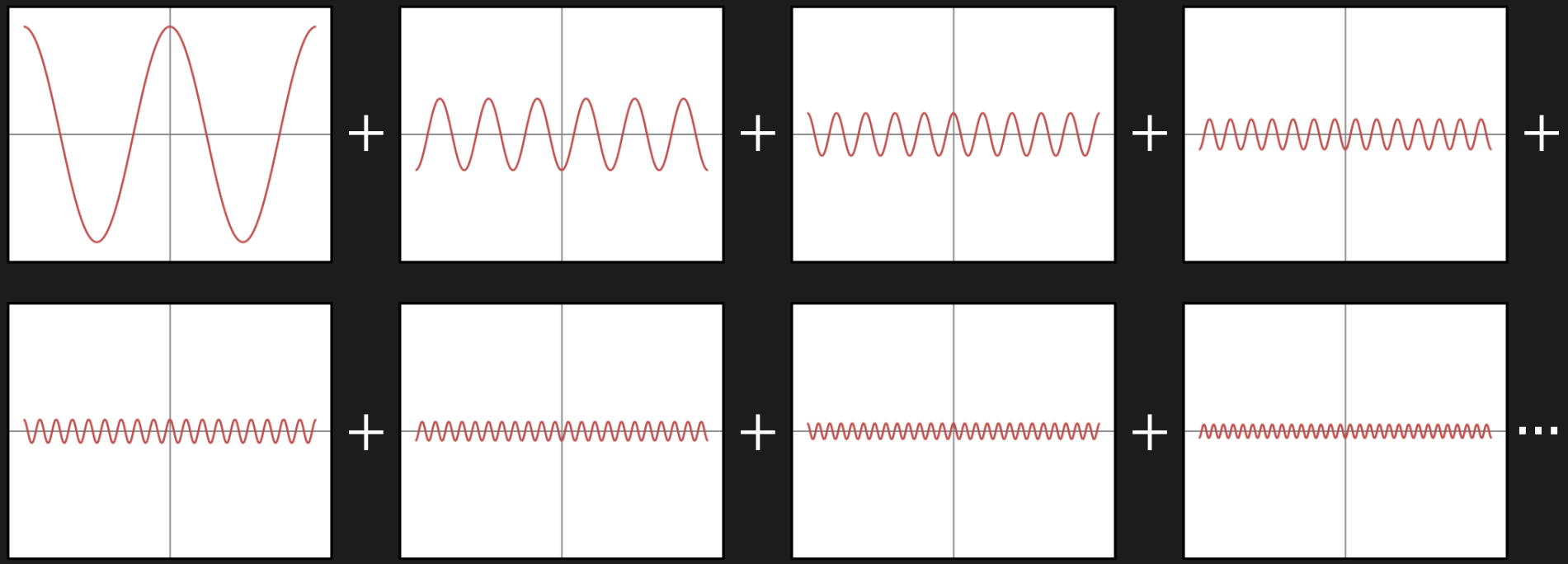
φ: Phase

u: Frequency $(1/T)$

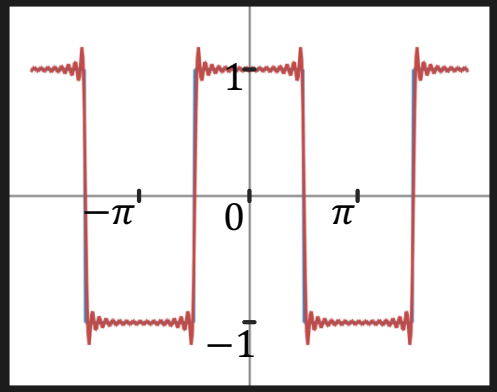
Fourier Series



Square Wave
(Period 2π)

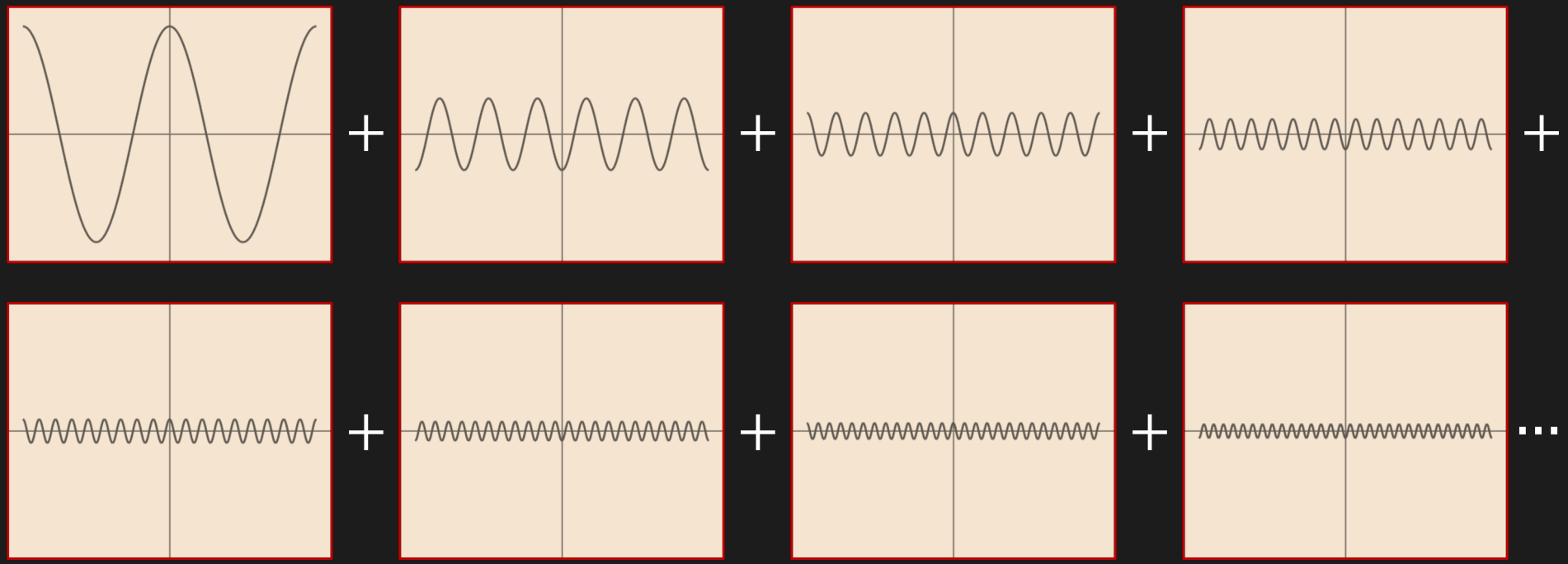


Fourier Series

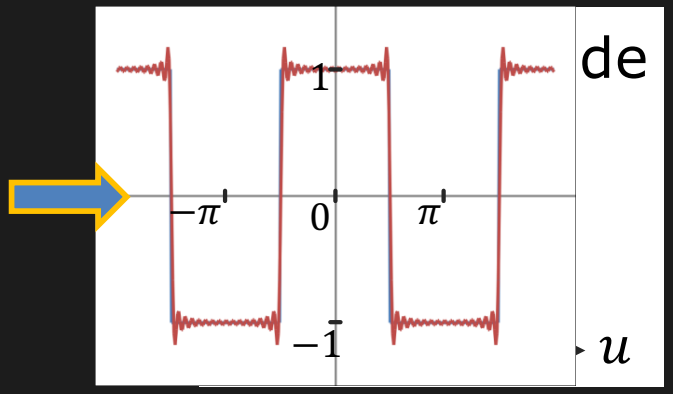


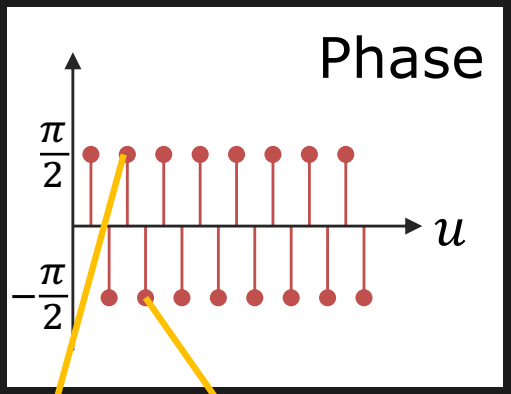
Square Wave (Period 2π)

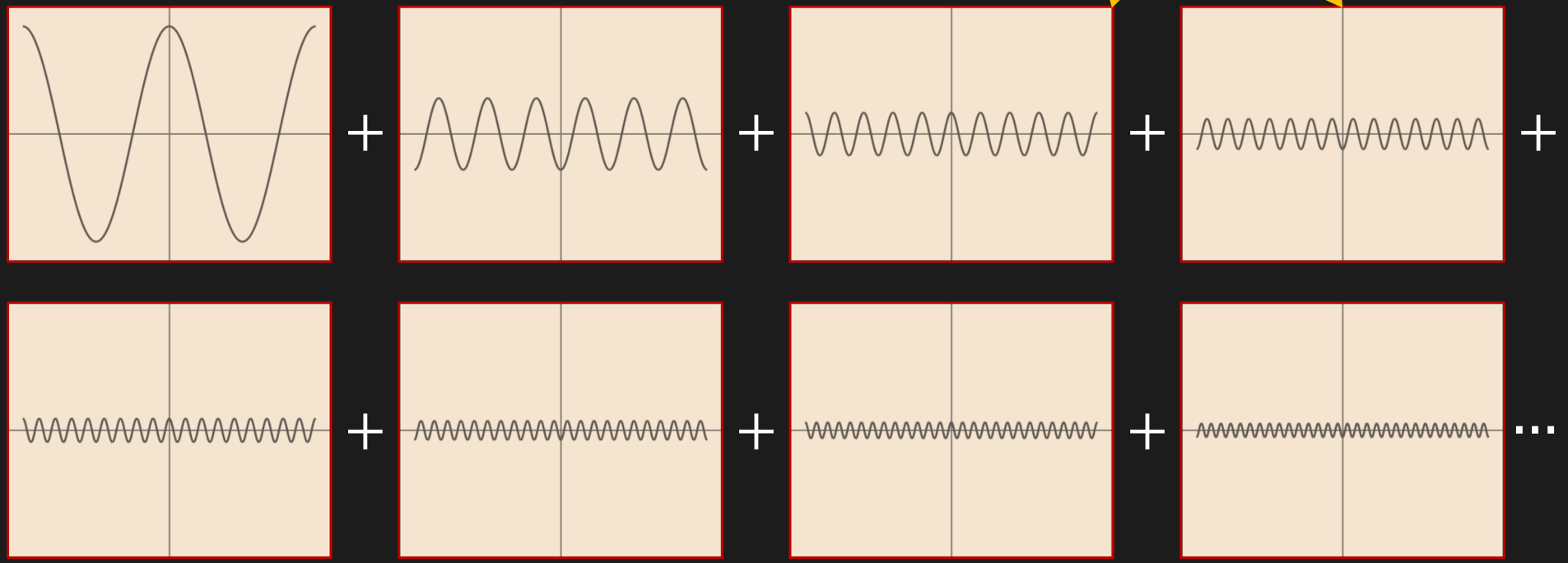
Sum of Sinusoids



An Alternate Representation of Signal







Sinusoid

Orthogonal bases

$$\int_{-\pi}^{\pi} \sin nx \cdot \cos mx \, dx = 0$$

$$\int_{-\pi}^{\pi} \sin nx \cdot \sin mx \, dx = 0 \; (n \neq m)$$

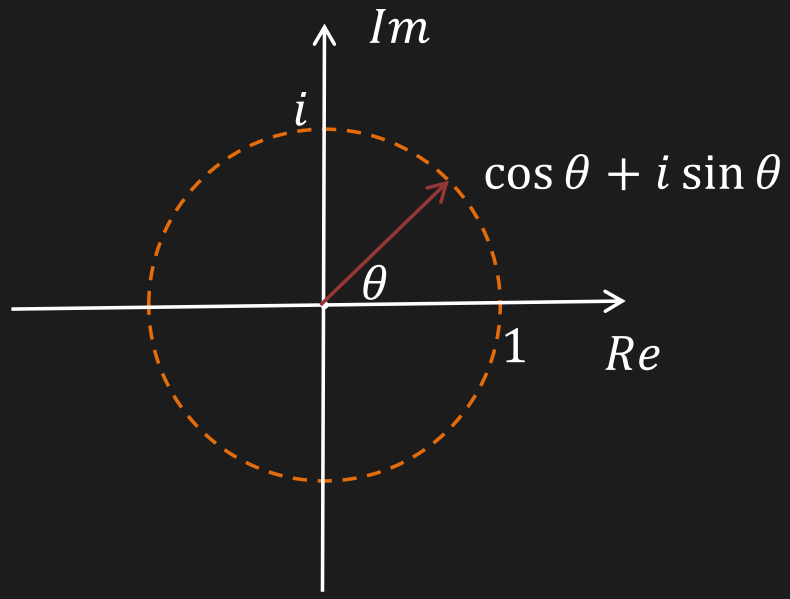
$$\int_{-\pi}^{\pi} \cos nx \cdot \cos mx \, dx = 0 \; (n \neq m)$$

$$\int_{-\pi}^{\pi} \sin nx \cdot \sin mx \, dx = 1 \; (n = m)$$

$$\int_{-\pi}^{\pi} \cos nx \cdot \cos mx \, dx = 1 \; (n = m)$$

Exponential Sinusoid (Euler Formula)

What if the function is not periodic?



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

Finding FT and IFT

Fourier Transform:

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

x: space

u: frequency

Inverse Fourier Transform:

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

Fourier Transform is Complex!

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

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

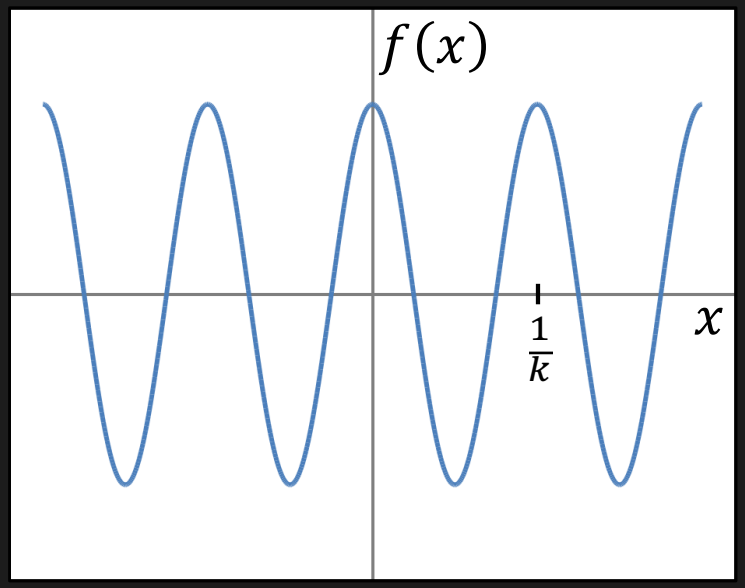
$$F(u) = \Re\mathrm{e}\{F(u)\} + i\,\Im\mathrm{m}\{F(u)\}$$

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

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

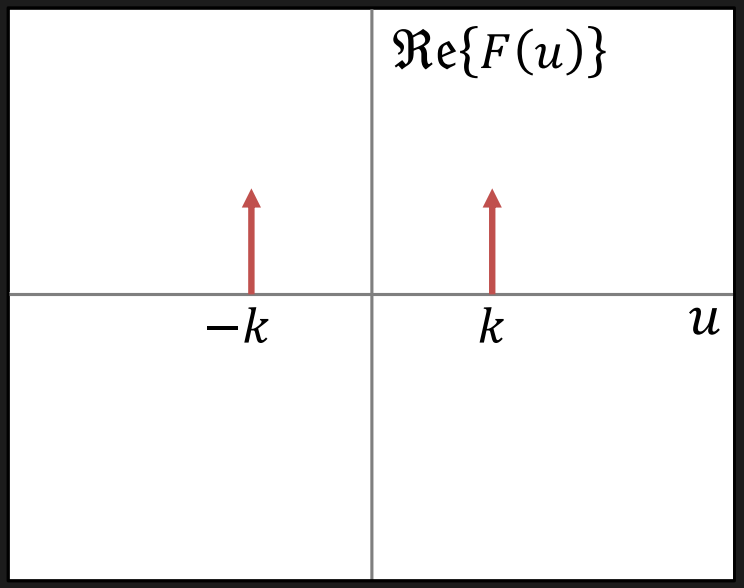
Fourier Transform Examples

Signal $f(x)$



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

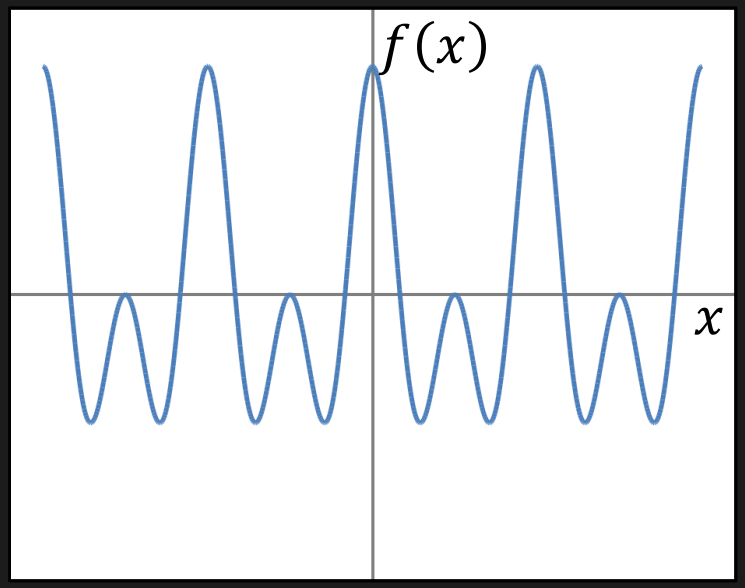
Fourier Transform $F(u)$



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

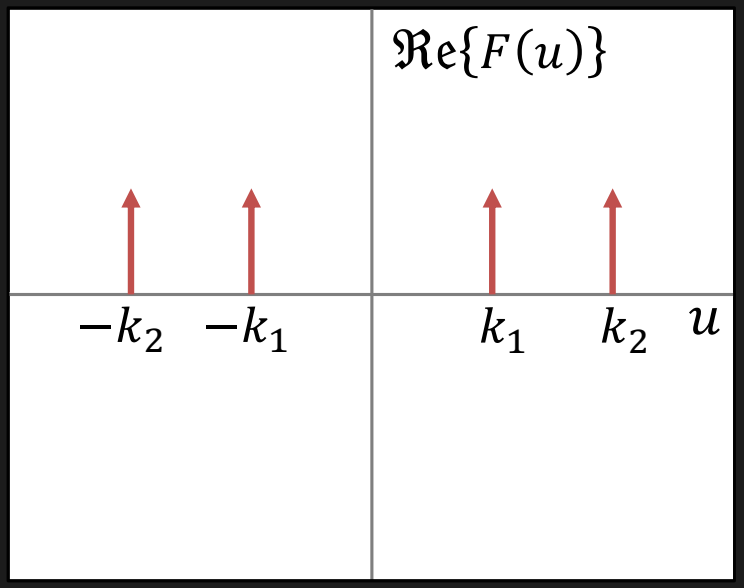
Fourier Transform Examples

Signal $f(x)$



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

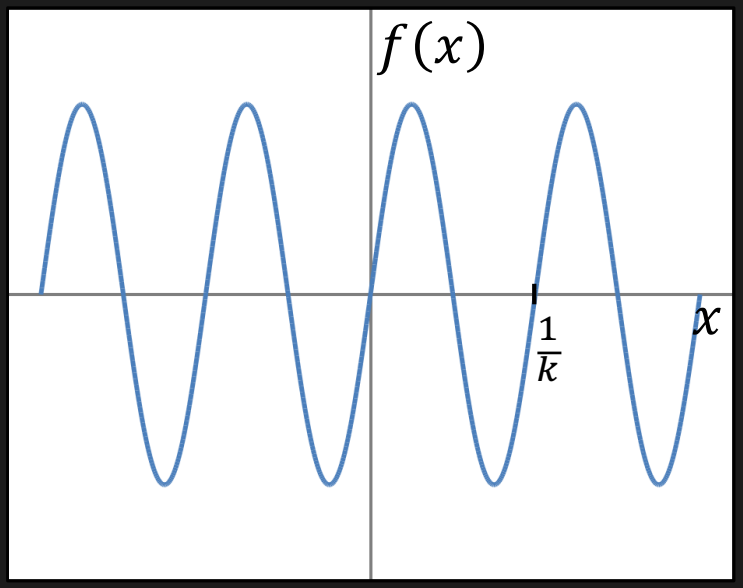
Fourier Transform $F(u)$



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

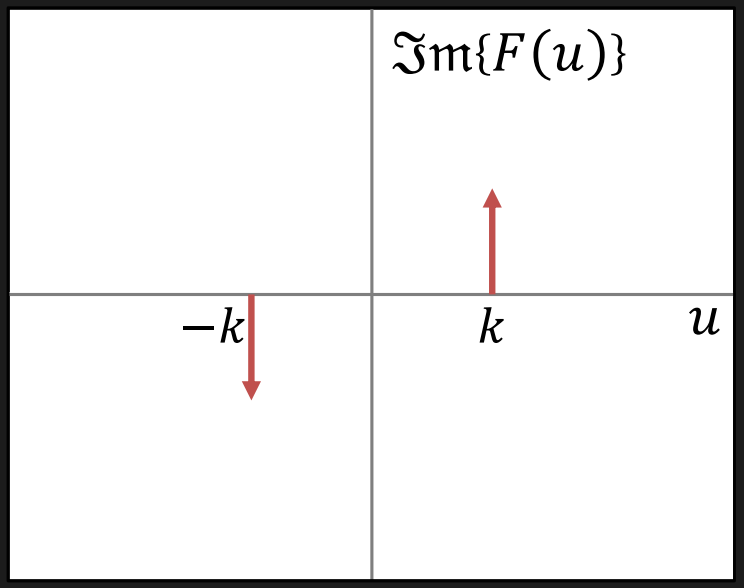
Fourier Transform Examples

Signal $f(x)$



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

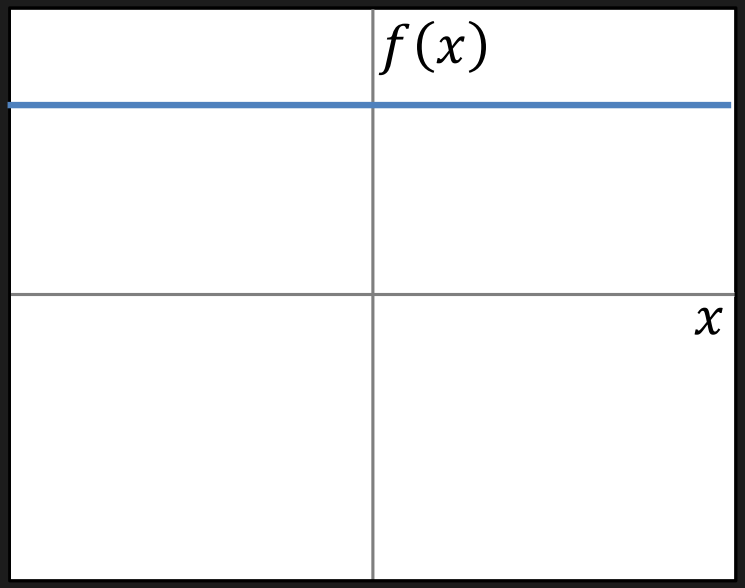
Fourier Transform $F(u)$



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

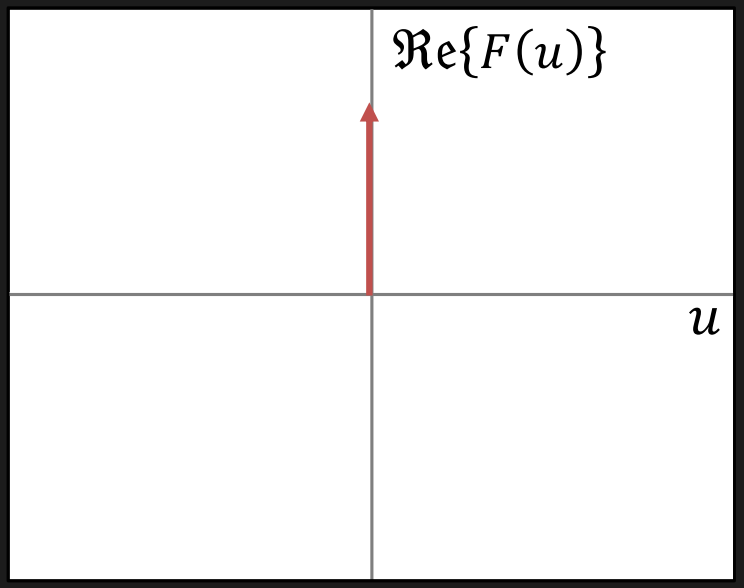
Fourier Transform Examples

Signal $f(x)$



$$f(x) = 1$$

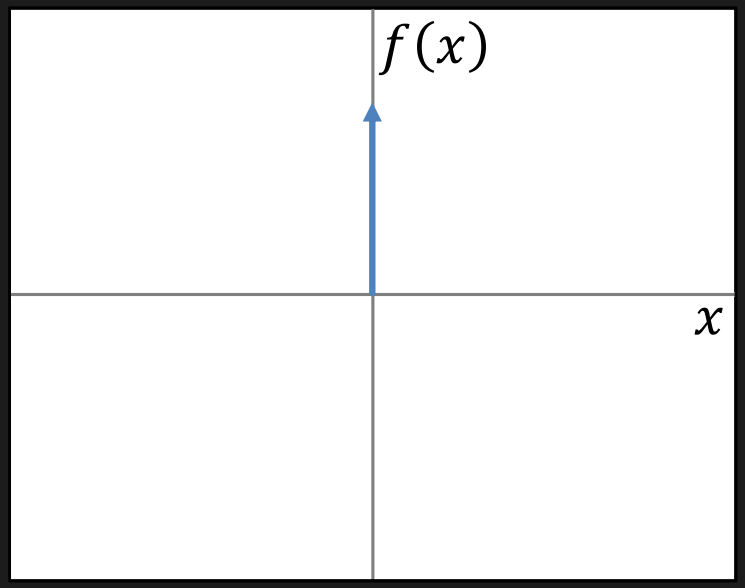
Fourier Transform $F(u)$



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

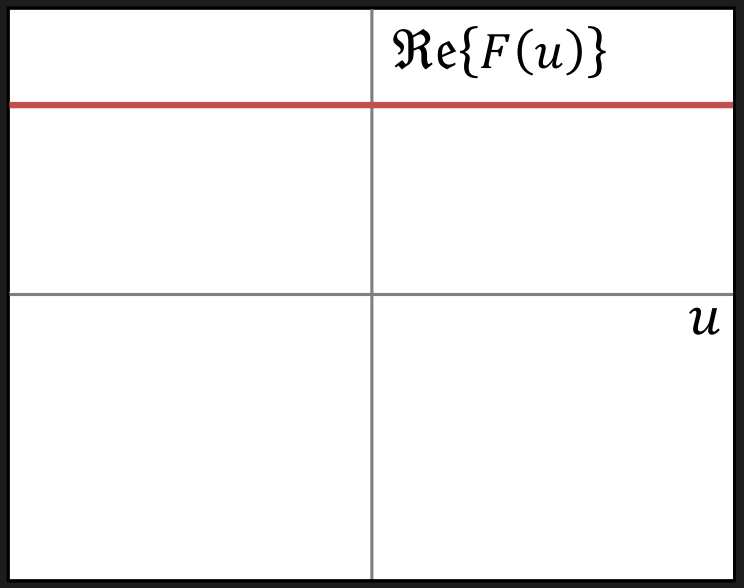
Fourier Transform Examples

Signal $f(x)$



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

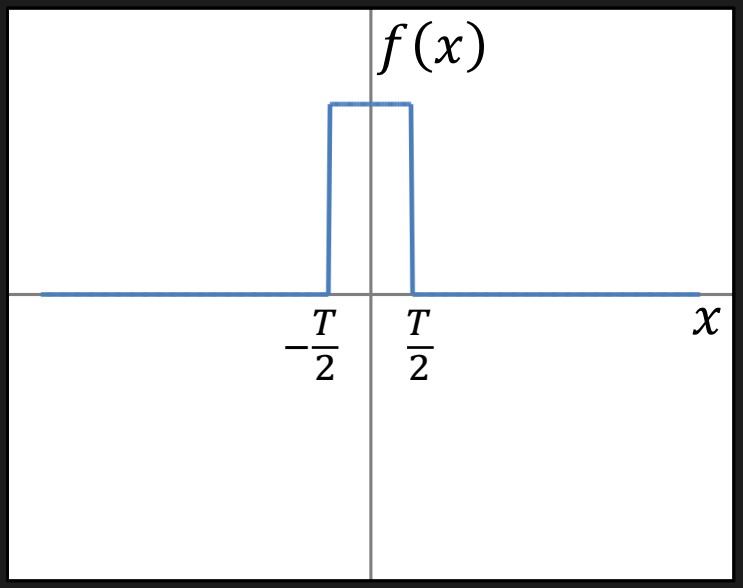
Fourier Transform $F(u)$



$$F(u) = 1$$

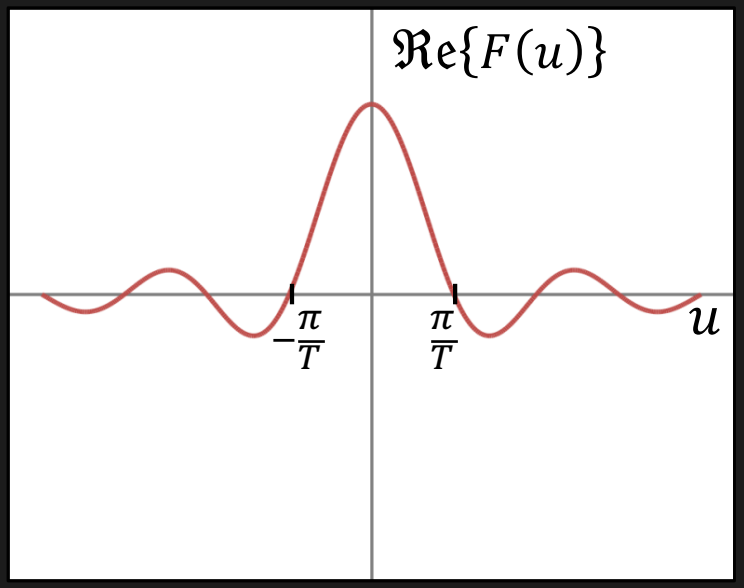
Fourier Transform Examples

Signal $f(x)$



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

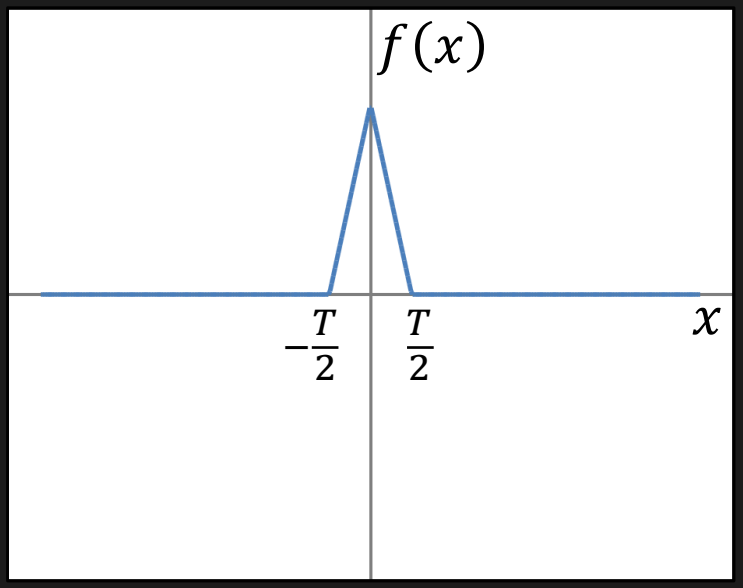
Fourier Transform $F(u)$



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

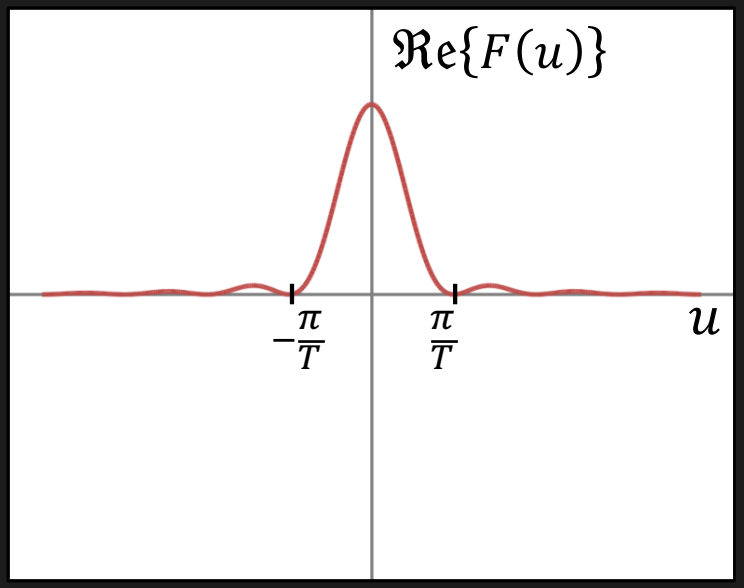
Fourier Transform Examples

Signal $f(x)$



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

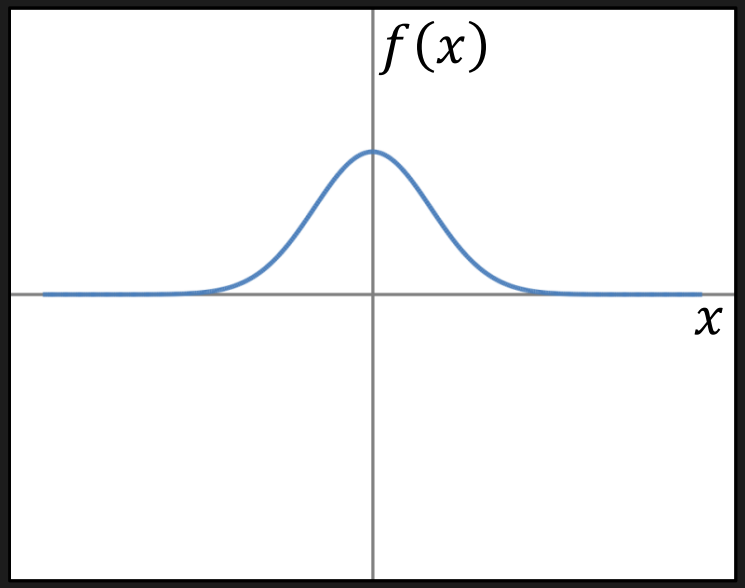
Fourier Transform $F(u)$



$$F(u) = T\,\mathrm{sinc}^2\,Tu$$

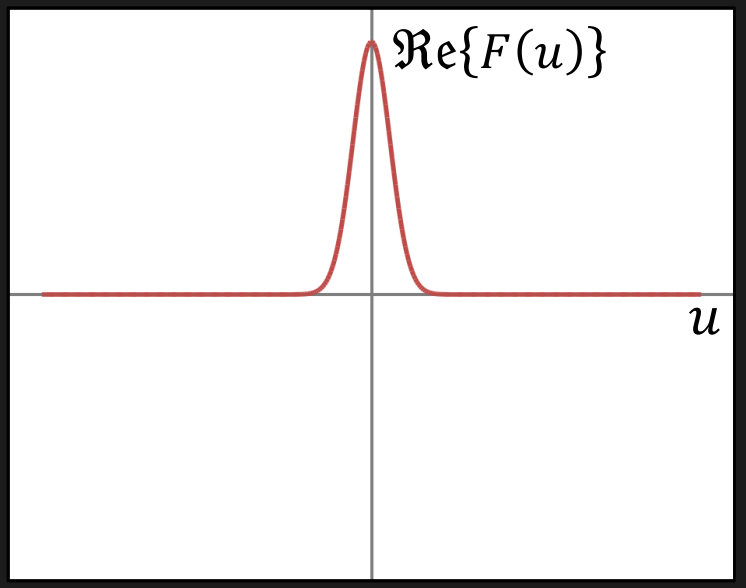
Fourier Transform Examples

Signal $f(x)$



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

Fourier Transform $F(u)$



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

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

The Convolution Theorem

Properties of Fourier Transform

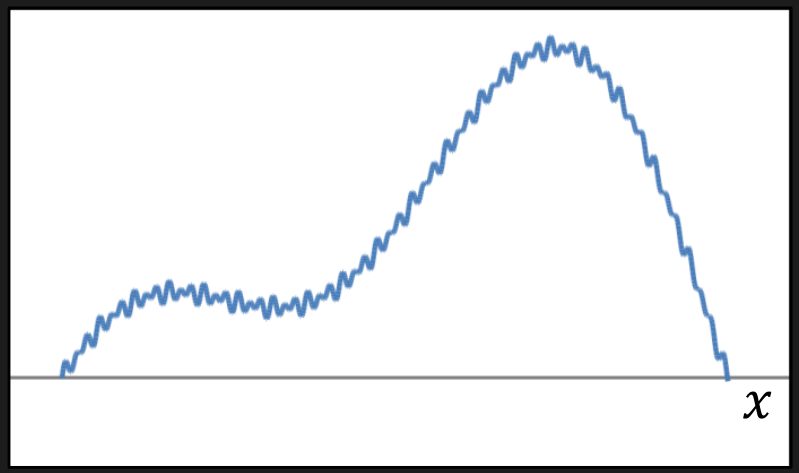
Property	Spatial Domain	Frequency Domain
Linearity	$\alpha f_1(x) + \beta f_2(x)$	$\alpha F_1(u) + \beta F_2(u)$
Scaling	$f(ax)$	$\frac{1}{\lvert a \rvert} F\left(\frac{u}{a}\right)$
Shifting	$f(x-a)$	$e^{-i2\pi ua} F(u)$
Differentiation	$\frac{d^n}{dx^n}\left(f(x)\right)$	$(i2\pi u)^n F(u)$

Convolution Using Fourier Transform

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

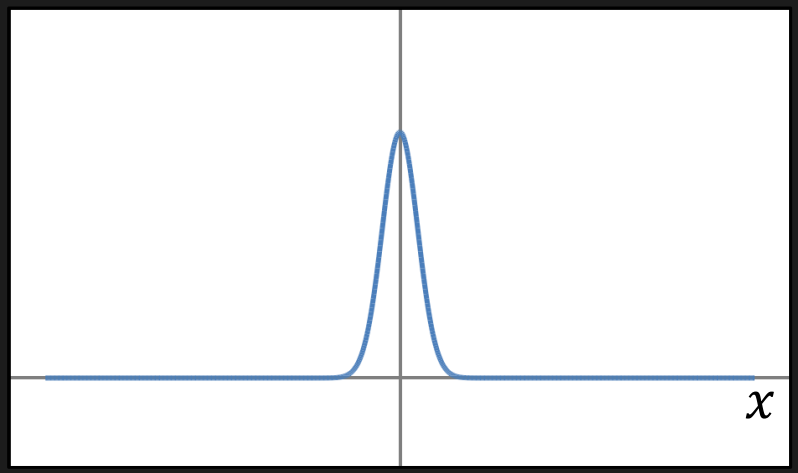
$$G(u) = F(u) \times H(u)$$

Gaussian Smoothing in Fourier Domain



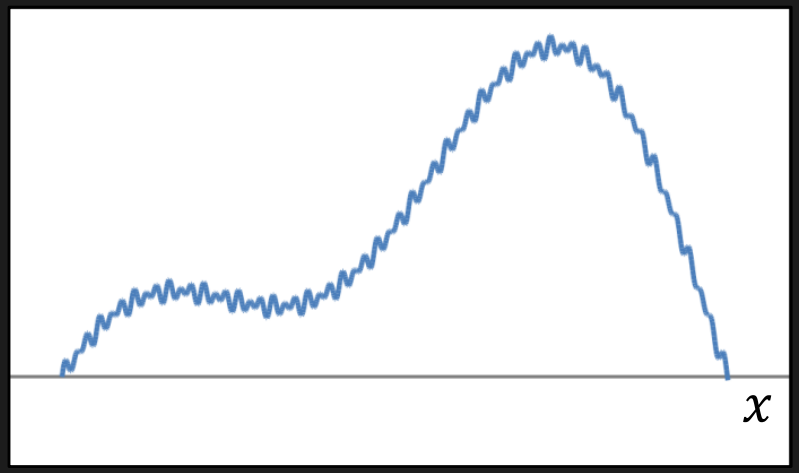
Noisy Signal $f(x)$

$*$

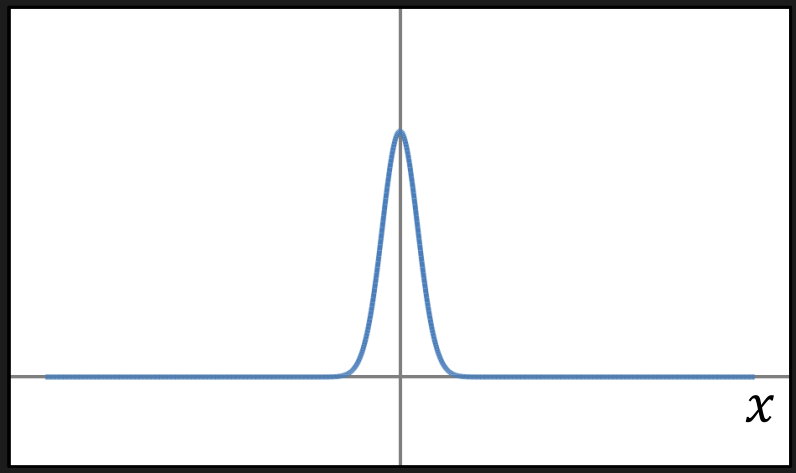


Gaussian Kernel $n_\sigma(x)$

Convolve the Noisy Signal with a Gaussian Kernel



*

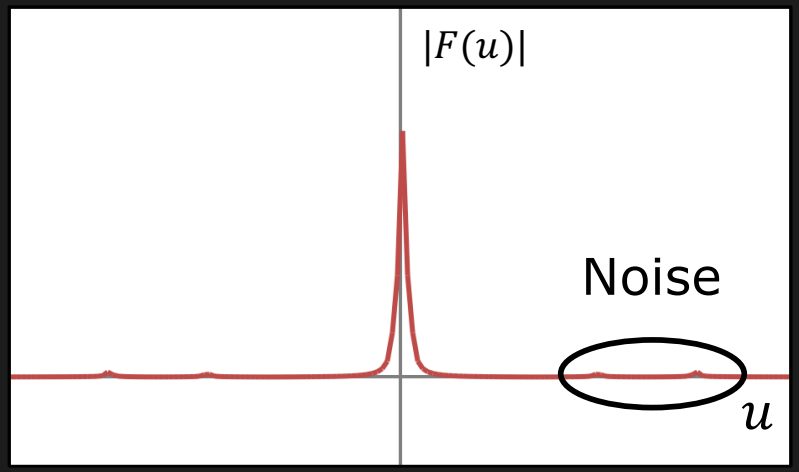


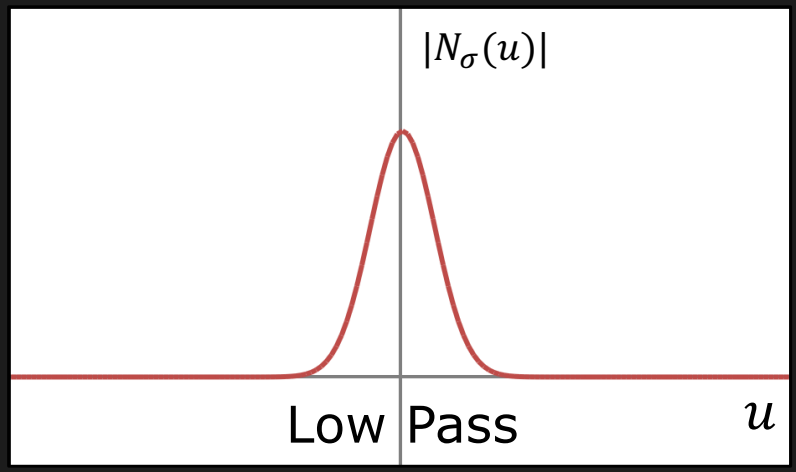
Noisy Signal $f(x)$

Gaussian Kernel $n_\sigma(x)$

FT

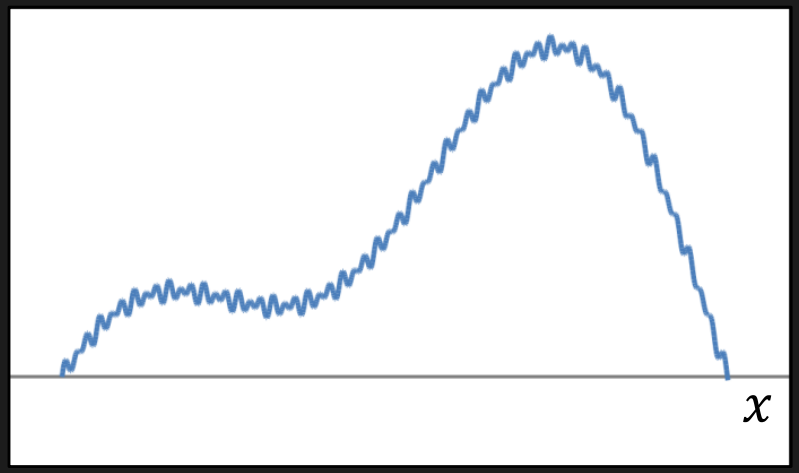
FT



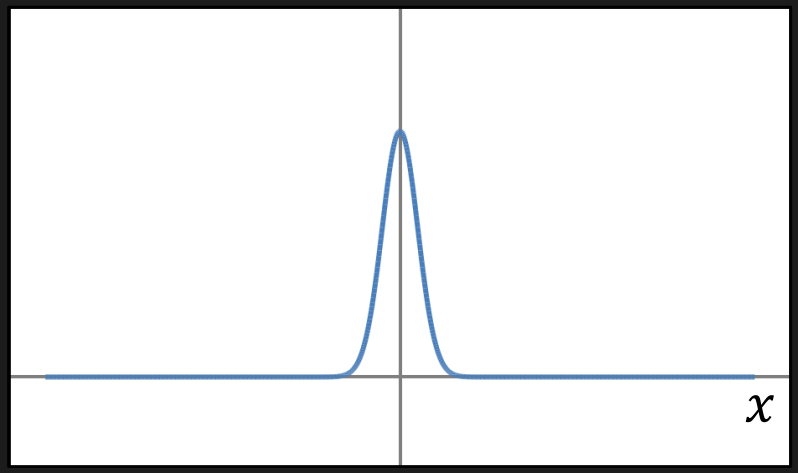


$F(u)$

$N_\sigma(u)$



*

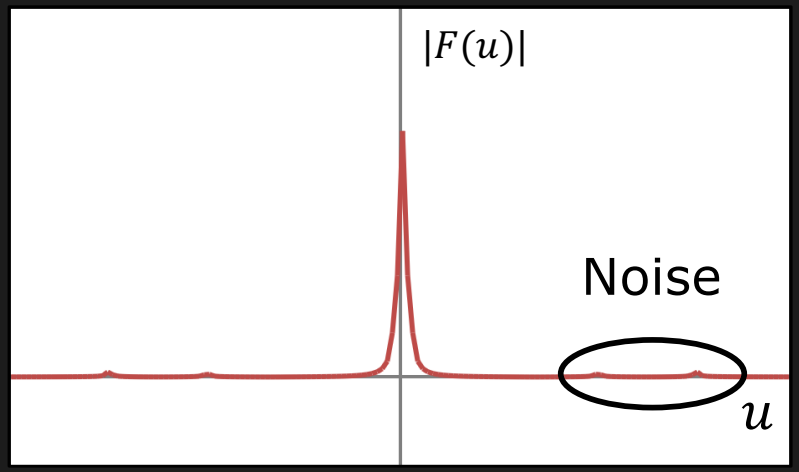


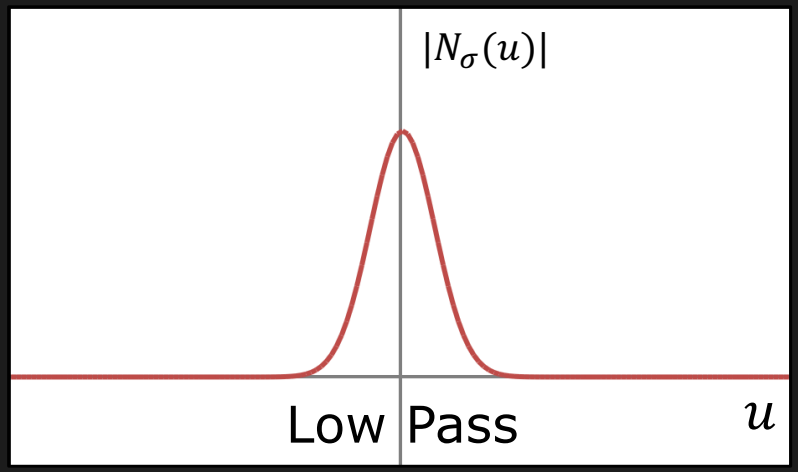
Noisy Signal $f(x)$

Gaussian Kernel $n_\sigma(x)$

FT

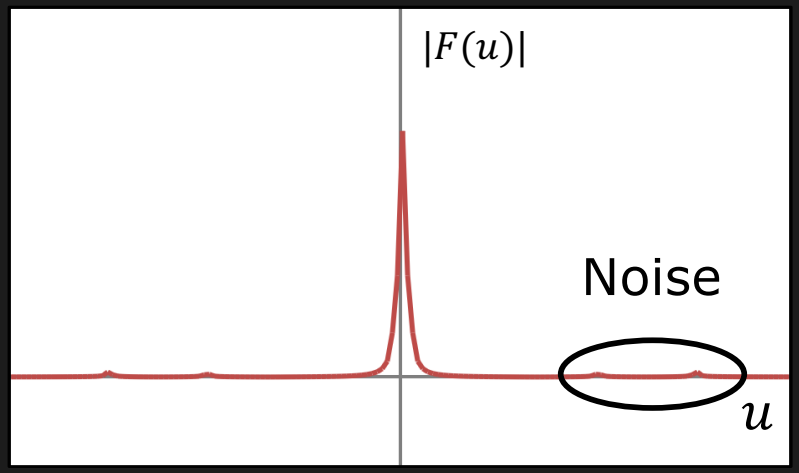
FT



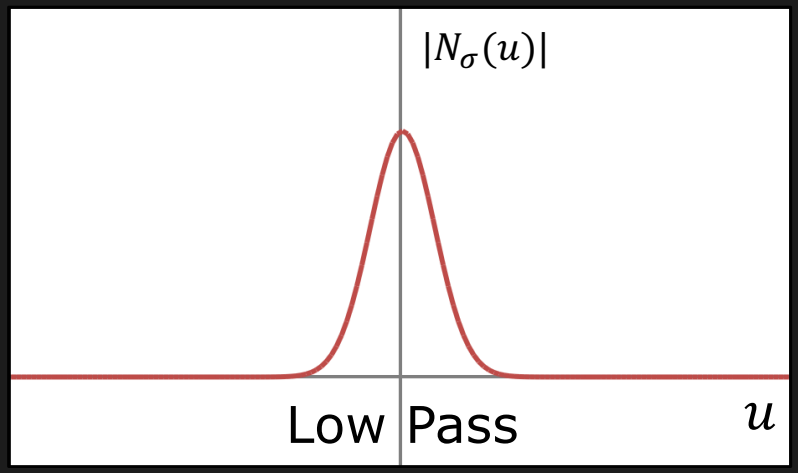


$F(u)$

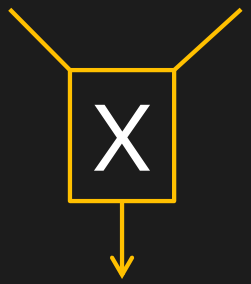
$N_\sigma(u)$

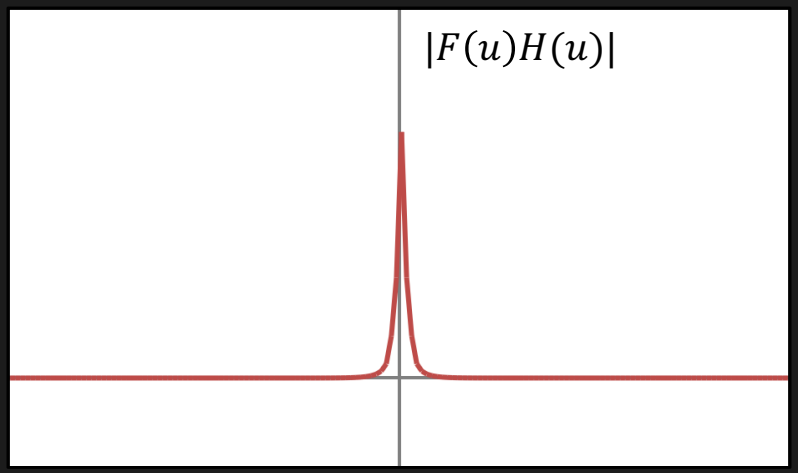


$F(u)$

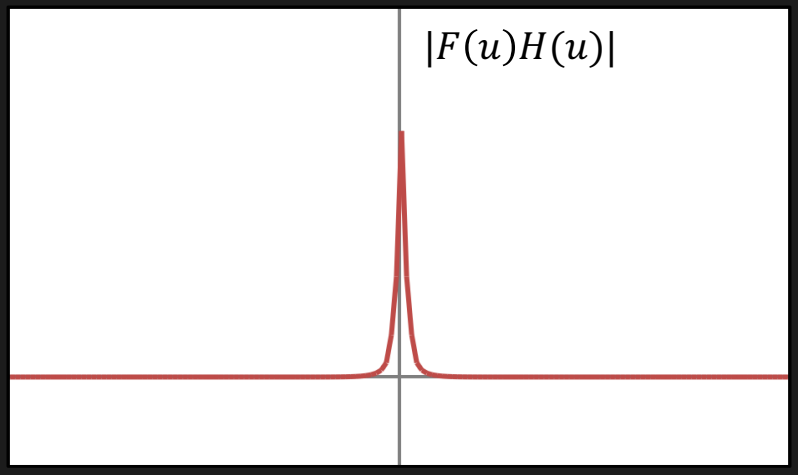


$N_\sigma(u)$



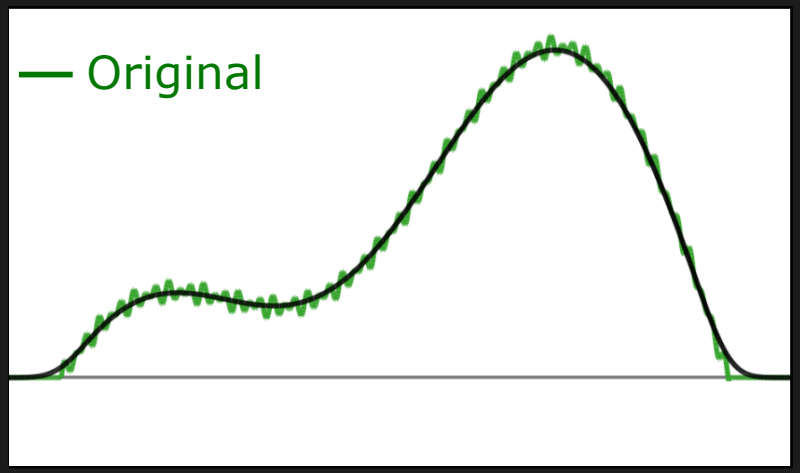


$F(u)H(u)$



$F(u)H(u)$

IFT



Gaussian Blurred Signal $g(x)$

Summary

Fourier Transforms: A way to represent signals in terms of their frequency content rather than their spatial structure

Essential concepts in this lecture:

1) The concept of the Fourier transform
2) The idea of frequency domain representations
3) The convolution theorem

Image Processing II

Computer Vision: CS 600.461/661

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) Nonlinear Image Processing

Minilectures 3, 4, and 5

Computer Vision: Algorithms and Appications (Chapter 3.3,3.4)
Szelinski, 2011 (available online)

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):

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

$p = 0 \ldots M-1$

$q = 0 \ldots N-1$

p and q are frequencies along m and n, respectively

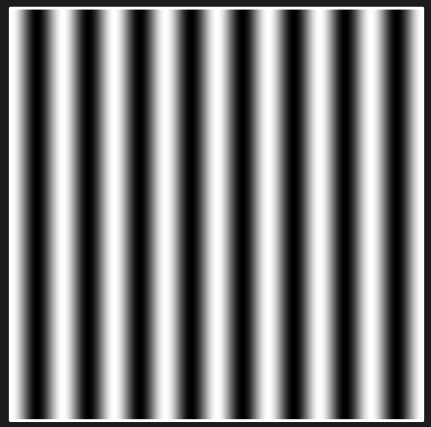
Inverse Discrete Fourier Transform (IDFT):

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

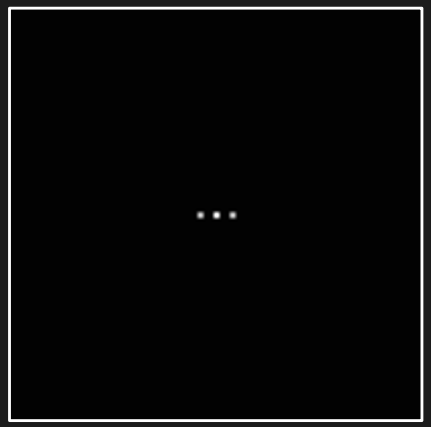
$m = 0 \ldots M-1$

$n = 0 \ldots N-1$

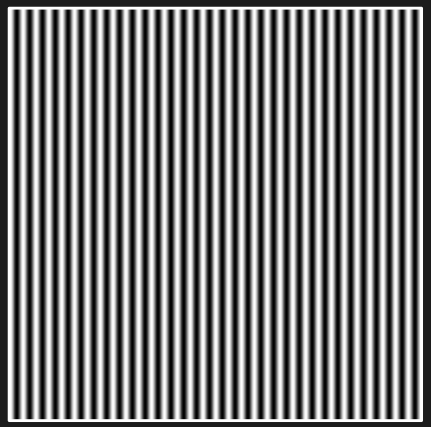
2D Fourier Transform: Example 1



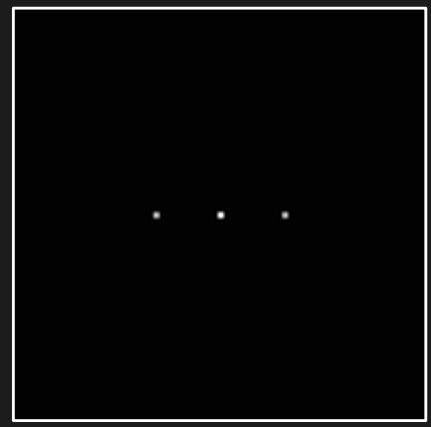
$f(m,n)$



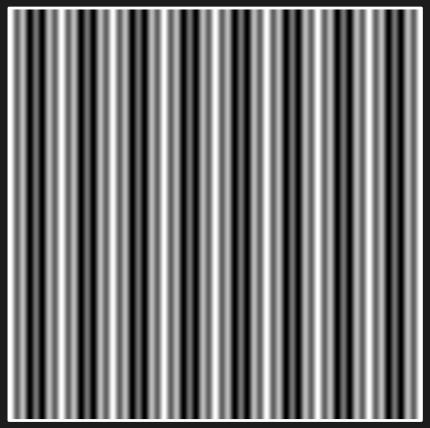
$\log(|F(p,q)|)$



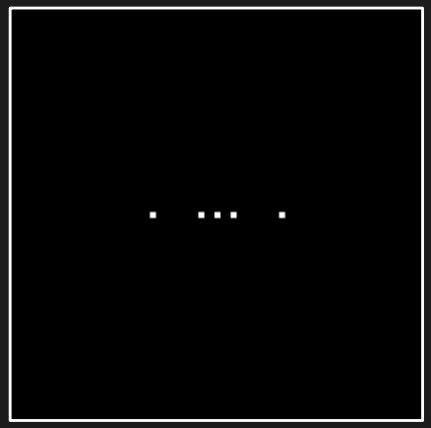
$g(m,n)$



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



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



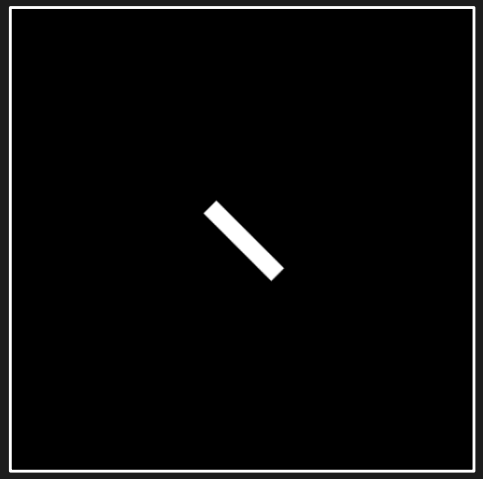
$\log(|F(p,q) + G(p,q)|)$

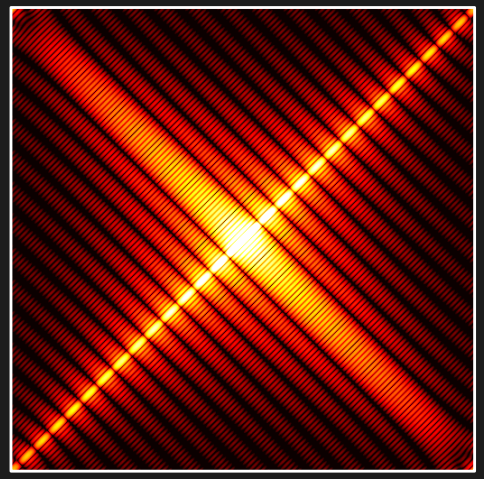
Note: $\log(|F|)$ is used just for display

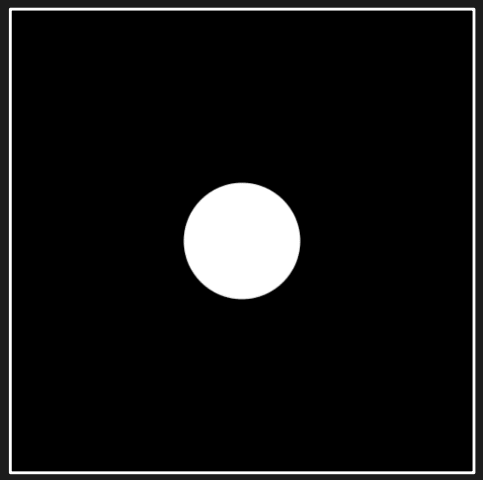
2D Fourier Transform: Example 2

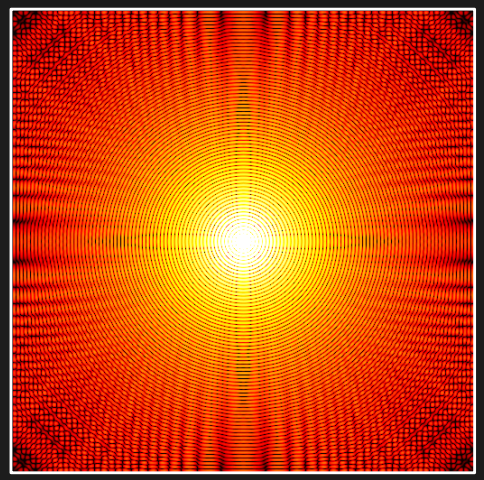
$f(m,n)$

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









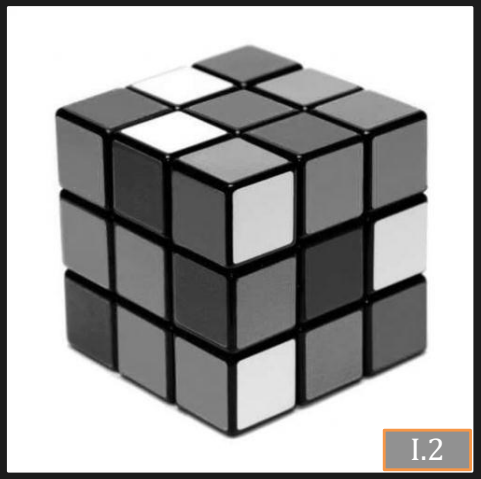
Min

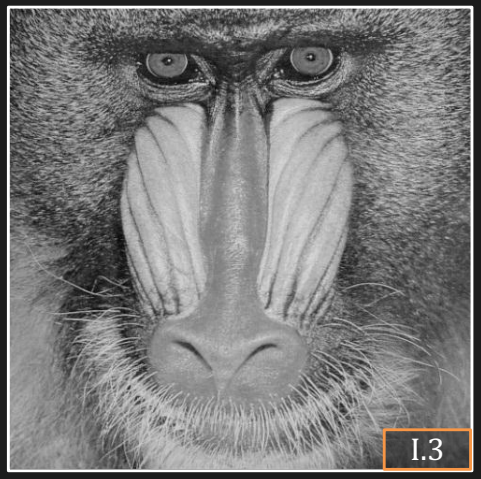
Max

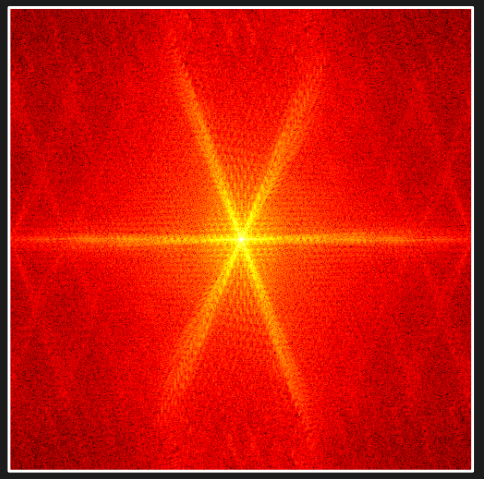
2D Fourier Transform: Example 3

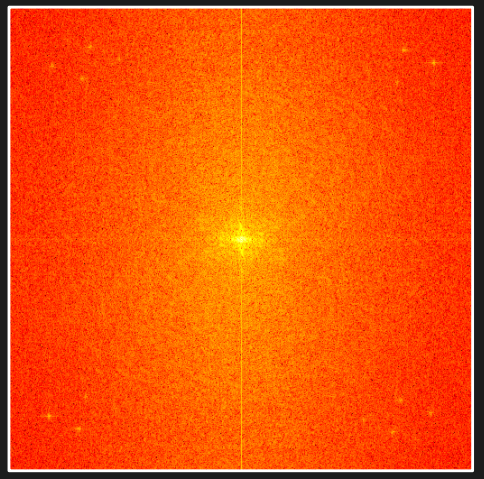
$f(m,n)$

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







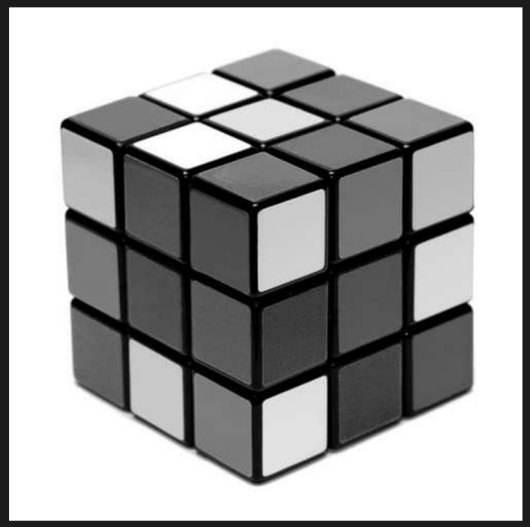


Min

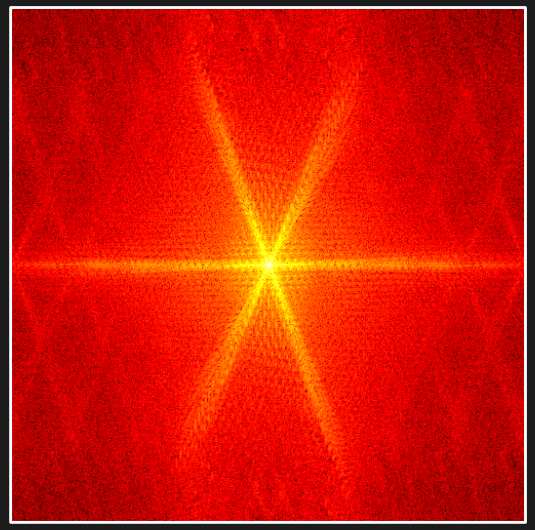
Max

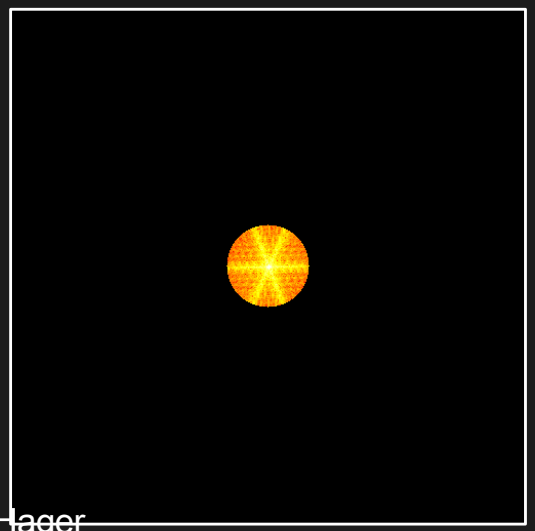
Low Pass Filtering

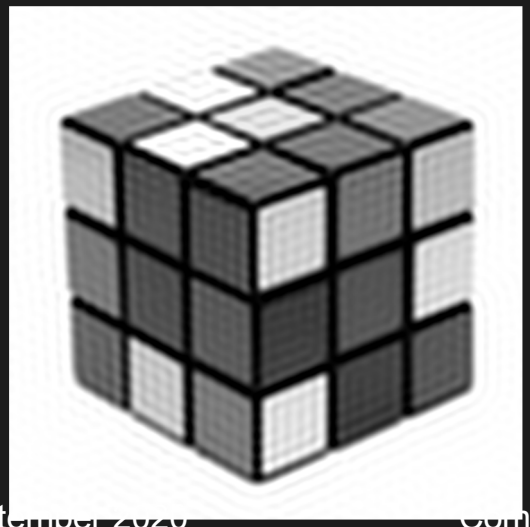
$f(m,n)$



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

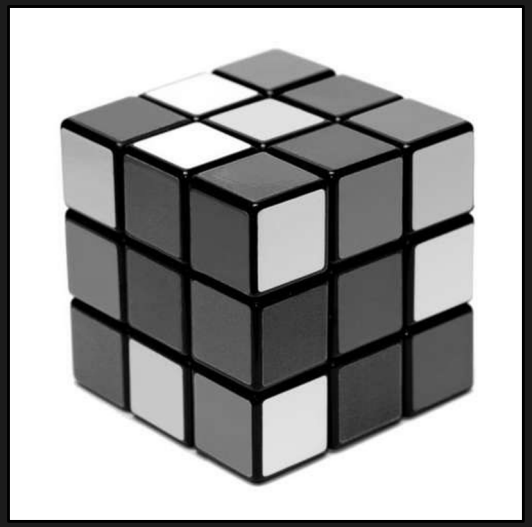




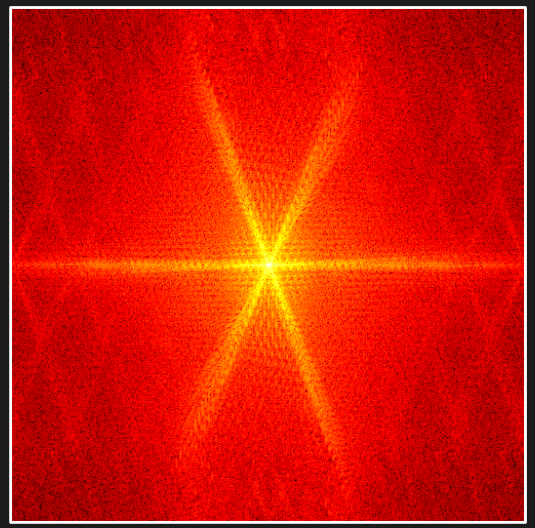


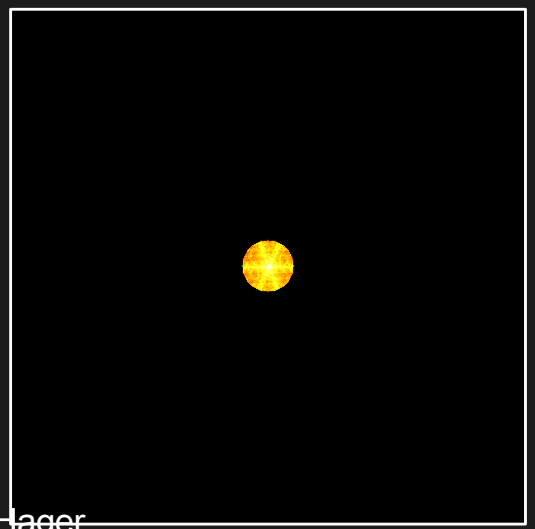
Low Pass Filtering

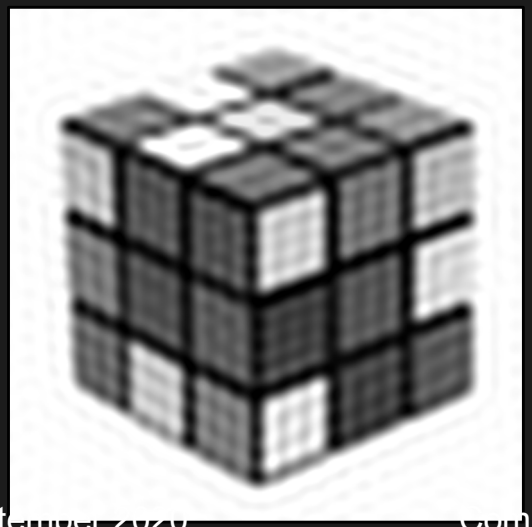
$f(m,n)$



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

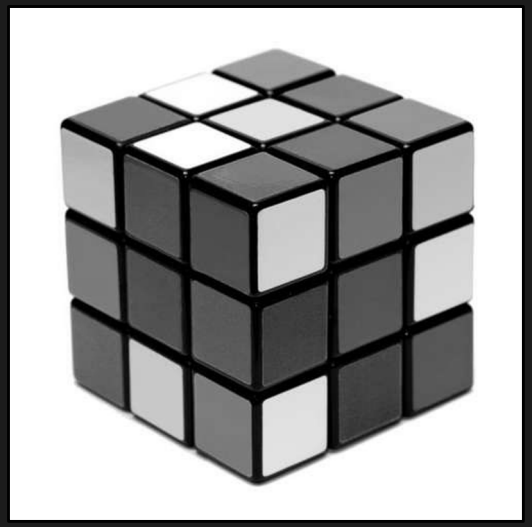




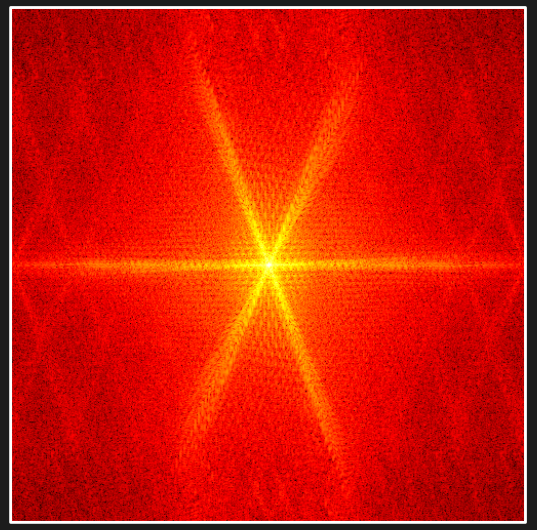


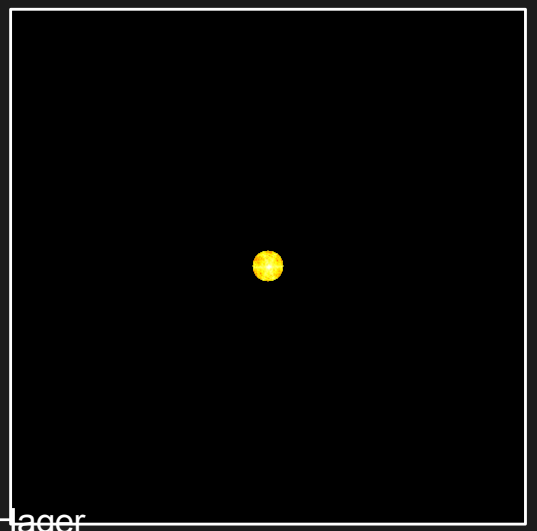
Low Pass Filtering

$f(m,n)$



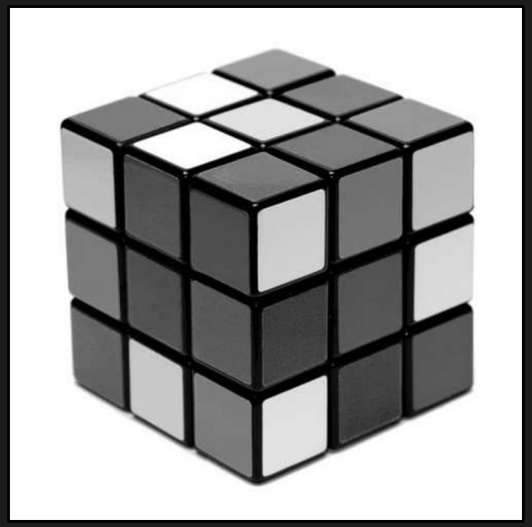
$\log(|F(p,q)|)$



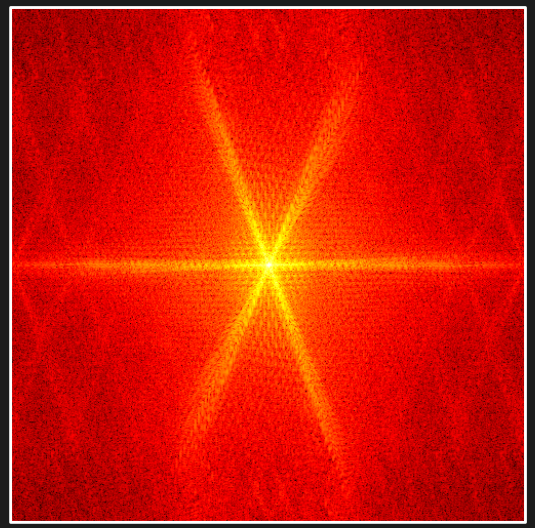


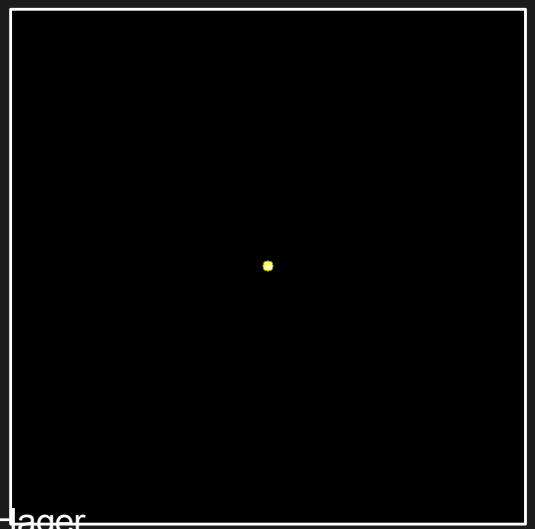
Low Pass Filtering

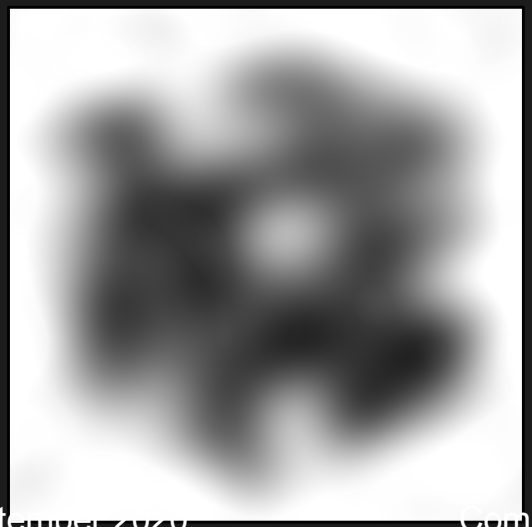
$f(m,n)$



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

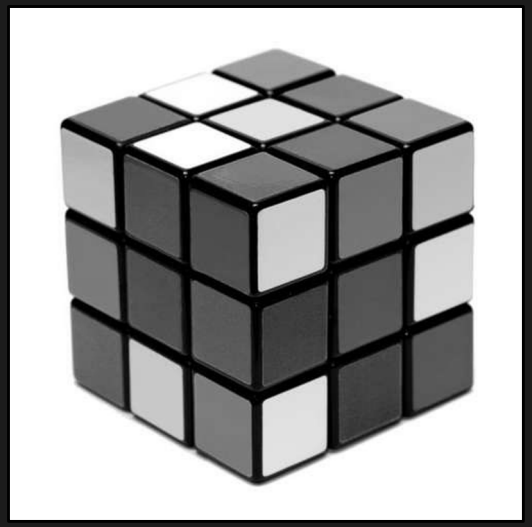




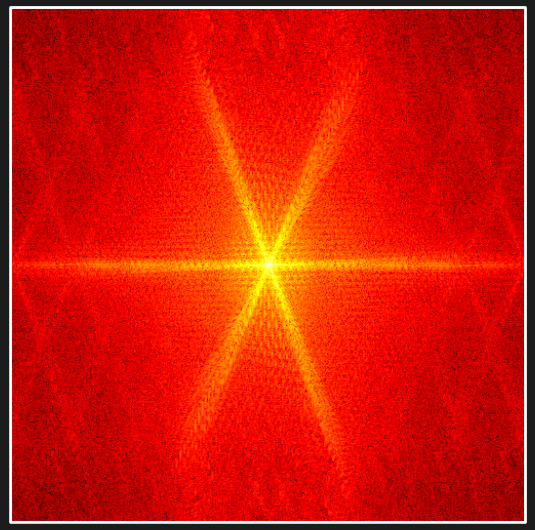


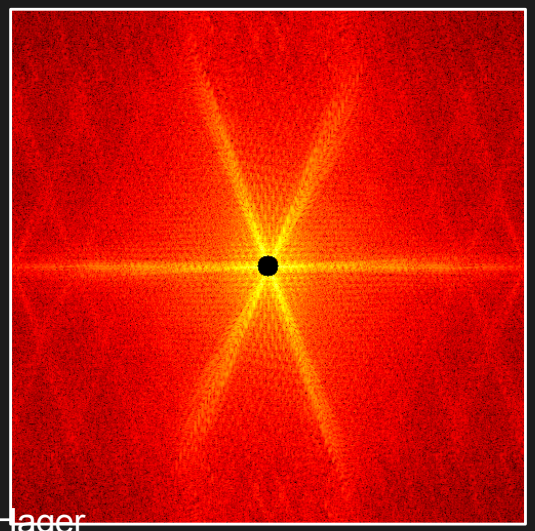
High Pass Filtering

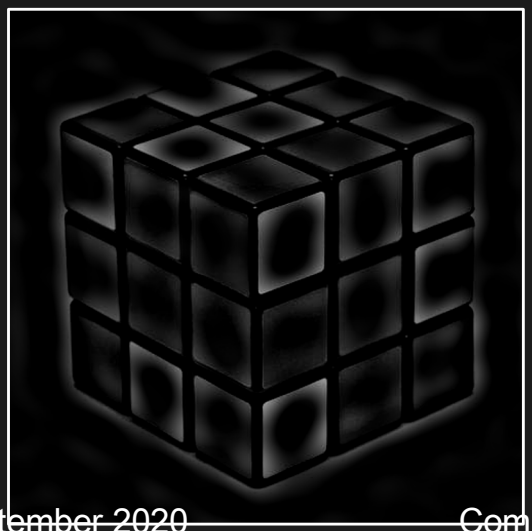
$f(m,n)$



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

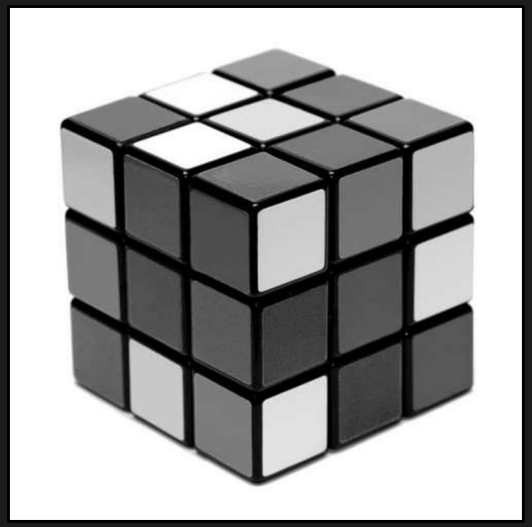




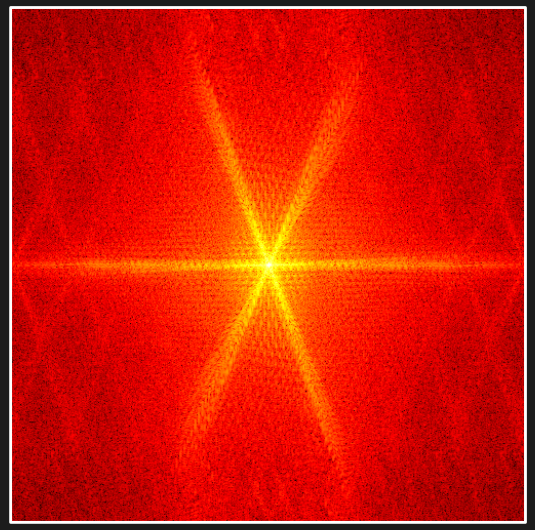


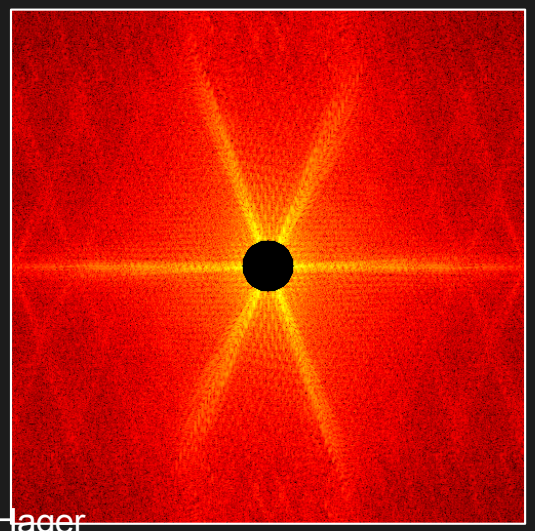
High Pass Filtering

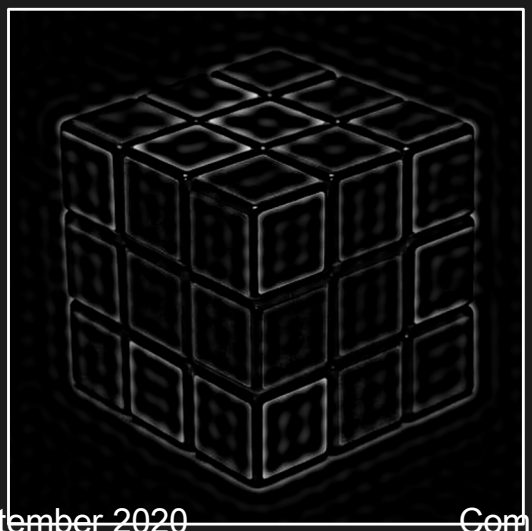
$f(m,n)$



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



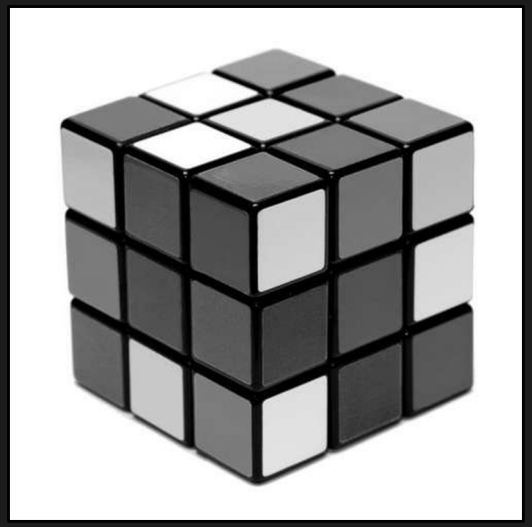


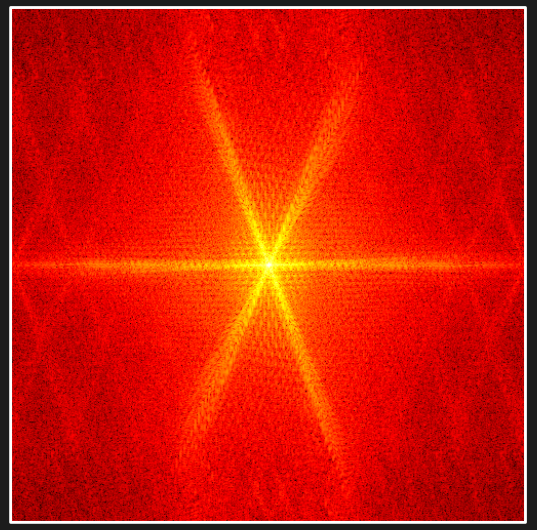


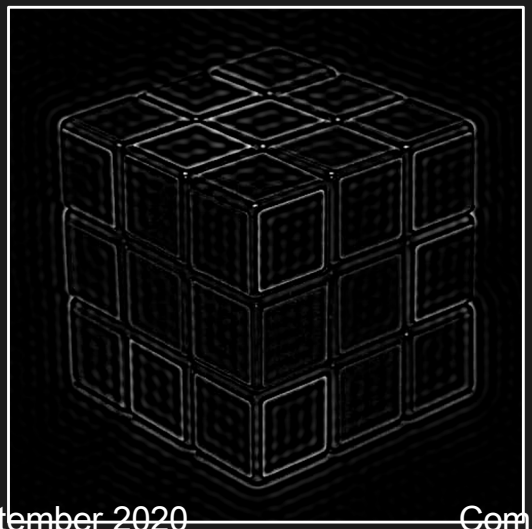
High Pass Filtering

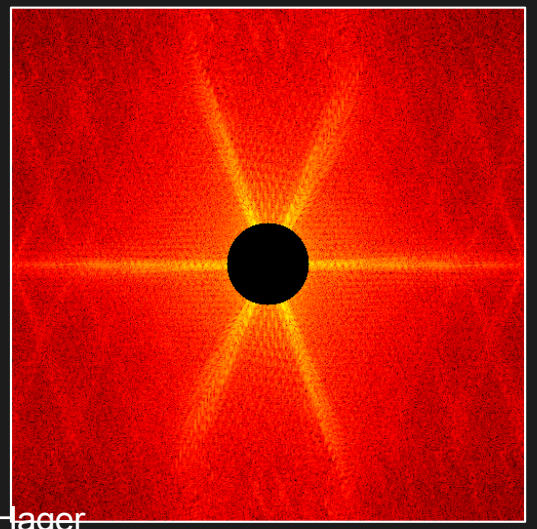
$f(m,n)$

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



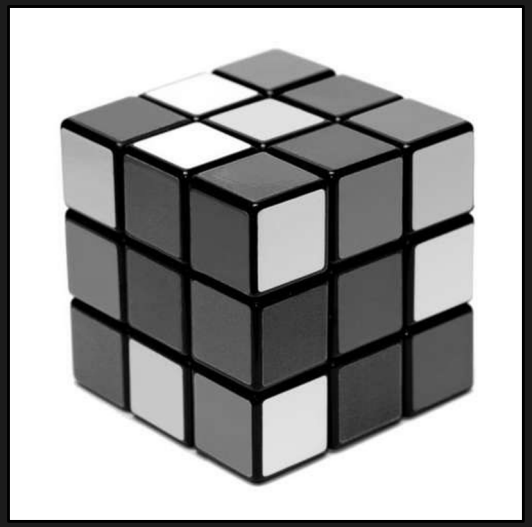




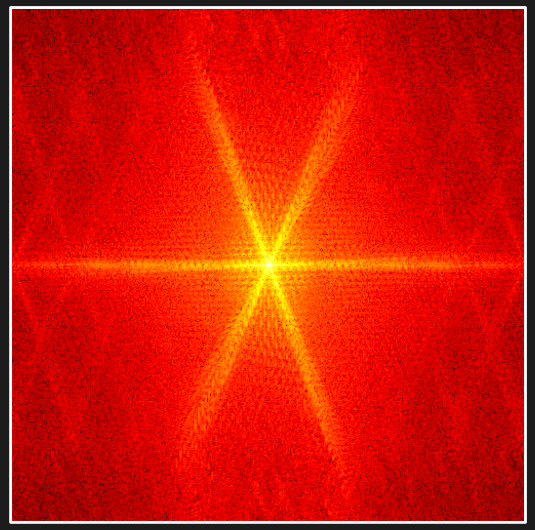


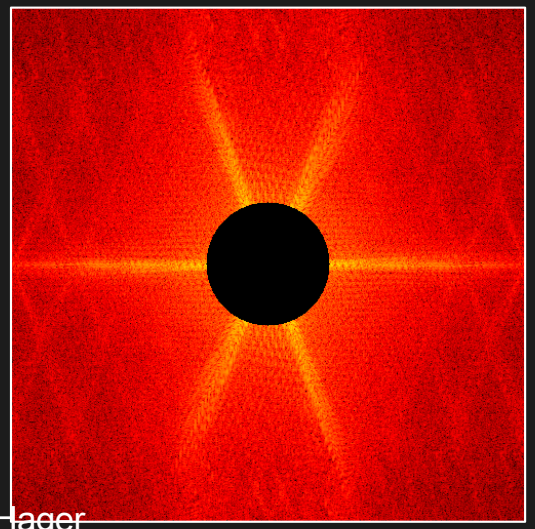
High Pass Filtering

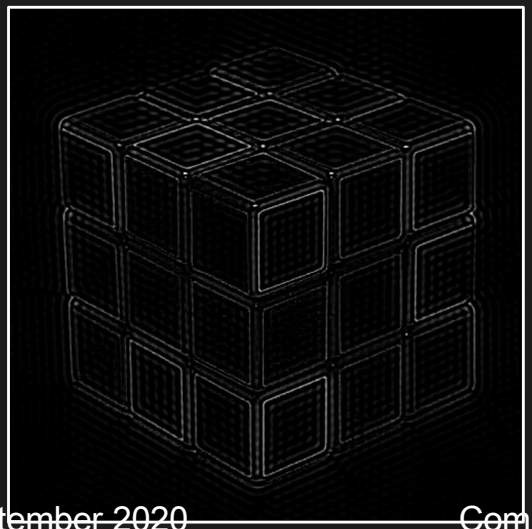
$f(m,n)$



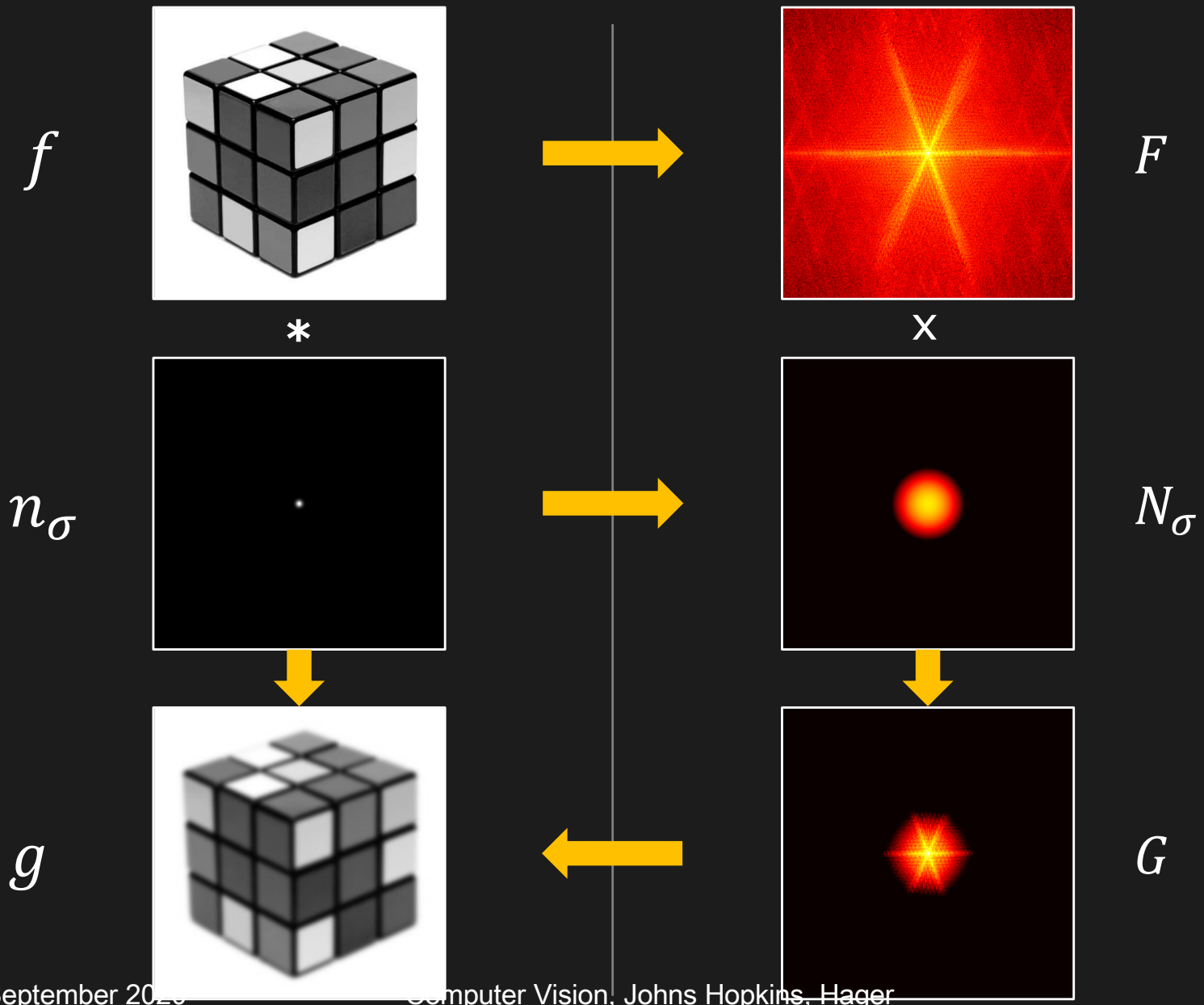
$\log(|F(p,q)|)$



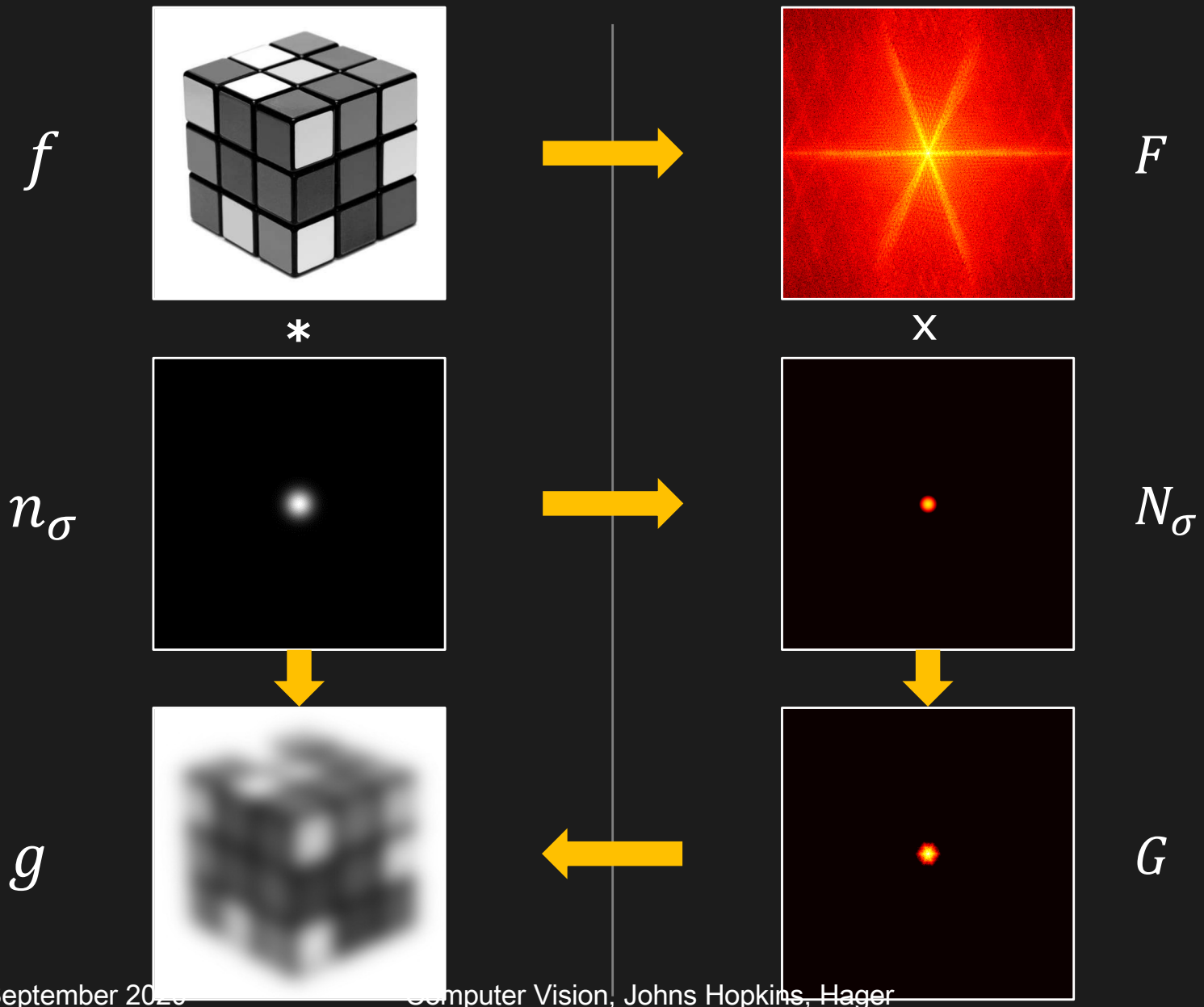




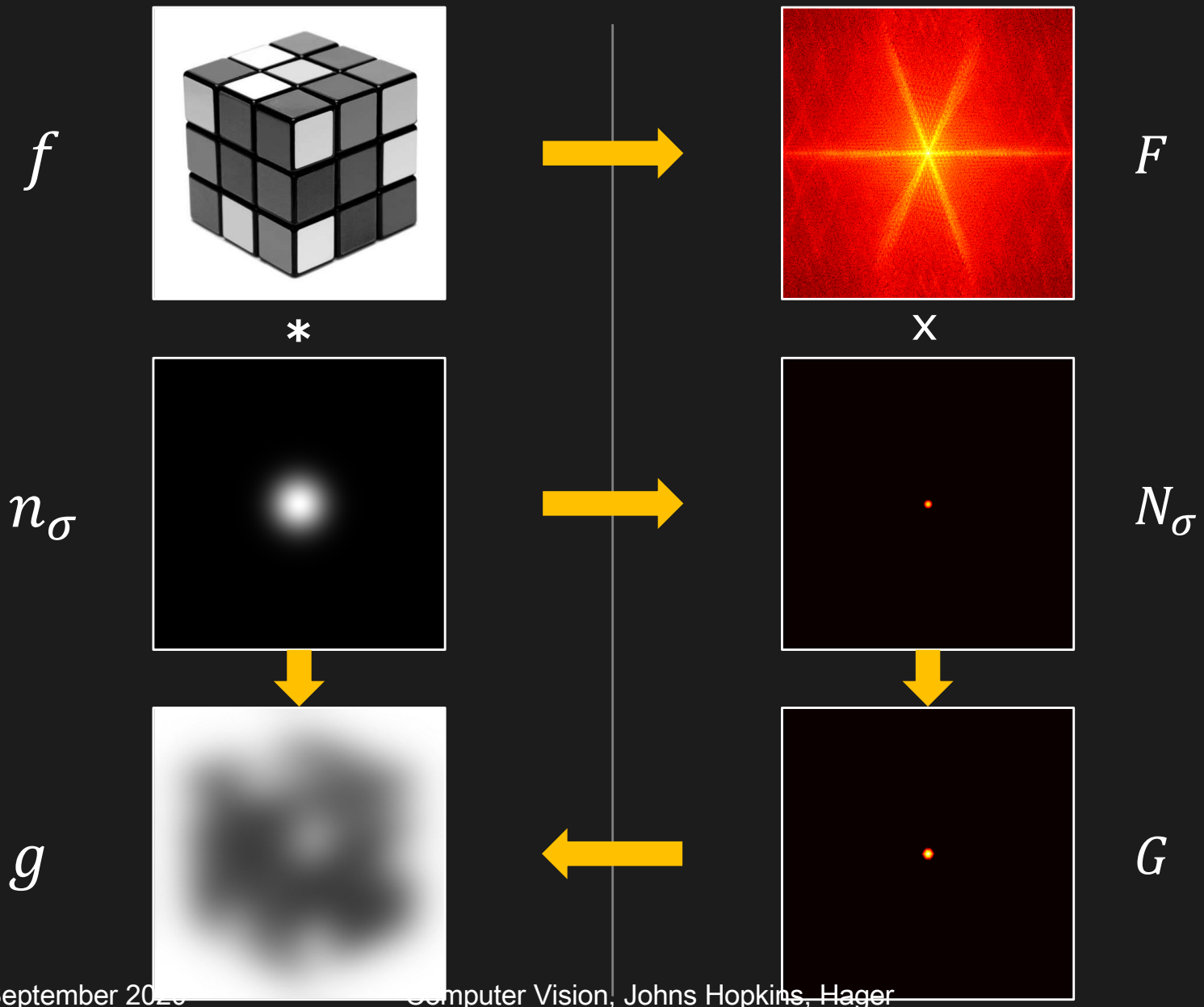
Gaussian Smoothing



Gaussian Smoothing



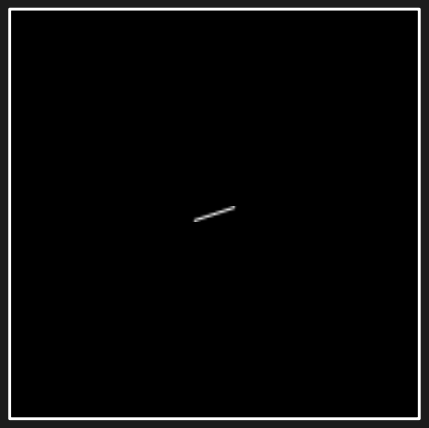
Gaussian Smoothing



Motion Blur

Scene $f(x,y)$

$*$



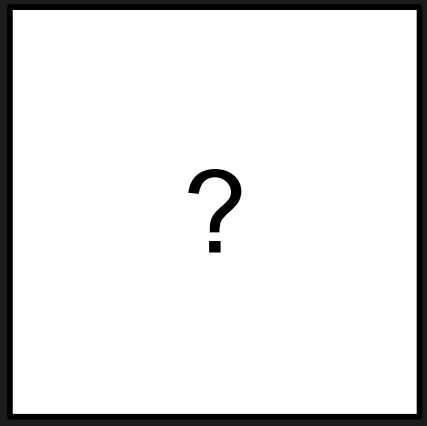
PSF $h(x,y)$
(Camera Shake)

$=$

Image $g(x,y)$

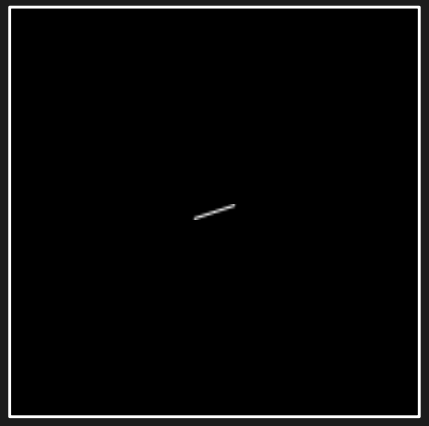
$$f(x,y) * h(x,y) = g(x,y)$$

Motion Blur



Scene $f(x,y)$

$*$



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

$=$

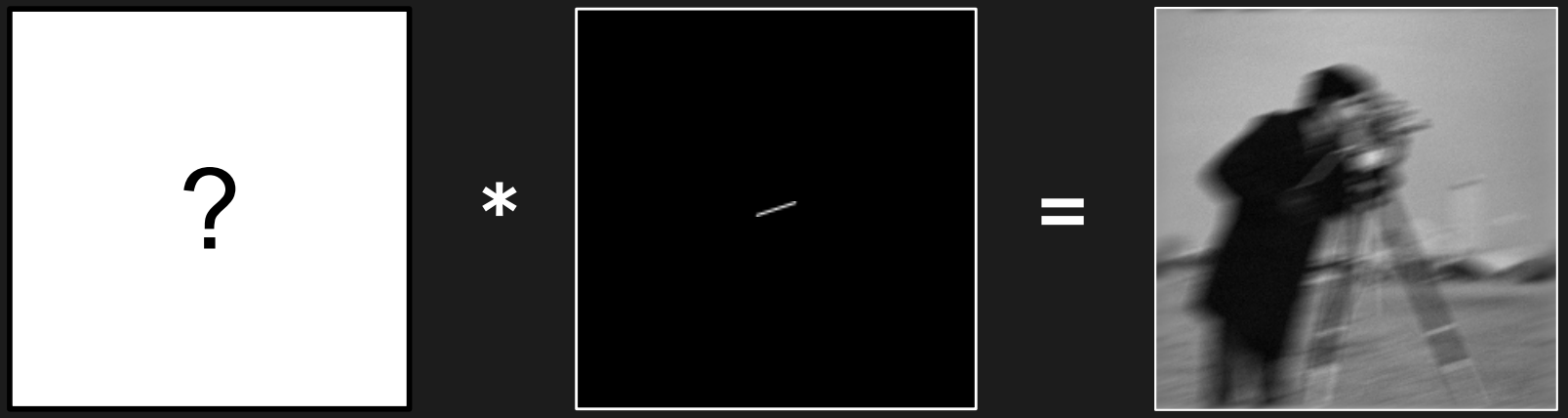
Image $g(x,y)$

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

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.

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

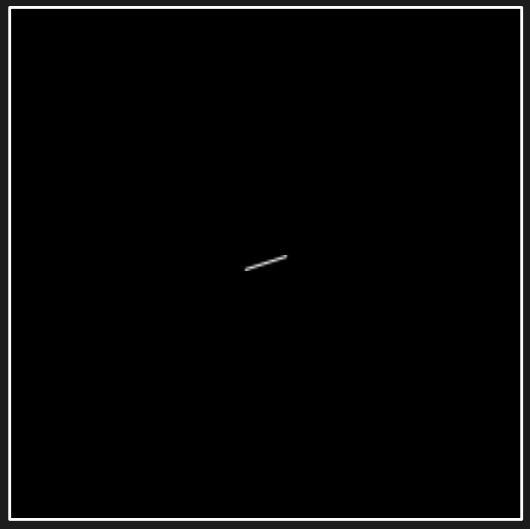
$$F'(u,v)H(u,v) \quad = G(u,v)$$

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

Motion Deblur: Deconvolution

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

/



=

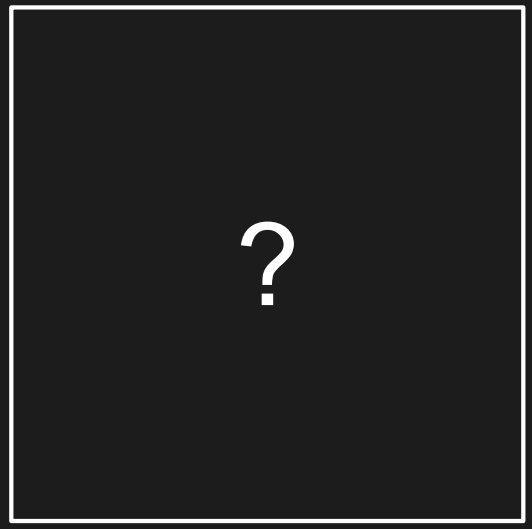


Image $g(x,y)$ PSF $h(x,y)$ Recovered $f'(x,y)$

Motion Deblur: Deconvolution

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

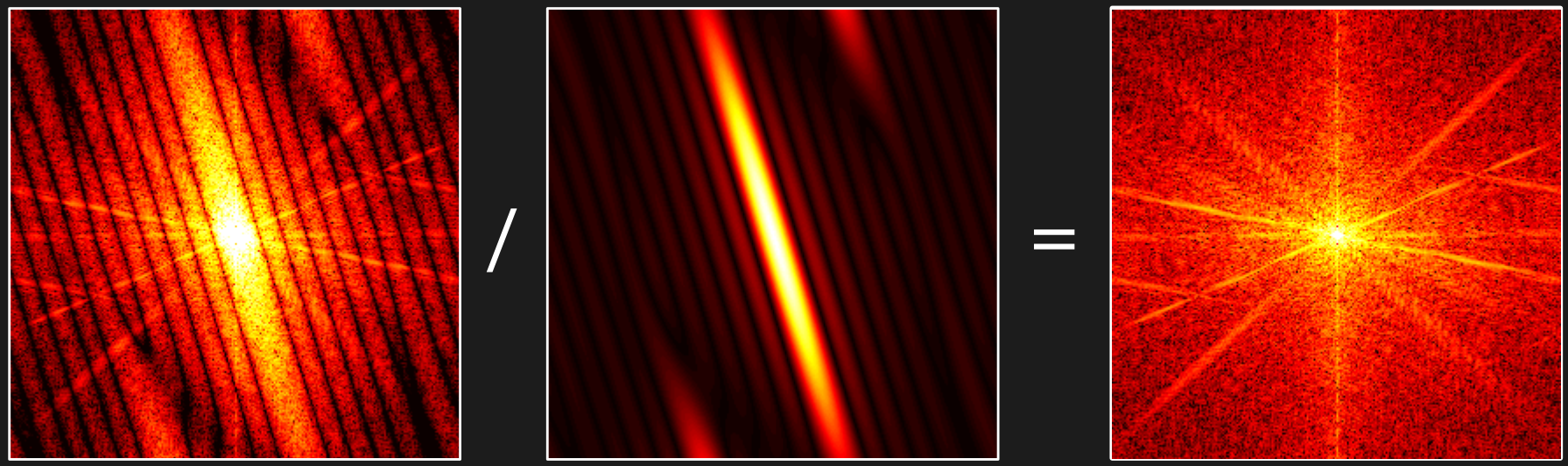


Image $G(u,v)$ PSF $H(u,v)$ Recovered $F'(u,v)$

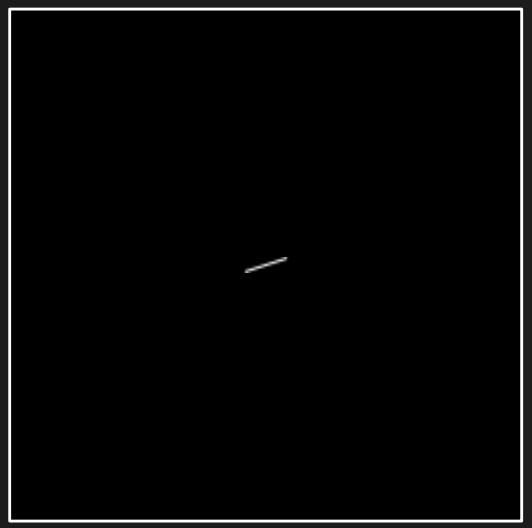
Step 1: Recover $F'(u,v)$ in Fourier Domain

Motion Deblur: Deconvolution

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

Image $g(x,y)$

/



PSF $h(x,y)$

=

Recovered $f'(x,y)$

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

Summary

Fourier Transforms: A way to represent signals in terms of their frequency content rather than their spatial structure

Essential concepts in this lecture:

1) The 2D Fourier transform in the discrete domain
2) The idea of 2D frequency domain representations
3) An example of using these ideas to deblur an image

Image Processing II

Computer Vision: CS 600.461/661

Image Processing II

Transform image to new one that is easier to manipulate.

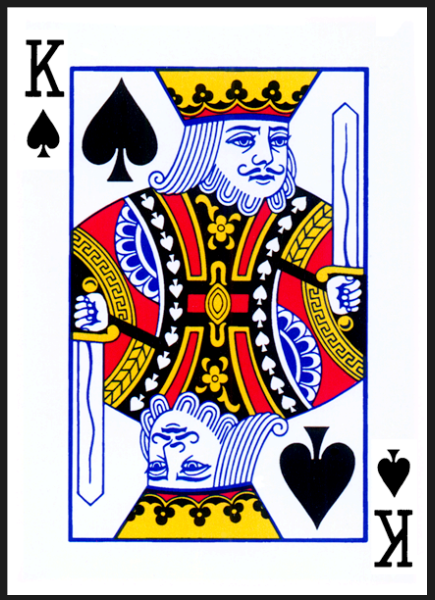
Topics:

(1) Correlation

(2) Median Filtering

Computer Vision: Algorithms and Applications (Chapter 3.2, 3.3)
Szelinski, 2011 (available online)

Template Matching



Template

How do we locate the template in the image?

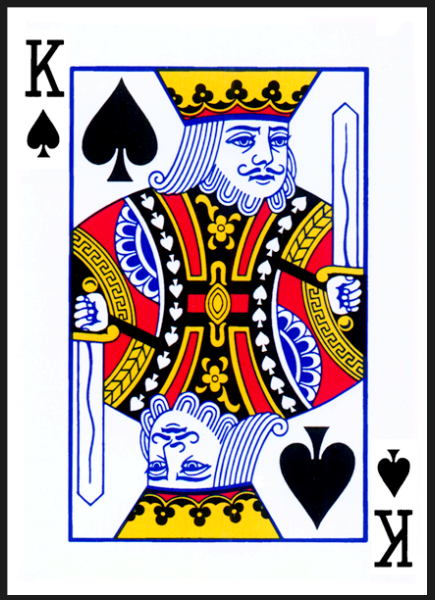
Minimize:

$$E[i,j] = \sum_m \sum_n (f[m,n] - t[m-i,n-j])^2$$

$$E[i,j] = \sum_m \sum_n (f^2[m,n] + t^2[m-i,n-j] - 2f[m,n]t[m-i,n-j])$$

Maximize

Template Matching



Template

How do we locate the template in the image?

Maximize:

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

(Cross-Correlation)

Convolution vs. Correlation

Convolution:

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

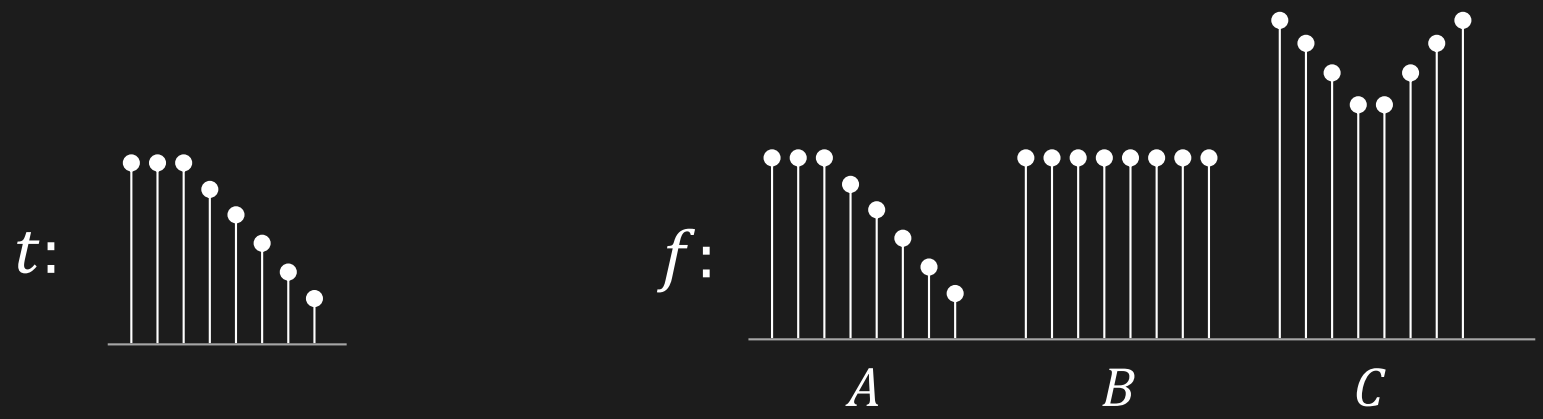
Correlation:

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

No Flipping in Correlation

Problem with Cross-Correlation

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



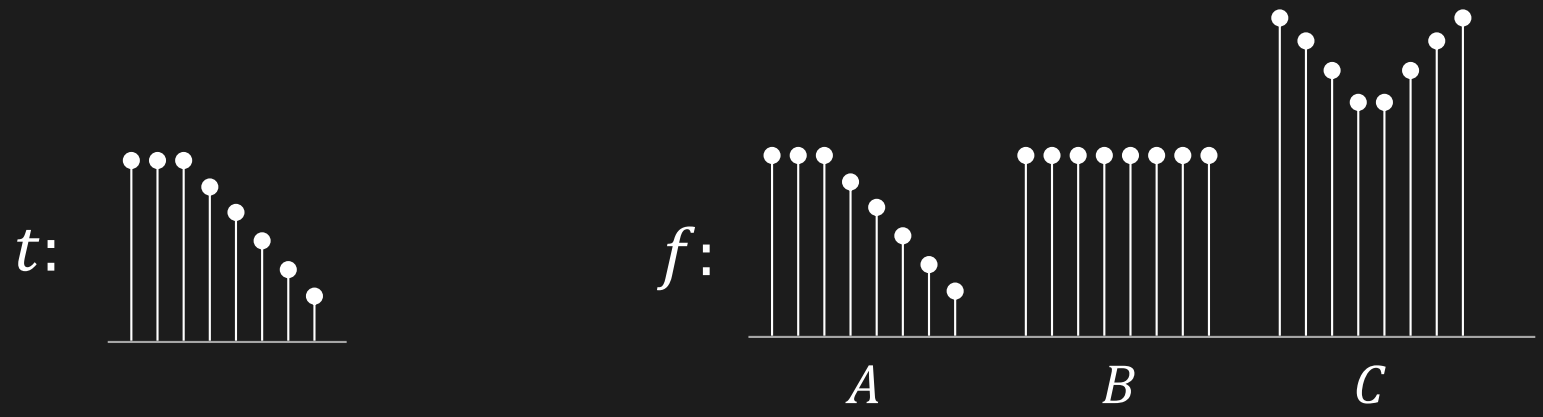
$$R_{tf}(C) > R_{tf}(B) > R_{tf}(A)$$

We need $R_{tf}(A)$ to be the maximum!

Normalized Cross-Correlation

Account for energy differences

$$N_{tf}[i,j] = \frac{\sum_m \sum_n f[m,n]t[m-i,n-j]}{\sqrt{\sum_m \sum_n f^2[m,n]}\sqrt{\sum_m \sum_n t^2[m-i,n-j]}}$$

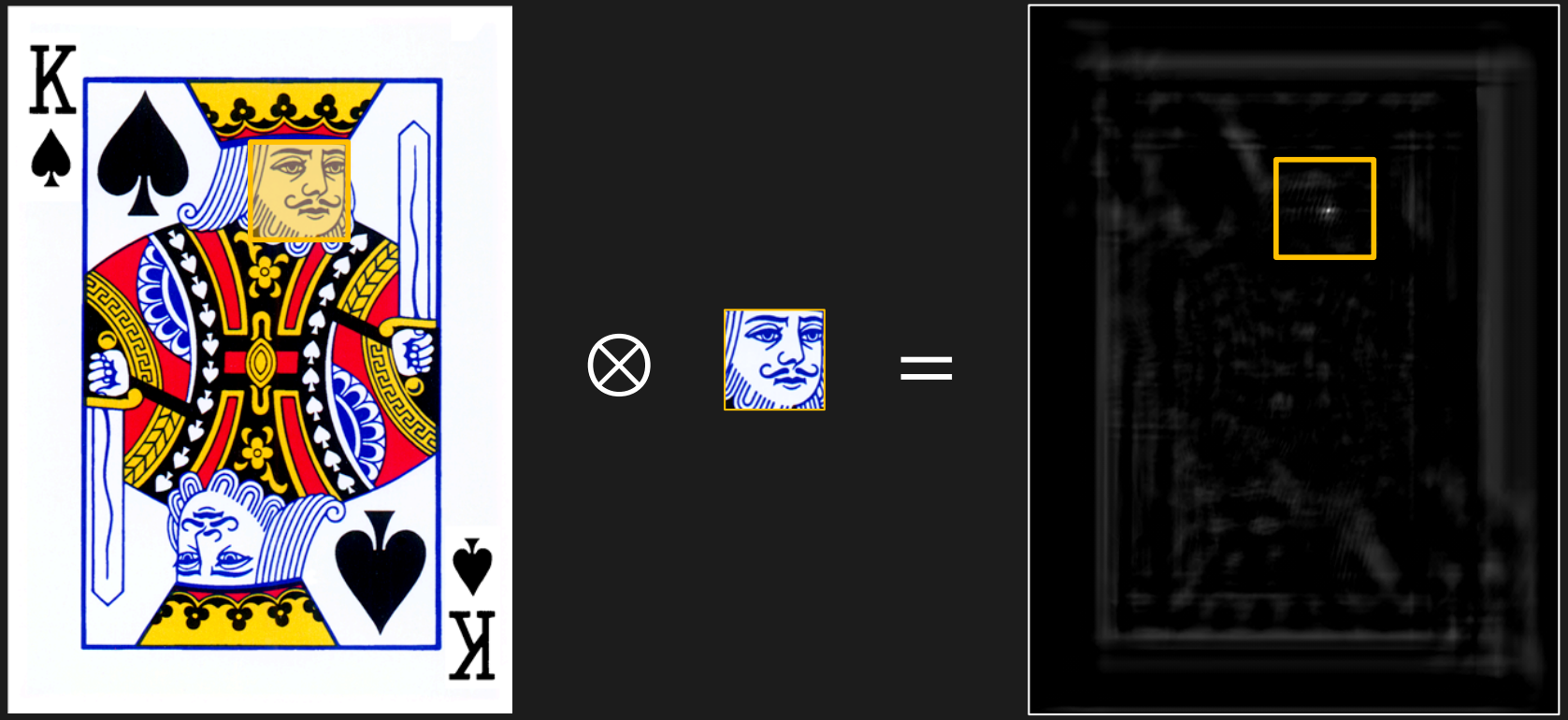


$$R_{tf}(A) > R_{tf}(B) > R_{tf}(C)$$

Normalized Cross-Correlation

Account for energy differences

$$N_{tf}[i,j] = \frac{\sum_m \sum_n f[m,n]t[m-i,n-j]}{\sqrt{\sum_m \sum_n f^2[m,n]}\sqrt{\sum_m \sum_n t^2[m-i,n-j]}}$$



Correlation: Issues

- Problem at borders
- Sensitive to object pose, scale and rotation
- Not good for general object recognition
- Good for feature detection
- Can be computationally expensive

Smoothing to Remove Image Noise

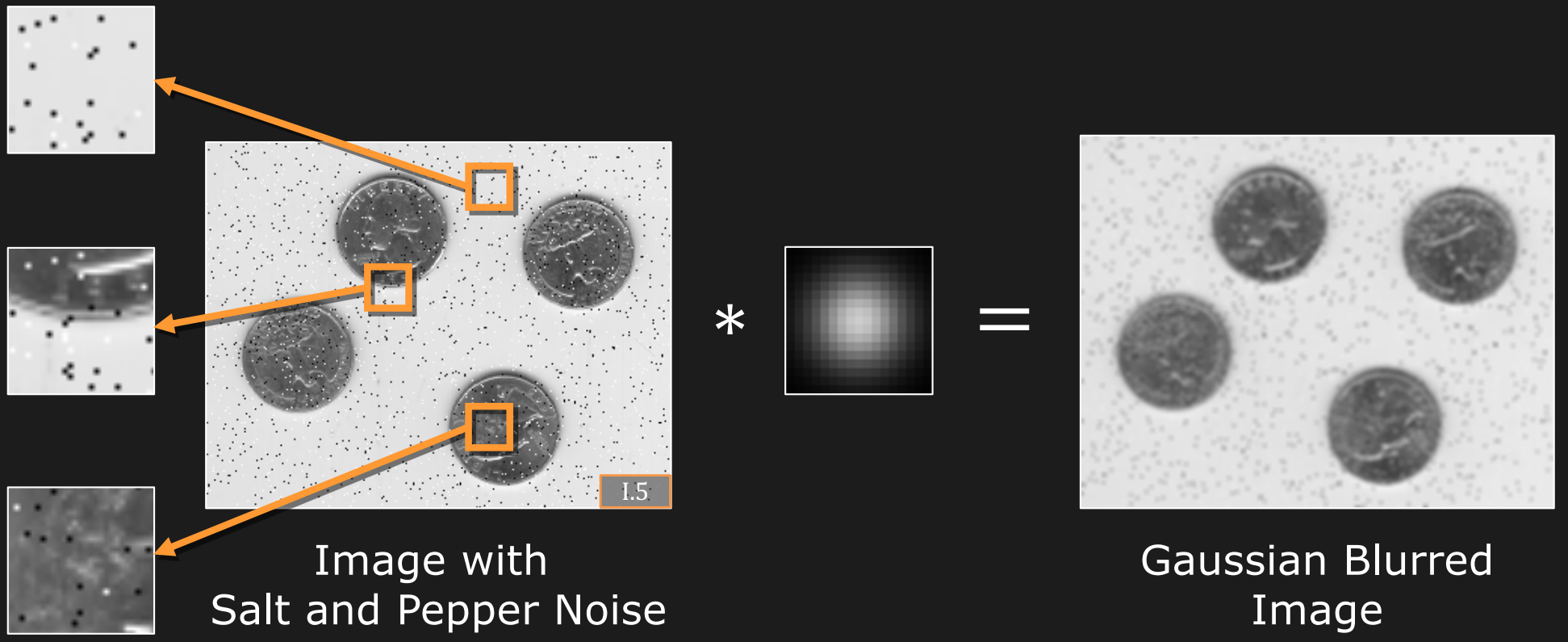


Image with
Salt and Pepper Noise

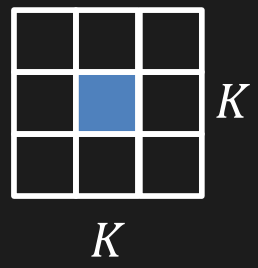
Gaussian Blurred
Image

Problem with Smoothing:

- Sensitive to Outliers (Noise)
- Smoothens Edges (Blur)

Median Filtering

1. Sort the K^2 values in window centered at the pixel
2. Assign the Middle value (Median) to pixel



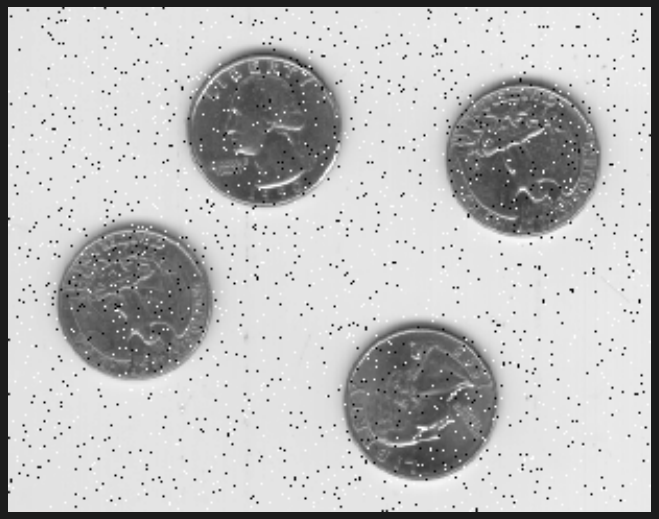


Image with
Salt and Pepper Noise

Median Filtered
Image ($K = 3$)

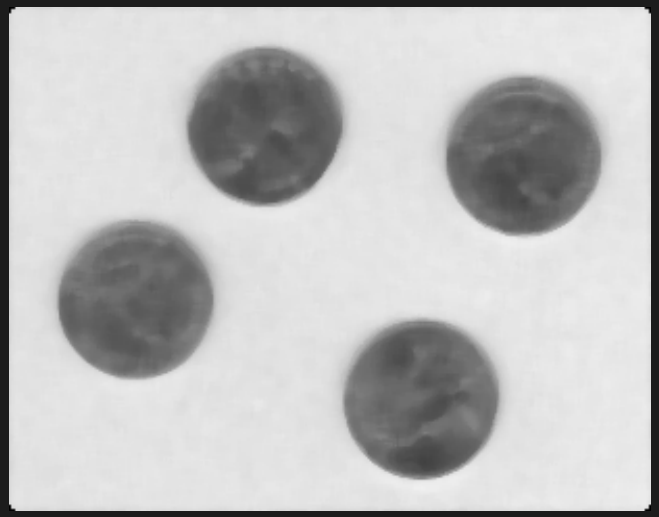
Non-linear Operation

(Cannot be implemented using Convolution)

Median Filtering

Not Effective when Image Noise is not a Simple Salt and Pepper Noise.

Image with Noise



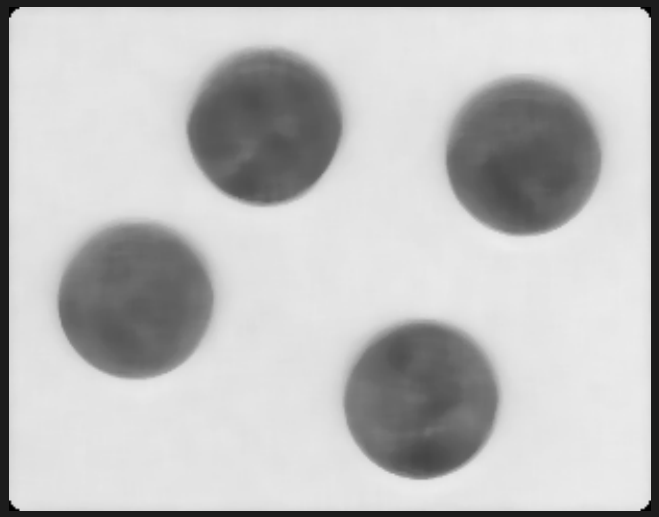
Median Filtered Image ($K = 7$)

Larger K causes Blurring of Image Detail

Median Filtering

Not Effective when Image Noise is not a Simple Salt and Pepper Noise.

Image with Noise



Median Filtered Image ($K = 11$)

Larger K causes Blurring of Image Detail

Summary

Convolution: A linear operator that allows us to transform one image to another

Essential concepts in this lecture:

Correlation vs. Convolution
Nonlinear filters that suppress noise

References: Textbooks

Computer Vision: Algorithms and Appications (Chapter 3.4)
Recommended Reading
Szelinski, 2011 (available online)

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 3, 4)
Horn, B. K. P., MIT Press

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

I.5 Matlab Demo Image

I.6 Adapted from http://1x.com/photo/24072

I.7 http://cdn04.cdn.socialitelife.com/wp-content/uploads/2009 /11/stars-without-makeup-photos-11062009-07.jpg