Image Processing II

Computer Vision: AI3604

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

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) Binary Image Processing

Lecture 2

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

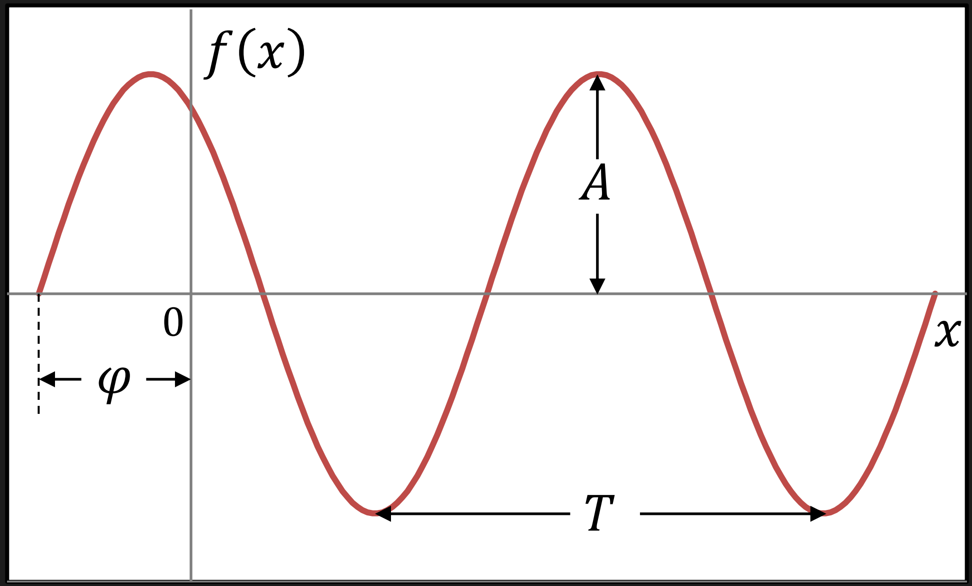
Jean Baptiste Joseph Fourier

(1768-1830)

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

Sinusoid

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



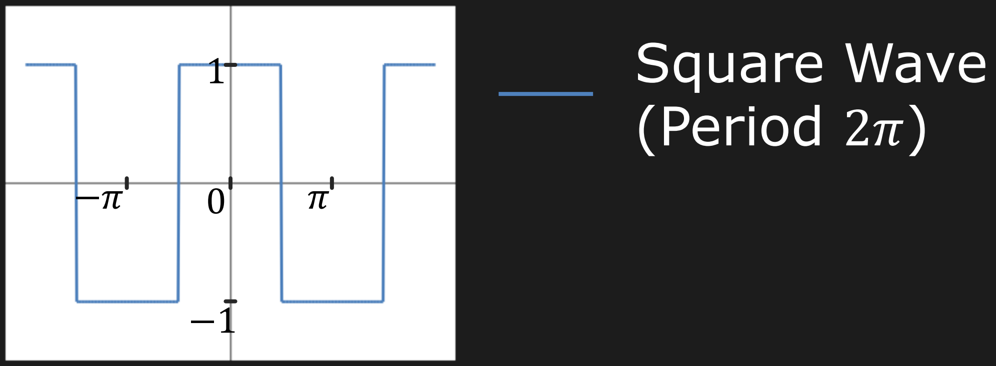
A: Amplitude

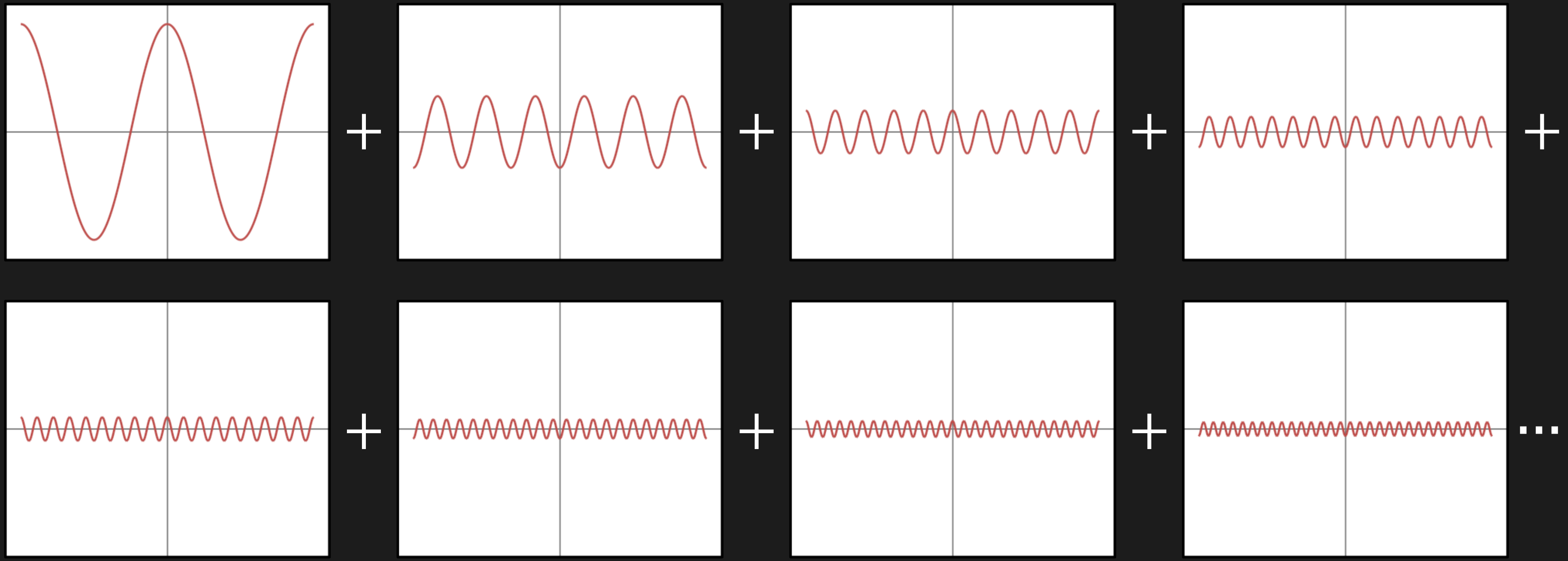
T: Period

φ: Phase

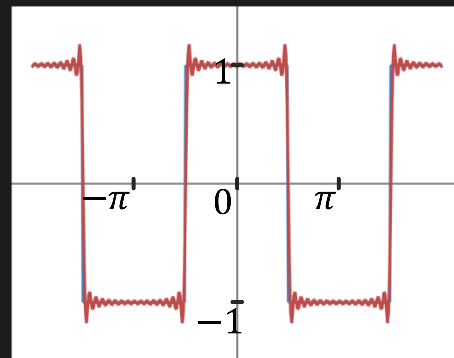
u: Frequency $(1/T)$

Fourier Series



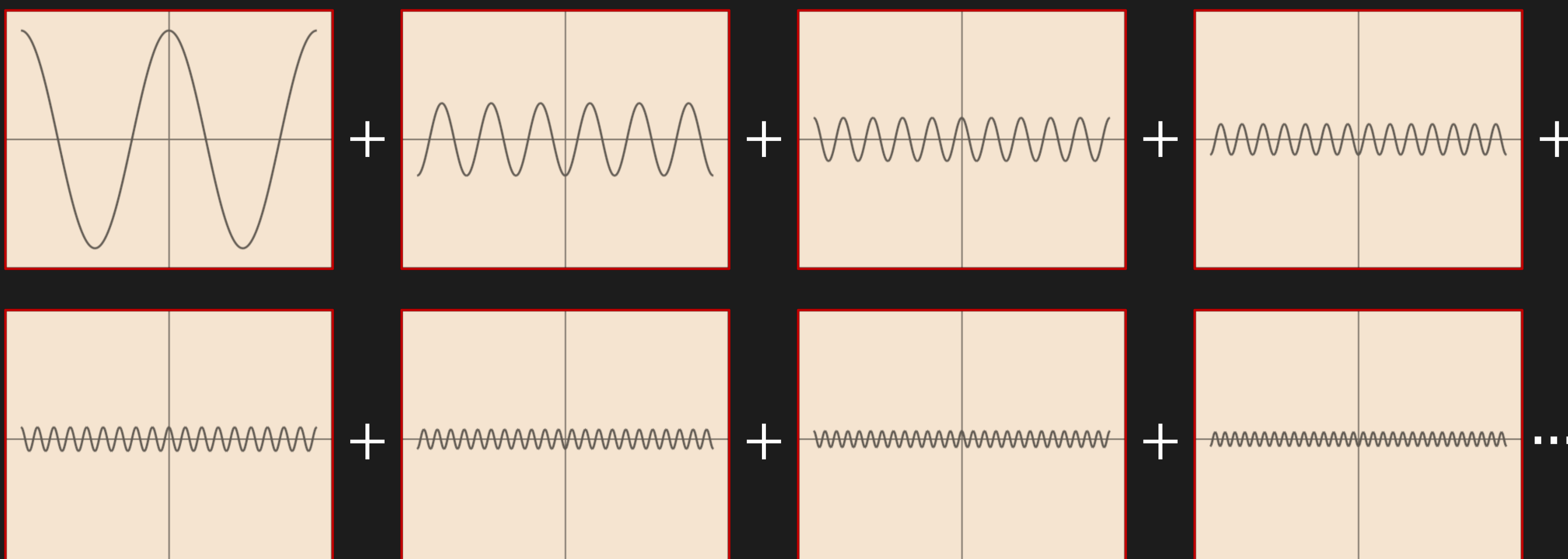


Fourier Series

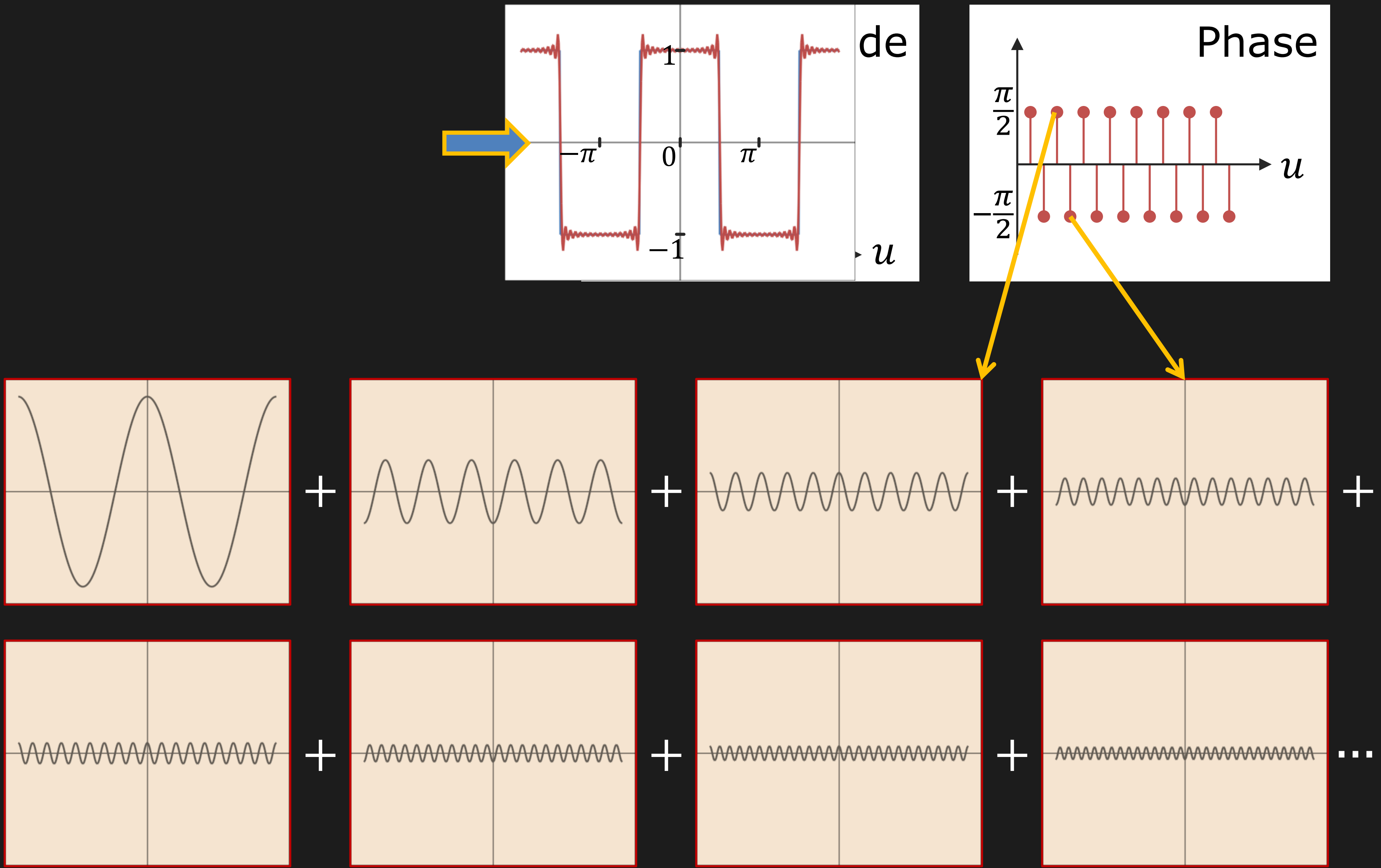


Square Wave (Period 2π)

Sum of Sinusoids



An Alternate Representation of Signal



Sinusoid

Orthogonal bases

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

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

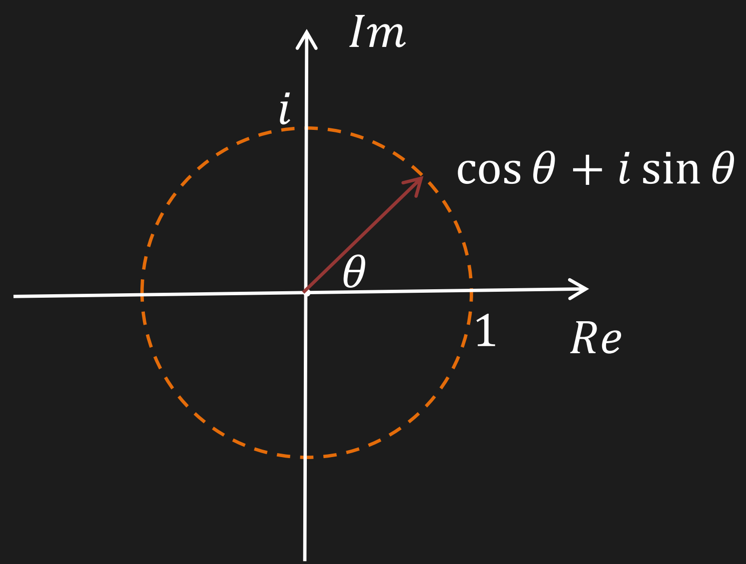
$$\int_{-\pi}^{\pi} \cos nx \cdot \cos mx \, dx = 0 \; (n \neq m)$$

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

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

Exponential Sinusoid (Euler Formula)

What if the function is not periodic?



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

$$i = \sqrt{-1}$$

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:

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

Fourier Transform is Complex!

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

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

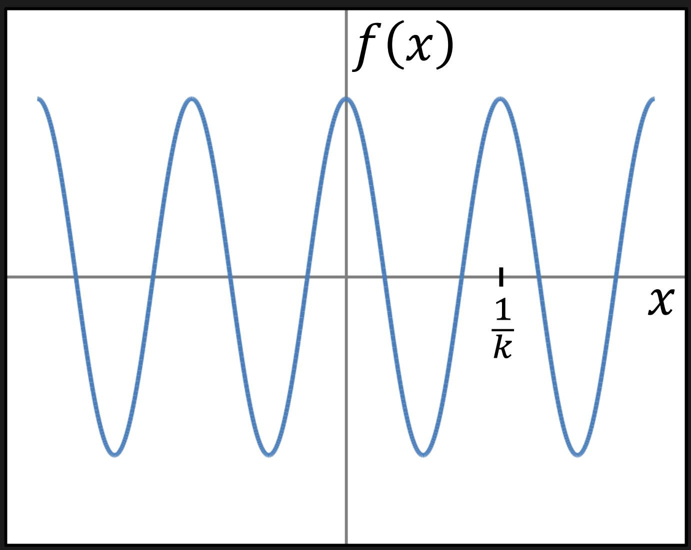
$$F(u) = \Re\{F(u)\} + i\,\Im\{F(u)\}$$

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

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

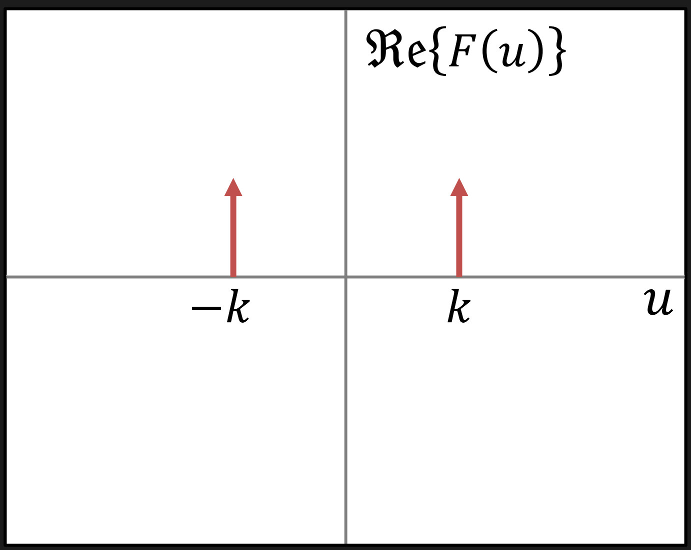
Fourier Transform Examples

Signal $f(x)$



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

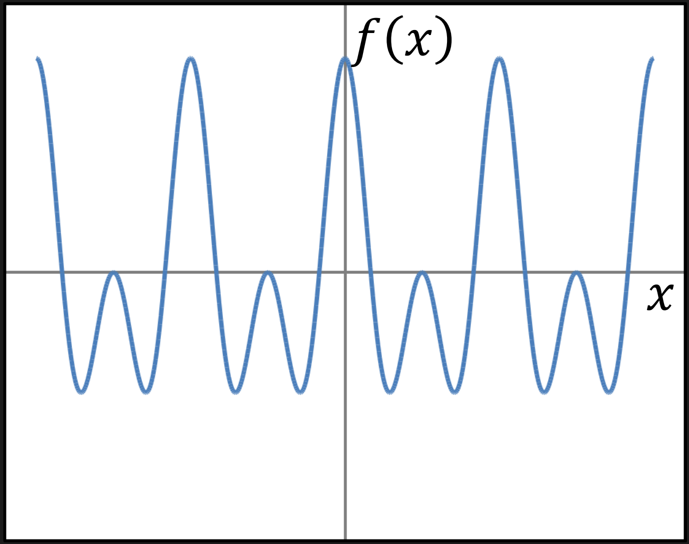
Fourier Transform $F(u)$



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

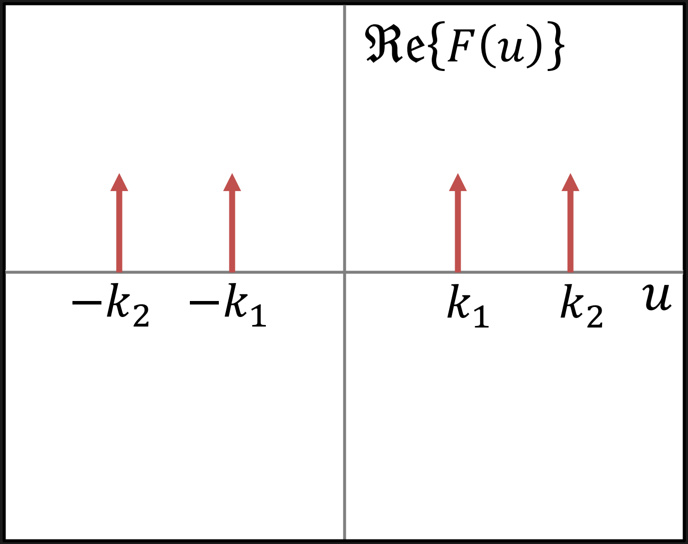
Fourier Transform Examples

Signal $f(x)$



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

Fourier Transform $F(u)$

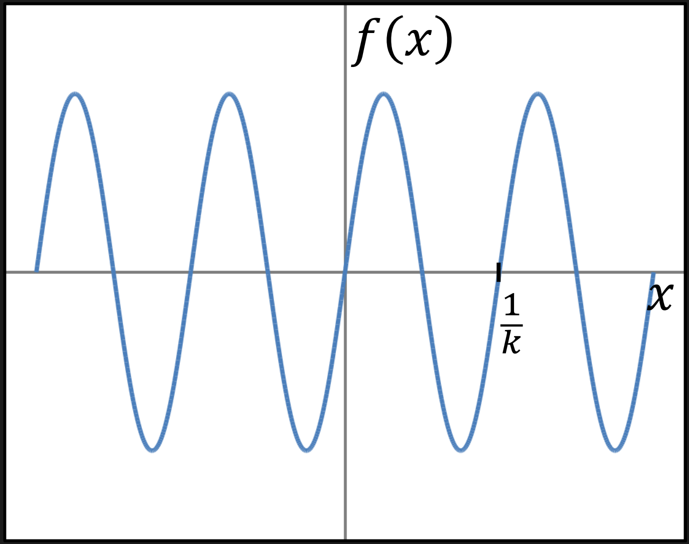


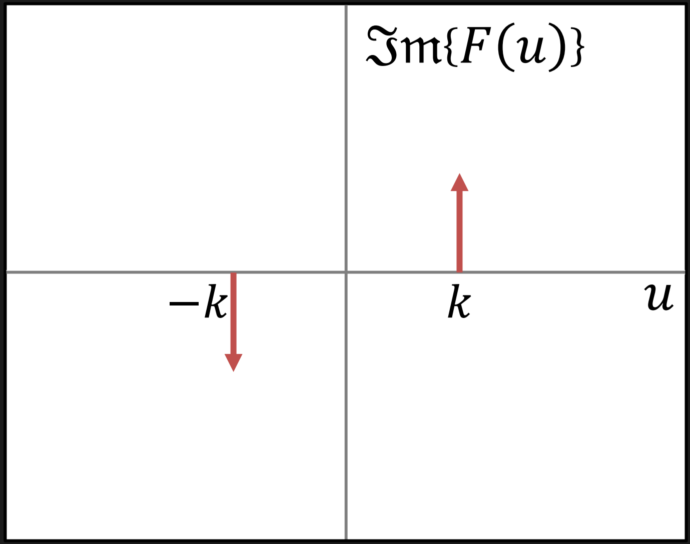
$$F(u) = \frac{1}{2}[\delta(u + k_1) + \delta(u - k_1) + \delta(u + k_2) + \delta(u - k_2)]$$

Fourier Transform Examples

Signal $f(x)$

Fourier Transform $F(u)$



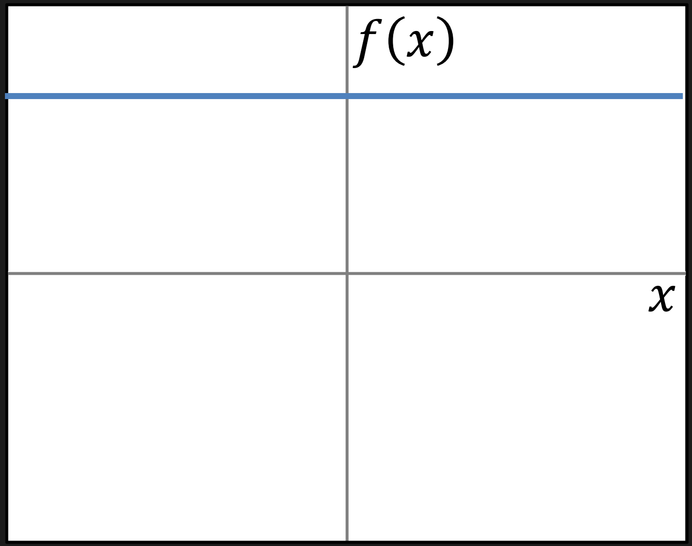


$$f(x) = \sin 2\pi kx$$

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

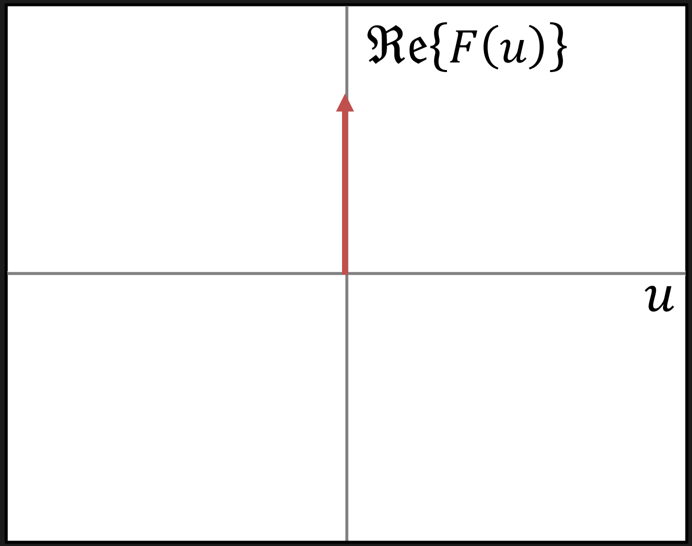
Fourier Transform Examples

Signal $f(x)$



$$f(x) = 1$$

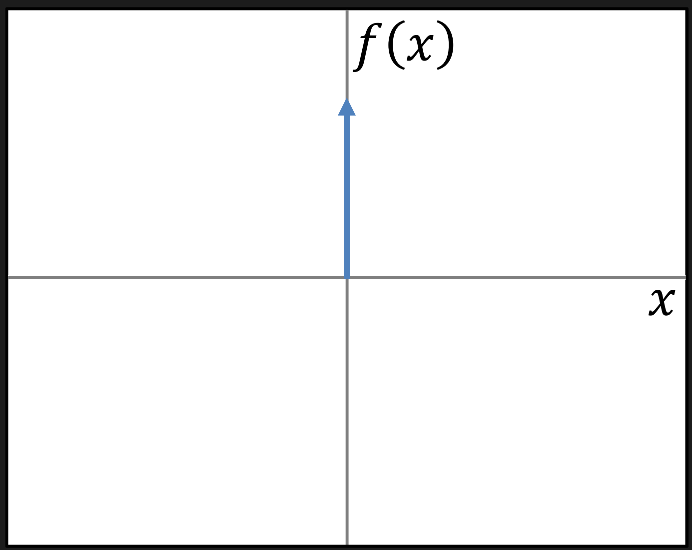
Fourier Transform $F(u)$



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

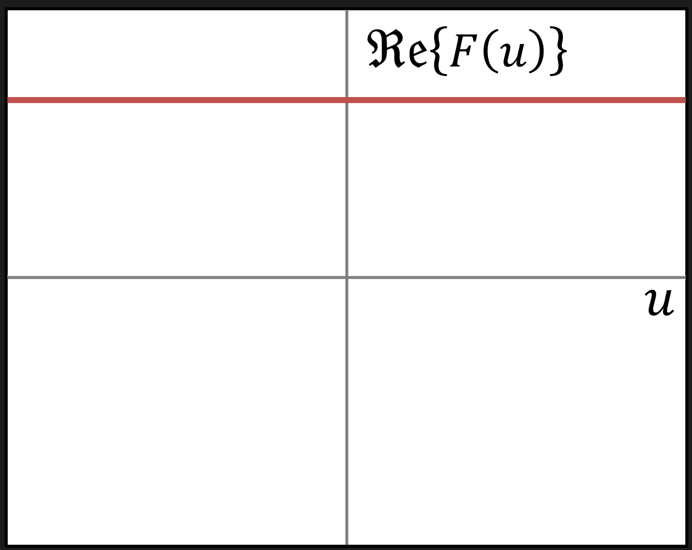
Fourier Transform Examples

Signal $f(x)$



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

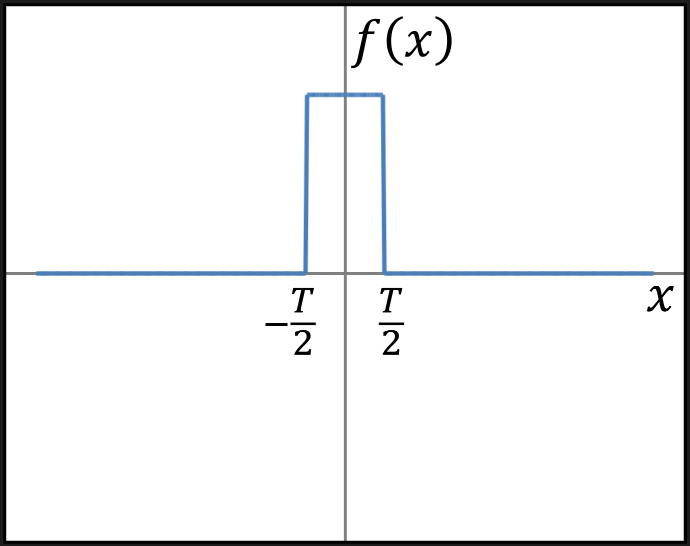
Fourier Transform $F(u)$



$$F(u) = 1$$

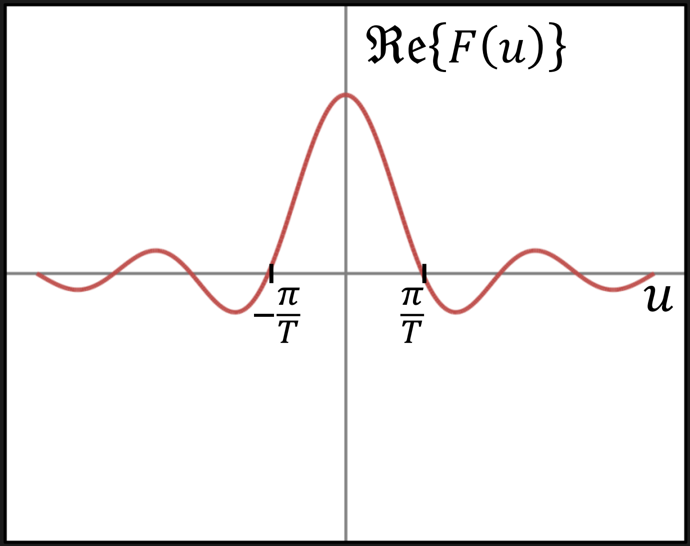
Fourier Transform Examples

Signal $f(x)$



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

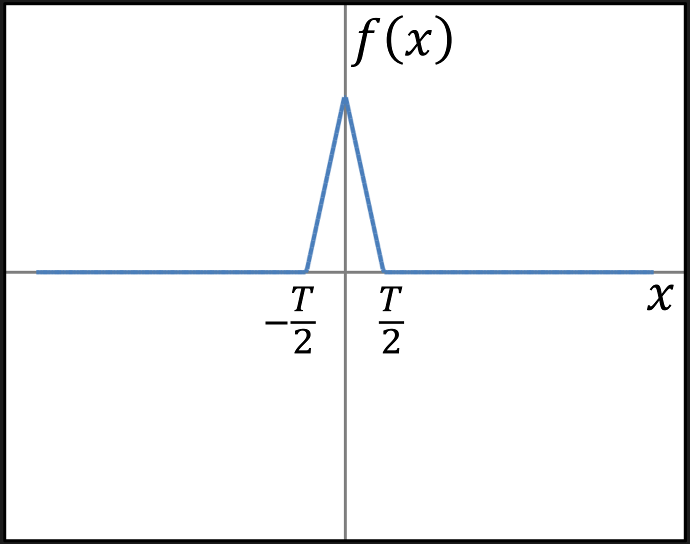
Fourier Transform $F(u)$



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

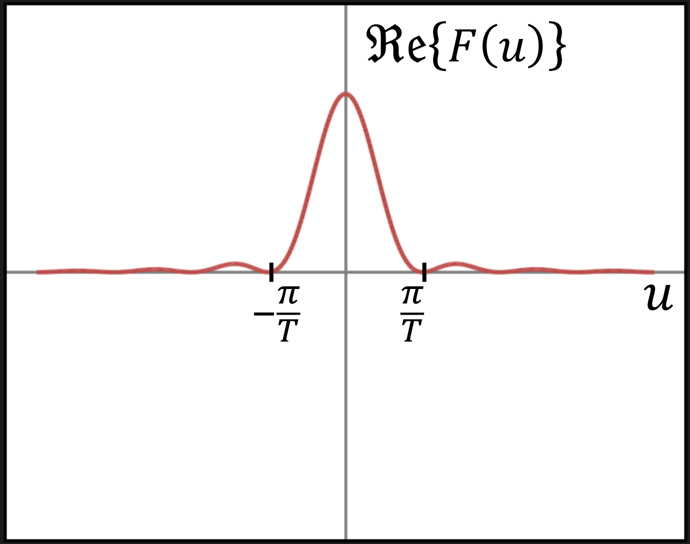
Fourier Transform Examples

Signal $f(x)$



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

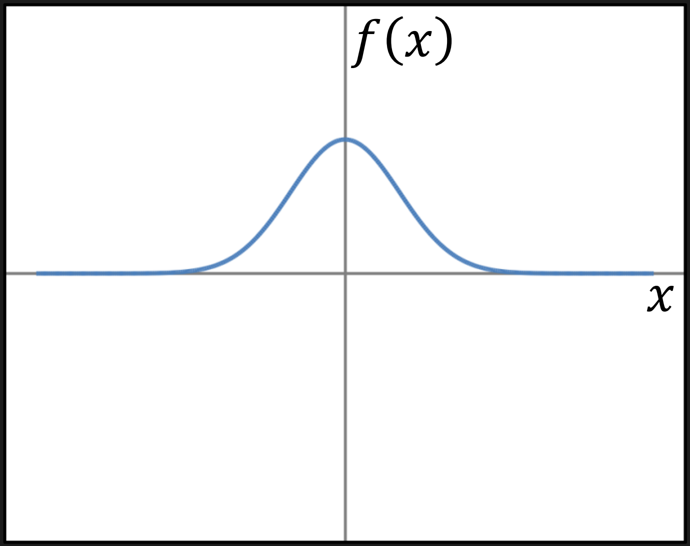
Fourier Transform $F(u)$



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

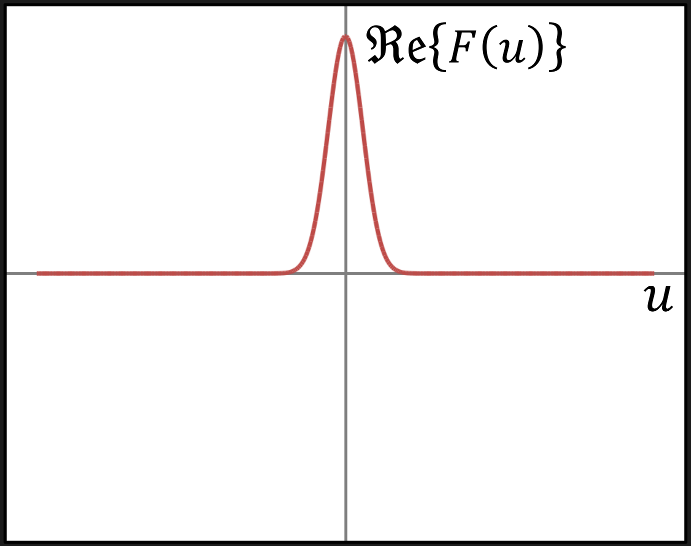
Fourier Transform Examples

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

The Convolution Theorem

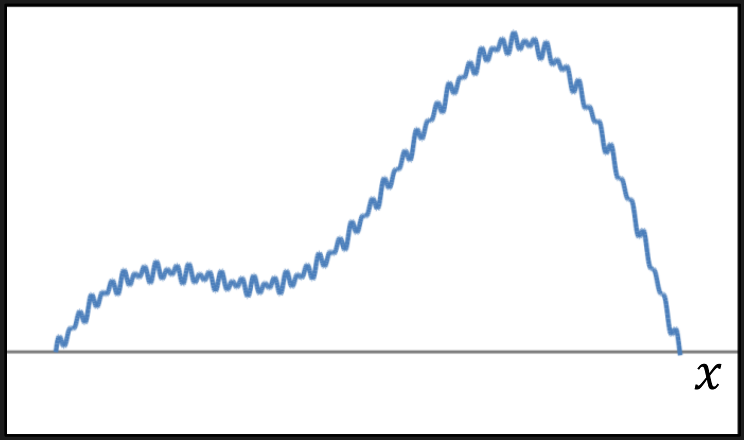
Properties of Fourier Transform

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

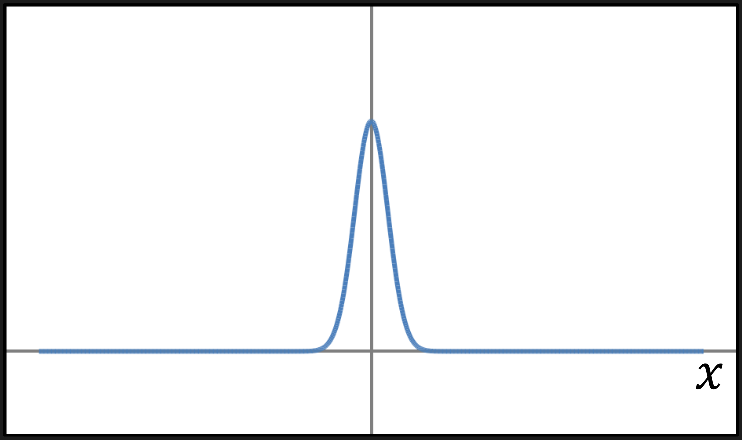
$$\begin{tikzcd}
g(x) & = & f(x) \arrow[d, "\boxed{\mathrm{FT}}"] & * & h(x) \arrow[d, "\boxed{\mathrm{FT}}"] \\
G(u) \arrow[u, "\boxed{\mathrm{IFT}}"] & = & F(u) & \times & H(u)
\end{tikzcd}$$

Gaussian Smoothing in Fourier Domain



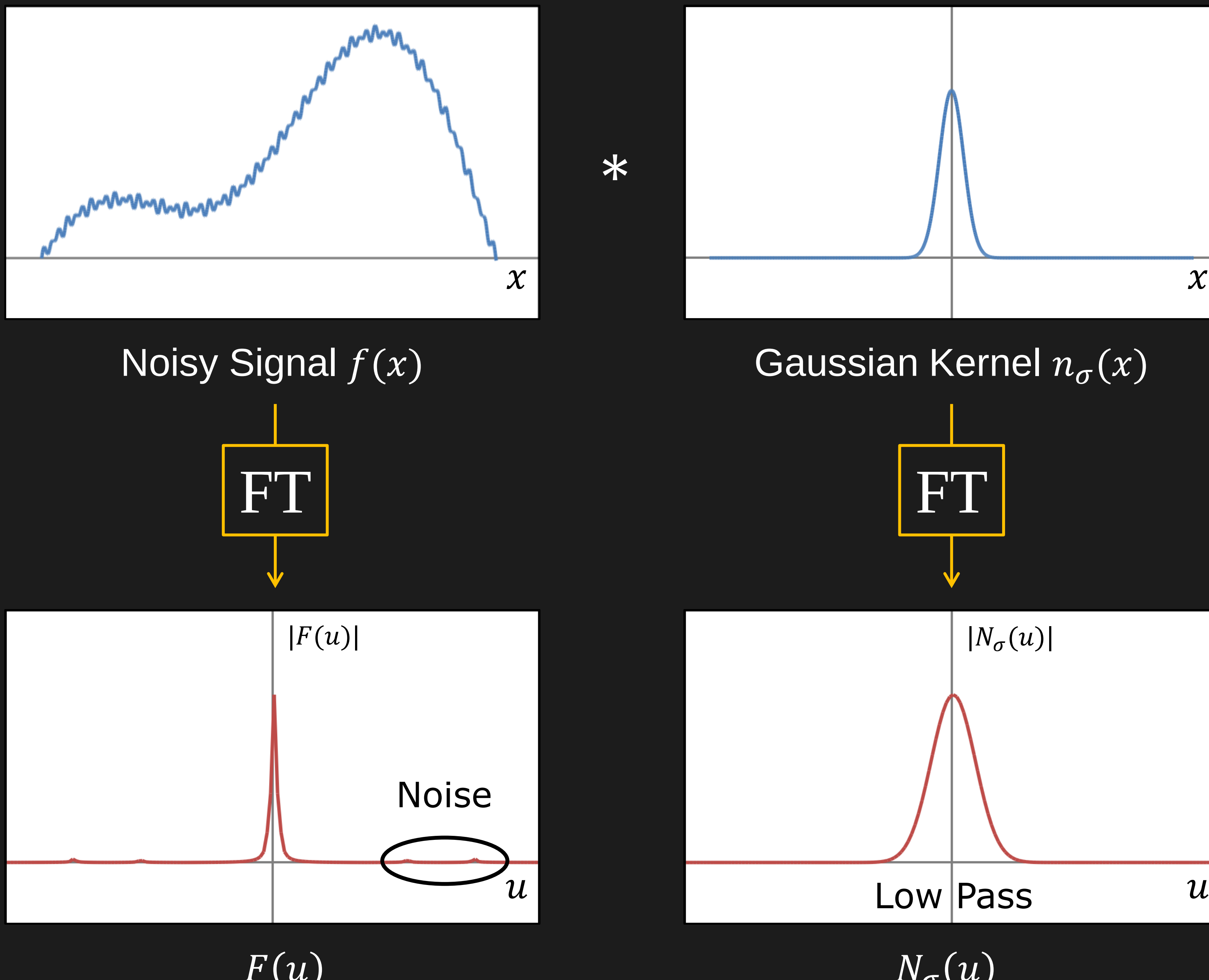
Noisy Signal $f(x)$

$*$

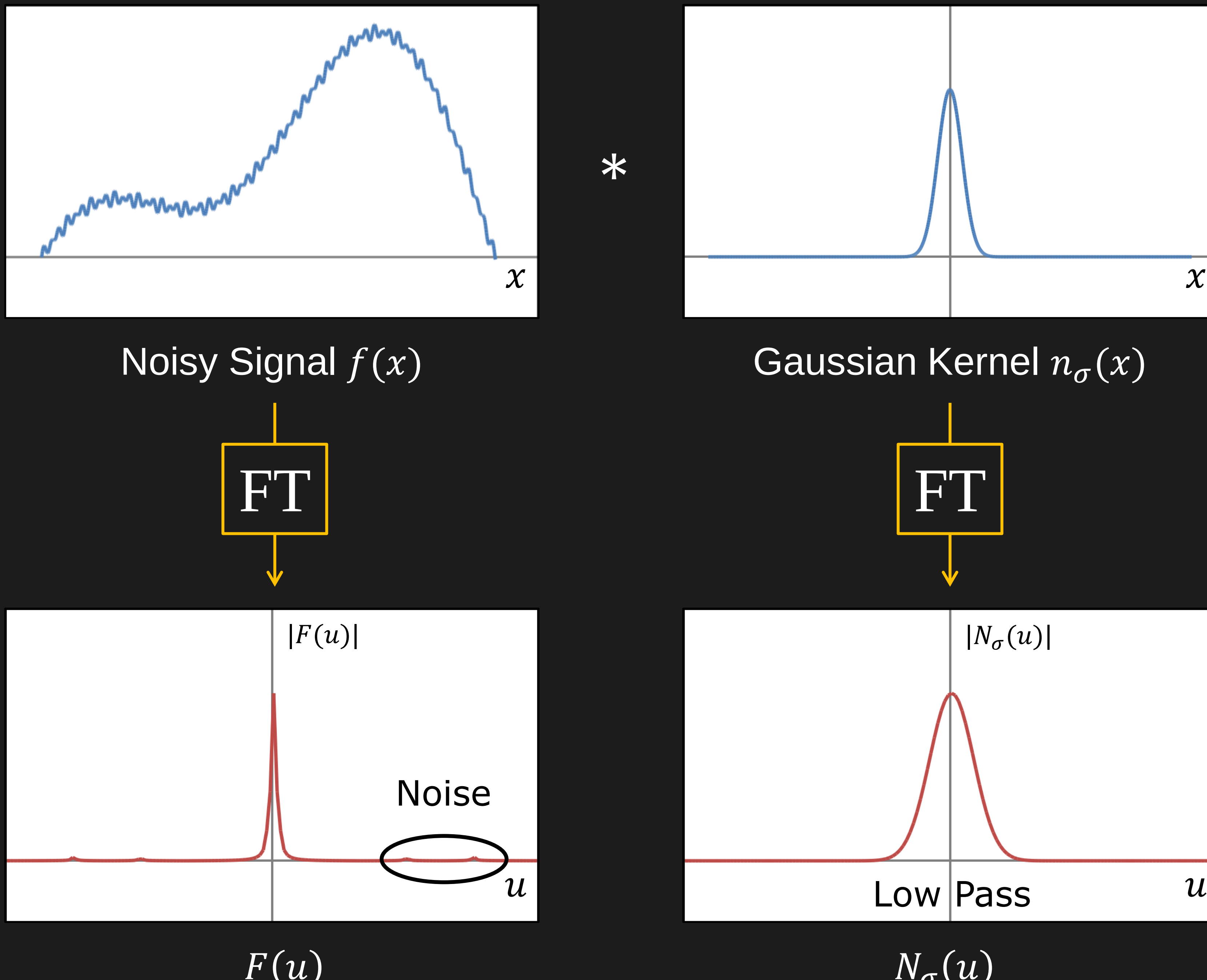


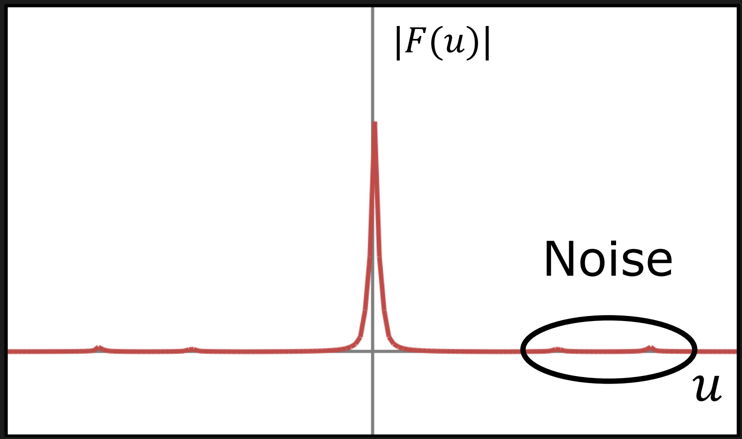
Gaussian Kernel $n_\sigma(x)$

Convolve the Noisy Signal with a Gaussian Kernel

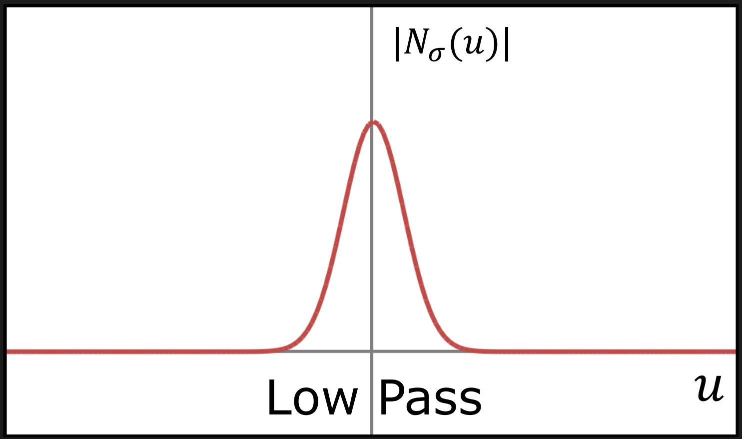


x
Noisy Signal $f(x)$
*
x
Gaussian Kernel $n_\sigma(x)$
FT
FT
$|F(u)|$
Noise
u
$F(u)$
$|N_\sigma(u)|$
Low
Pass
u
$N_\sigma(u)$

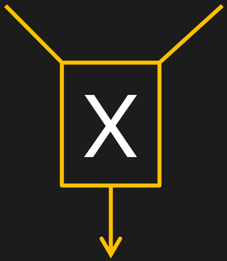

x
*
x
Noisy Signal $f(x)$
Gaussian Kernel $n_\sigma(x)$
FT
FT
$|F(u)|$
Noise
u
$|N_\sigma(u)|$
Low
Pass
u
$F(u)$
$N_\sigma(u)$

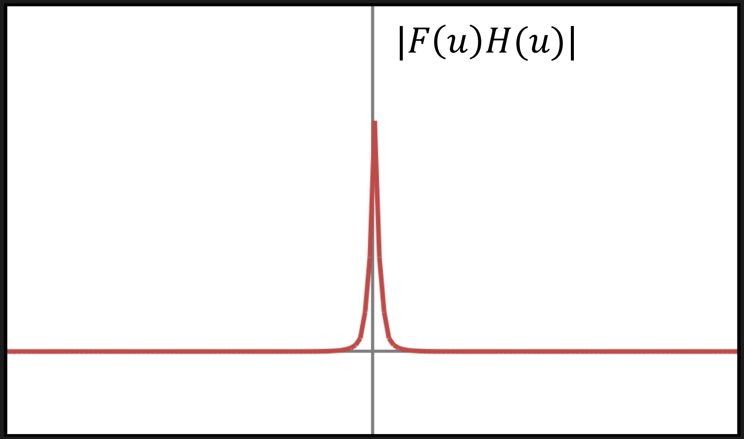


$F(u)$

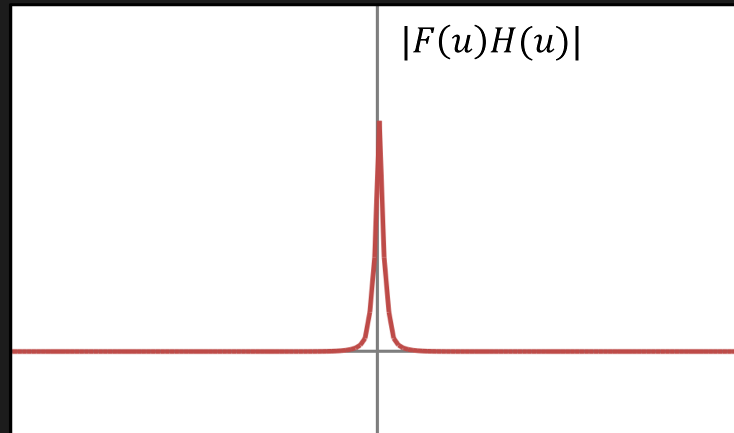


$N_\sigma(u)$



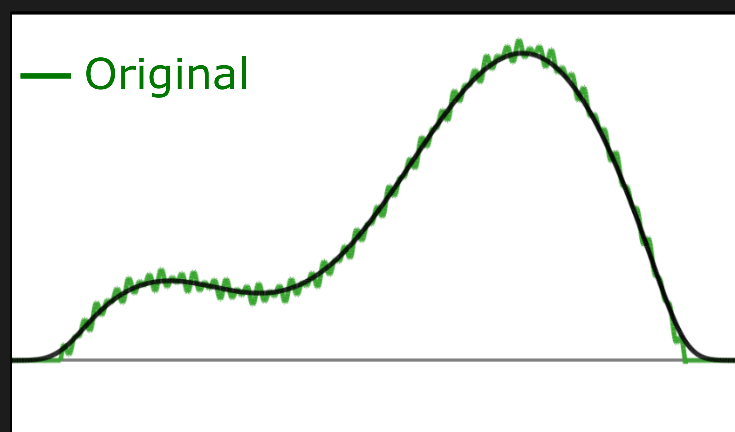


$F(u)H(u)$



$F(u)H(u)$

IFT



Gaussian Blurred Signal $g(x)$

2D Fourier Transform

Fourier Transform:

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

u and v are frequencies along x and y, respectively

Inverse Fourier Transform:

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

2D Fourier Transform: Discrete Images

Discrete Fourier Transform (DFT):

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

$p = 0 \dots M-1$

$q = 0 \dots N-1$

p and q are frequencies along m and n, respectively

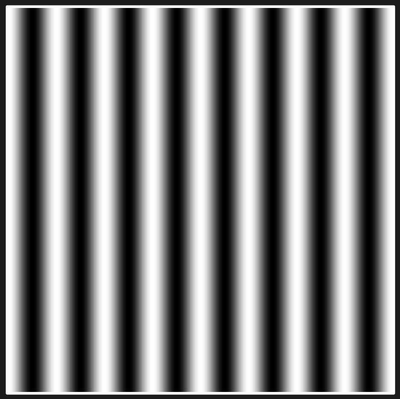
Inverse Discrete Fourier Transform (IDFT):

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

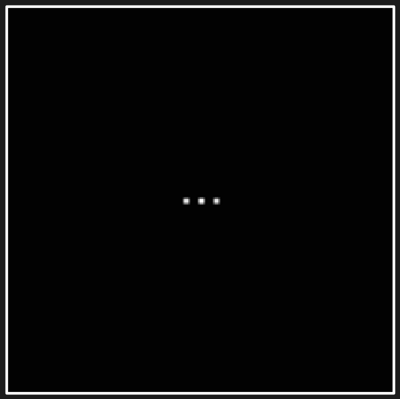
$m = 0 \dots M-1$

$n = 0 \dots N-1$

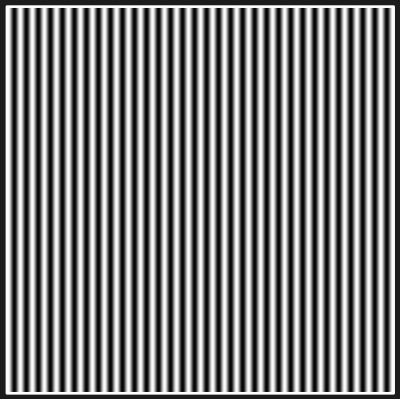
2D Fourier Transform: Example 1



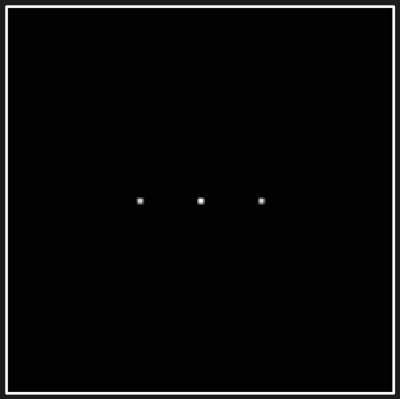
$f(m,n)$



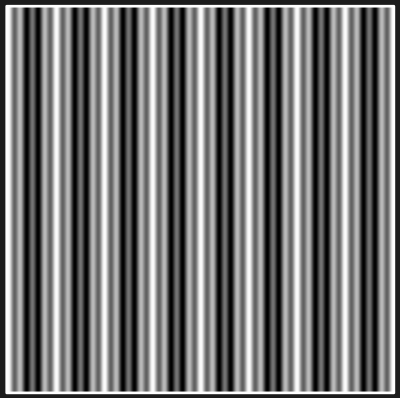
$\log(|F(p,q)|)$



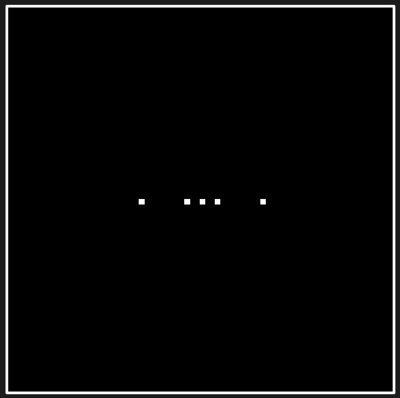
$g(m,n)$



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



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

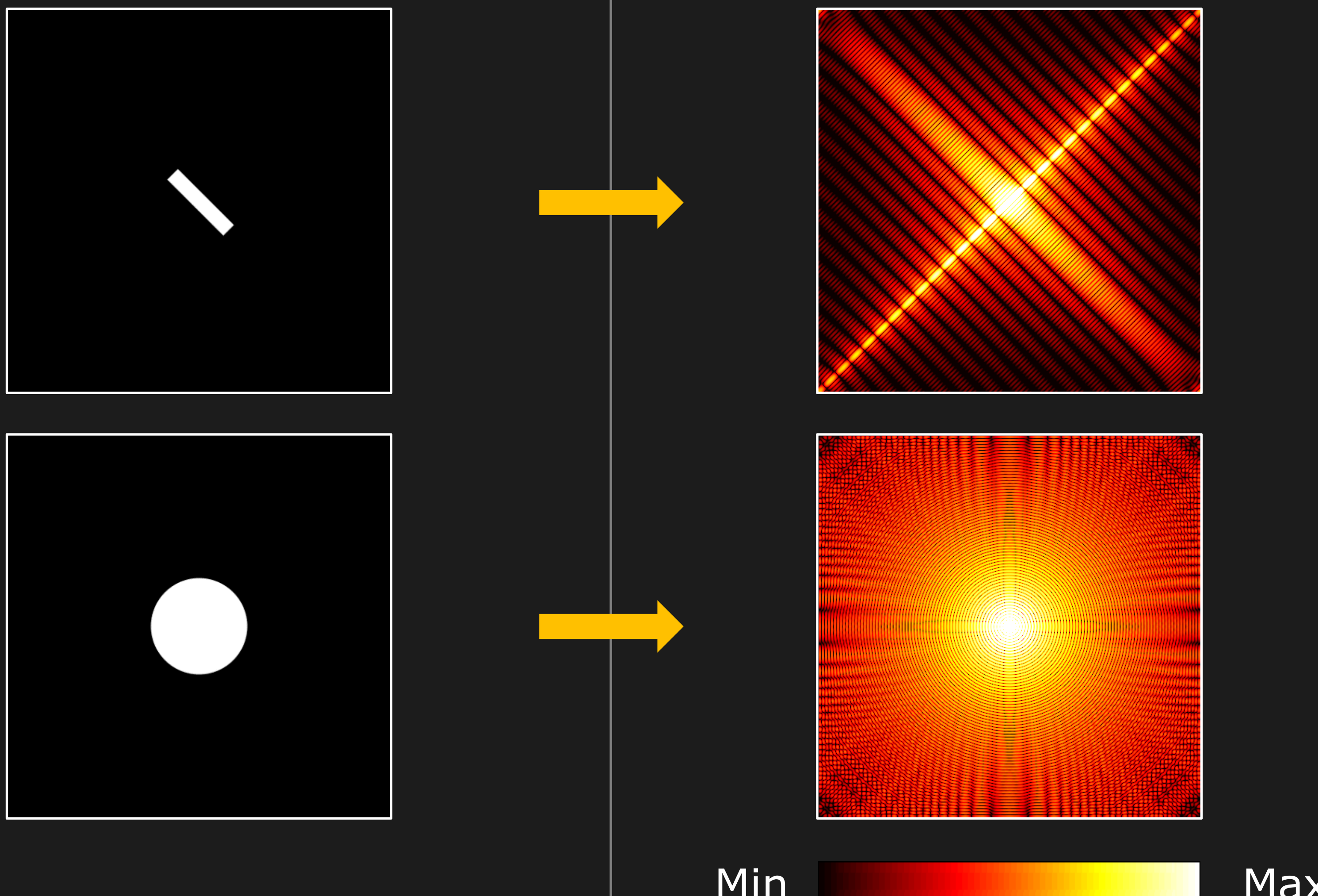


$\log(|F(p,q) + G(p,q)|)$

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

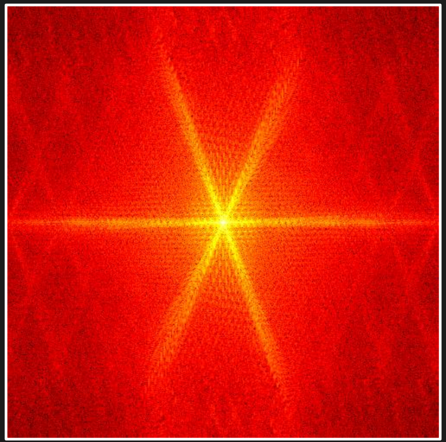
2D Fourier Transform: Example 2

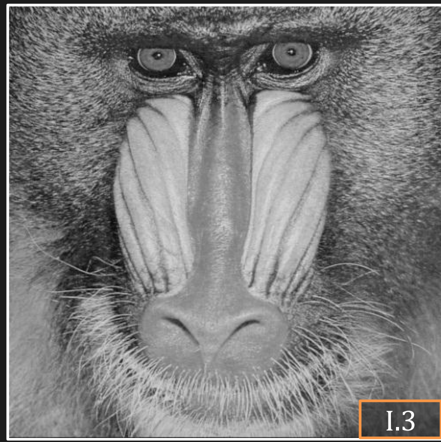
$f(m,n)$ $\quad$ $\log(|F(p,q)|)$

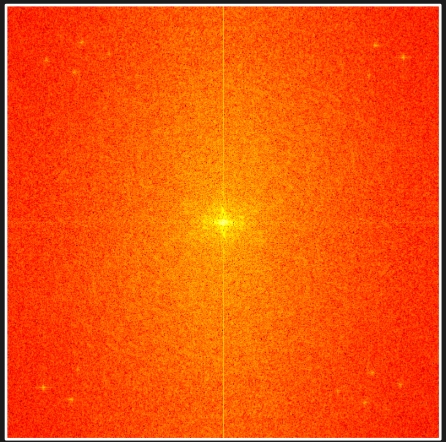


2D Fourier Transform: Example 3

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



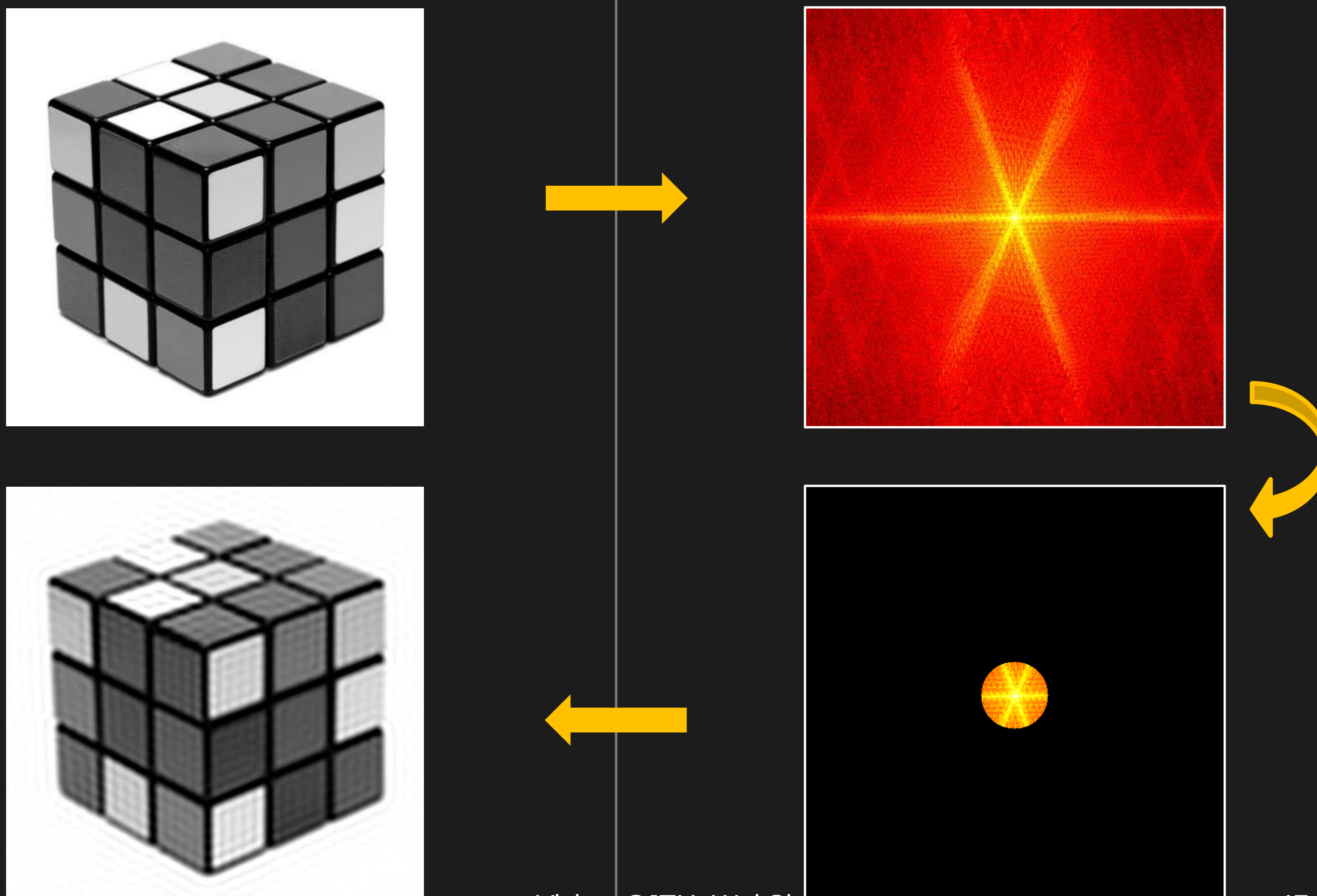




Min  Max

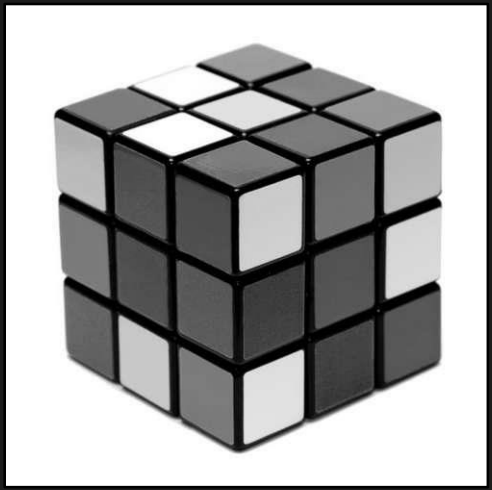
Low Pass Filtering

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

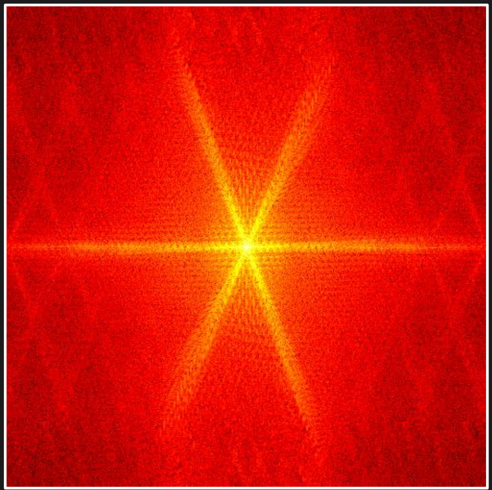


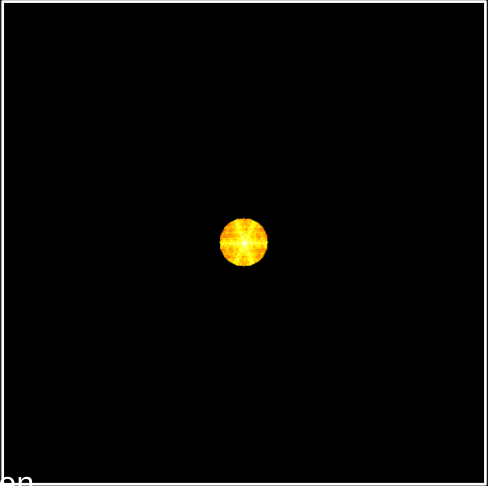
Low Pass Filtering

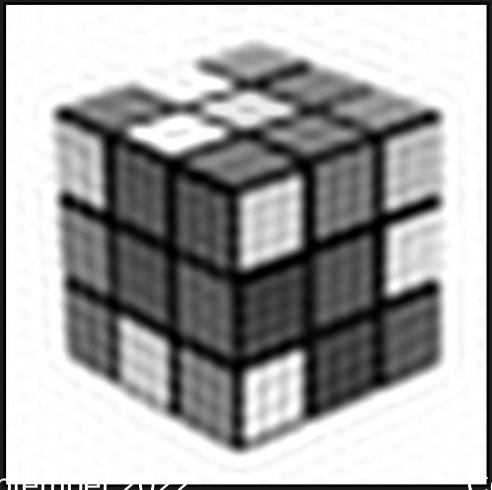
$f(m,n)$



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



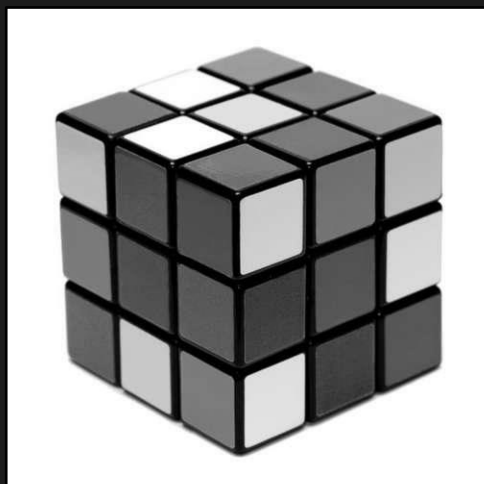


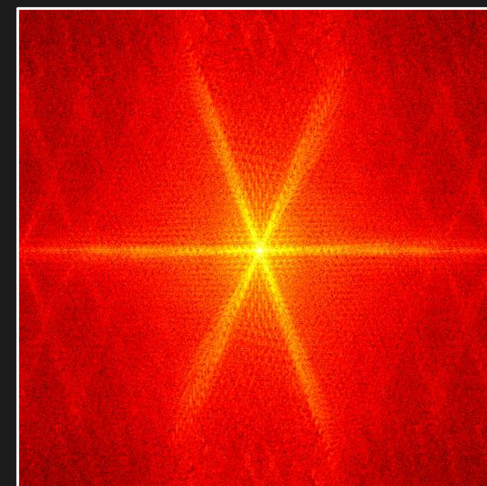


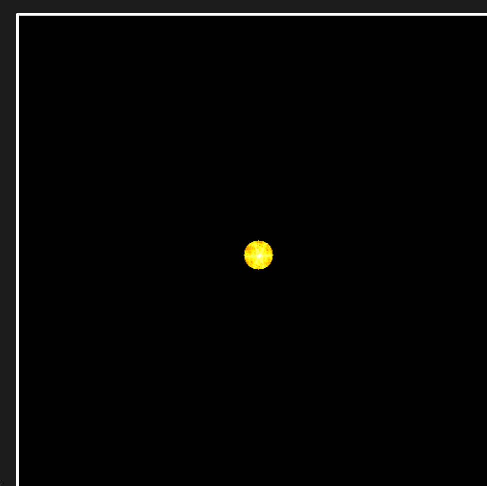
Low Pass Filtering

$f(m,n)$

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



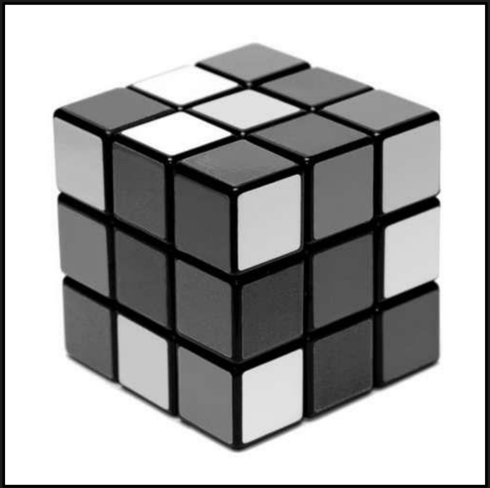


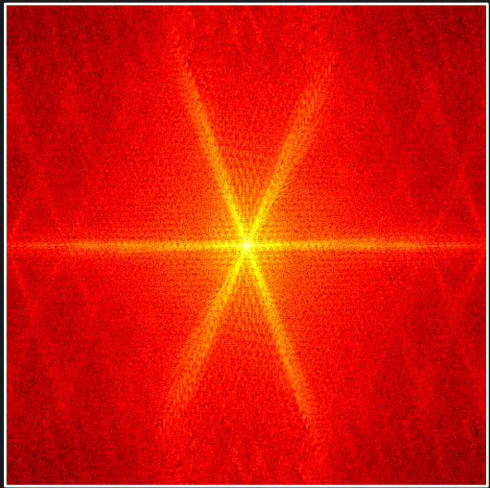


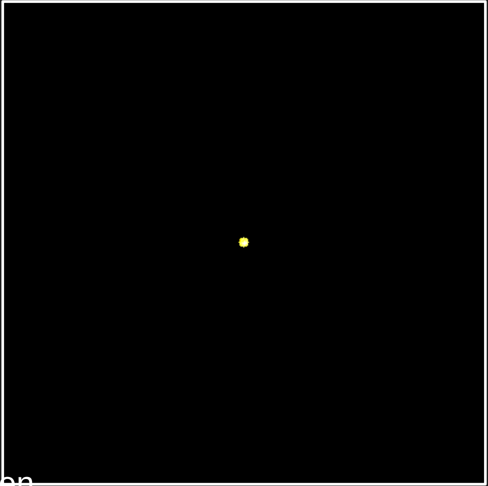
Low Pass Filtering

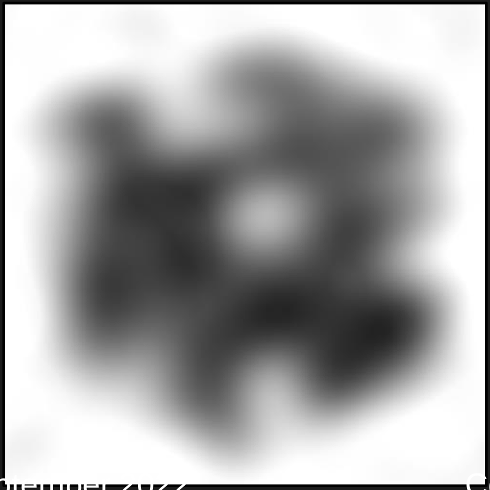
$f(m,n)$

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



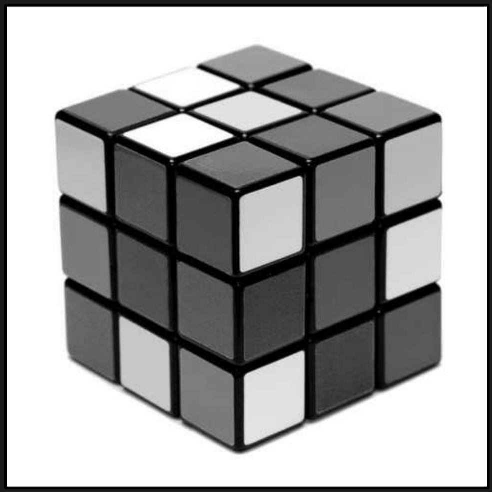




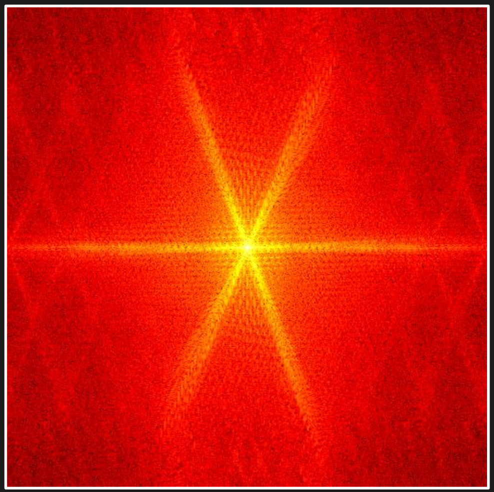


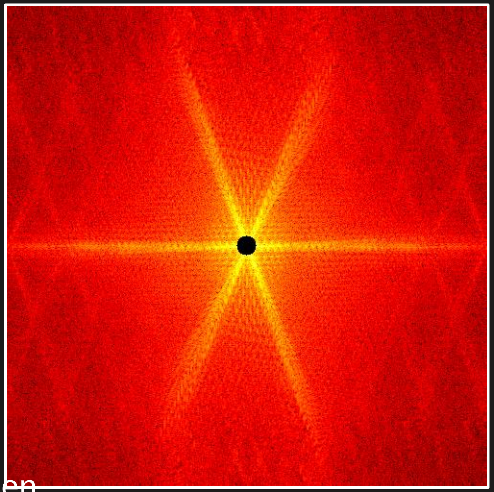
High Pass Filtering

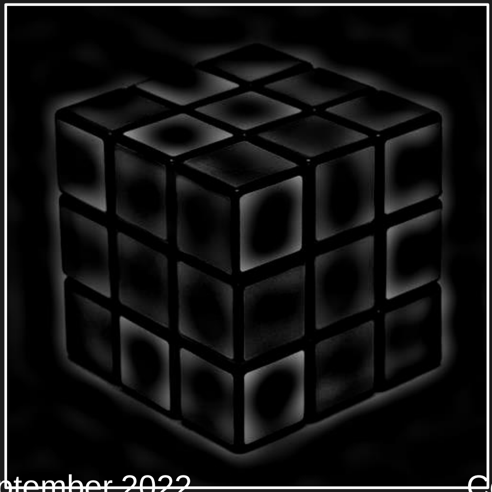
$f(m,n)$



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

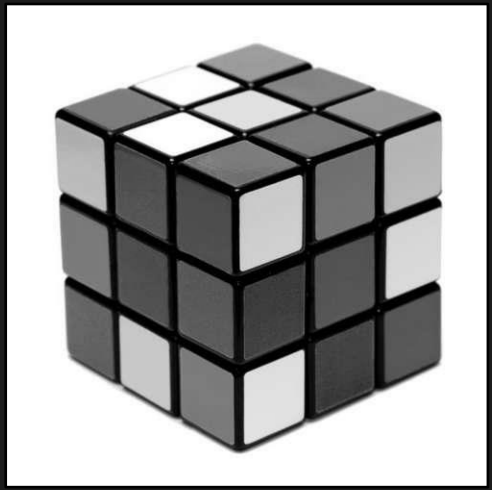




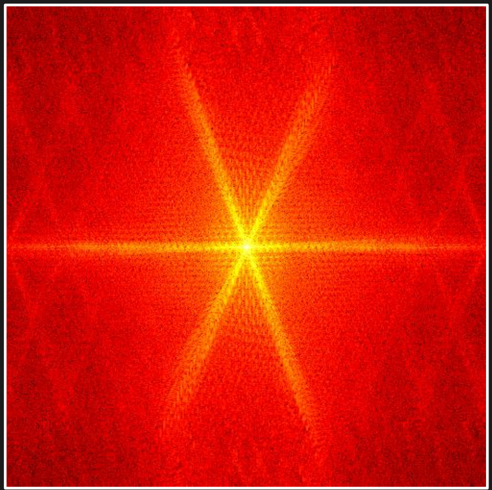


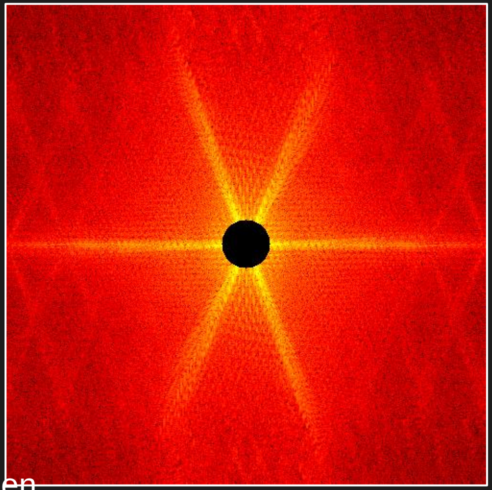
High Pass Filtering

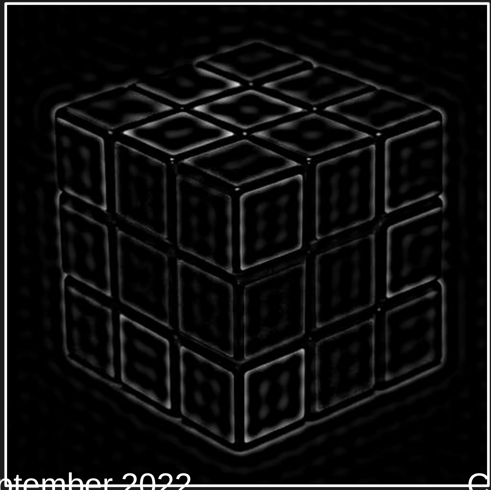
$f(m,n)$



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



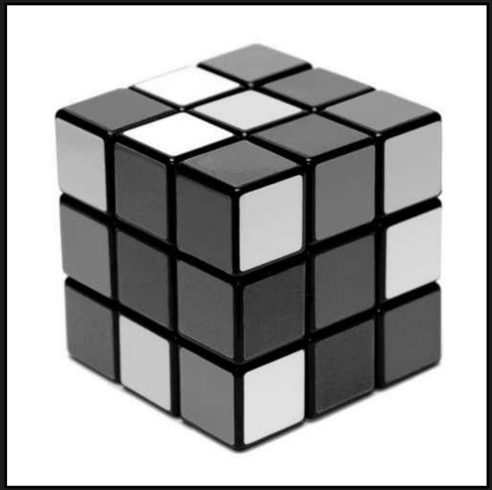


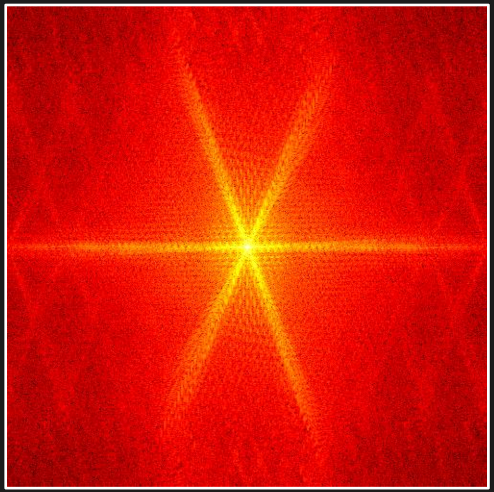


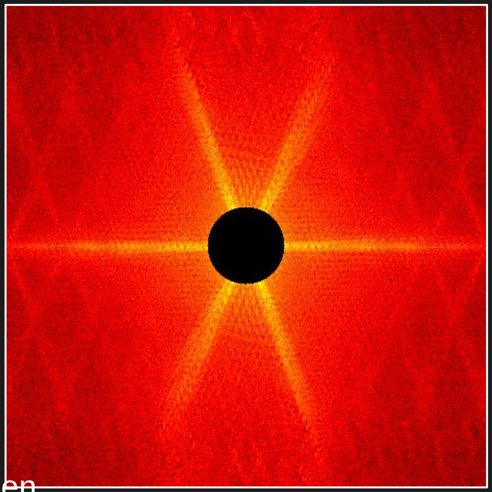
High Pass Filtering

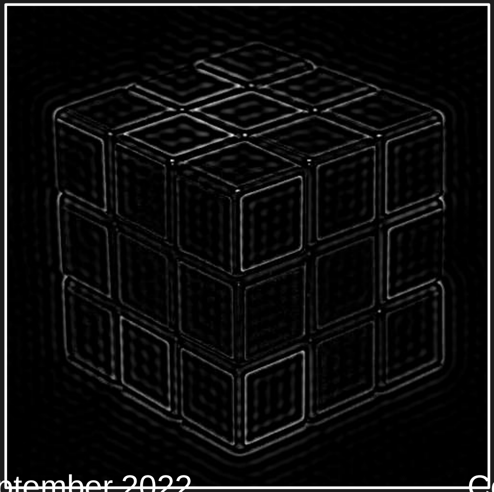
$f(m,n)$

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





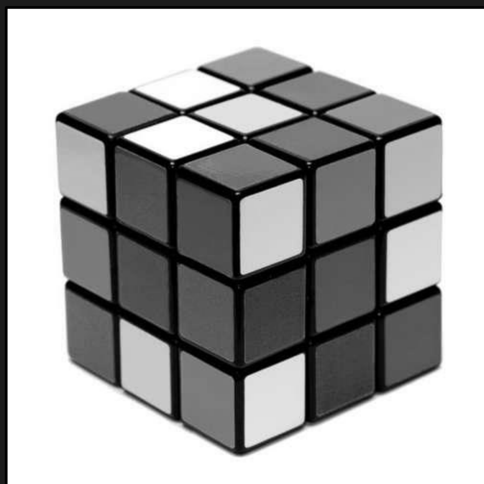


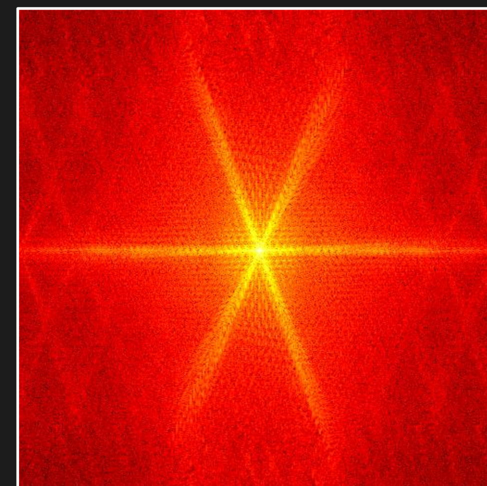


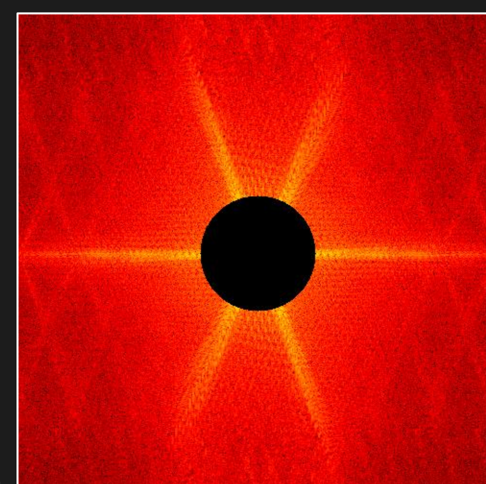
High Pass Filtering

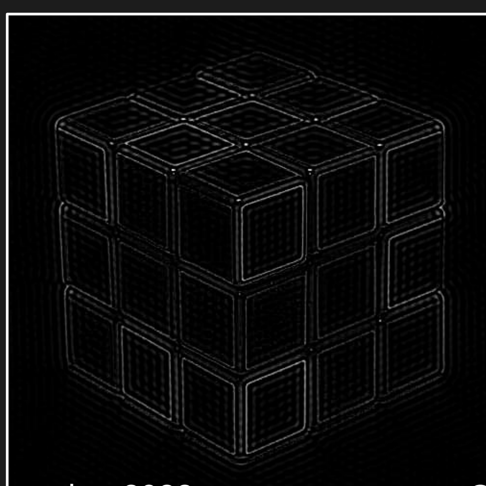
$f(m,n)$

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

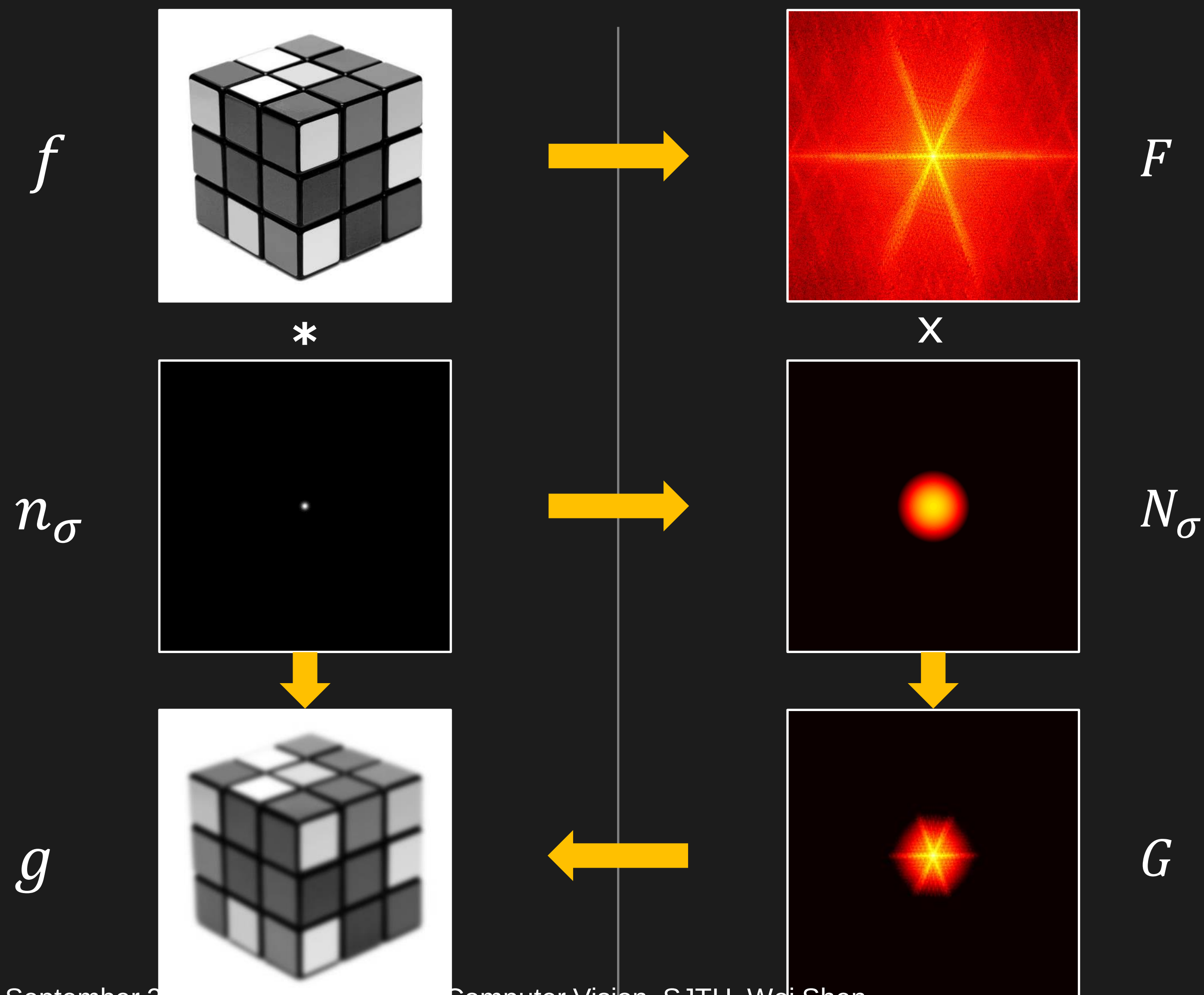




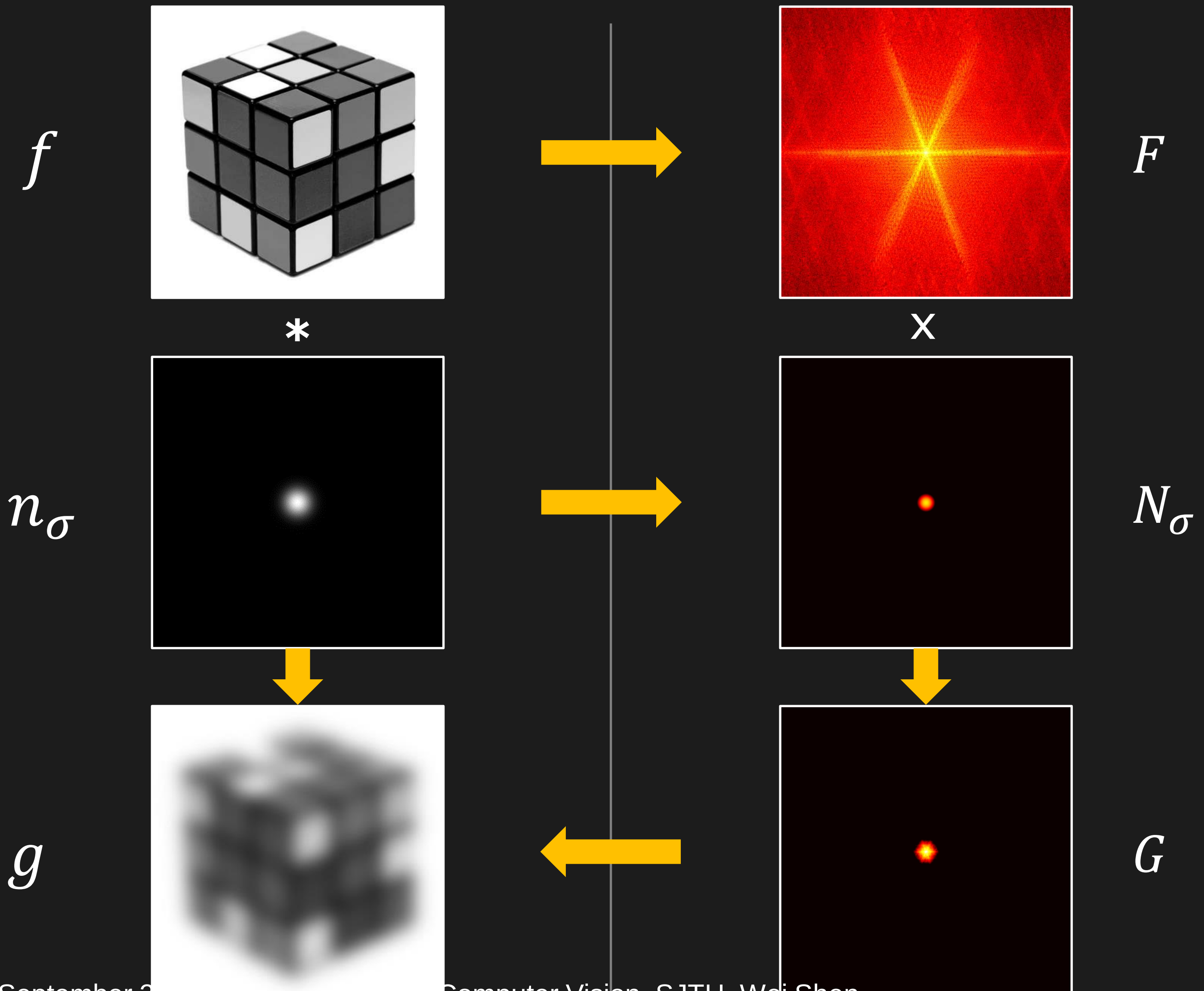




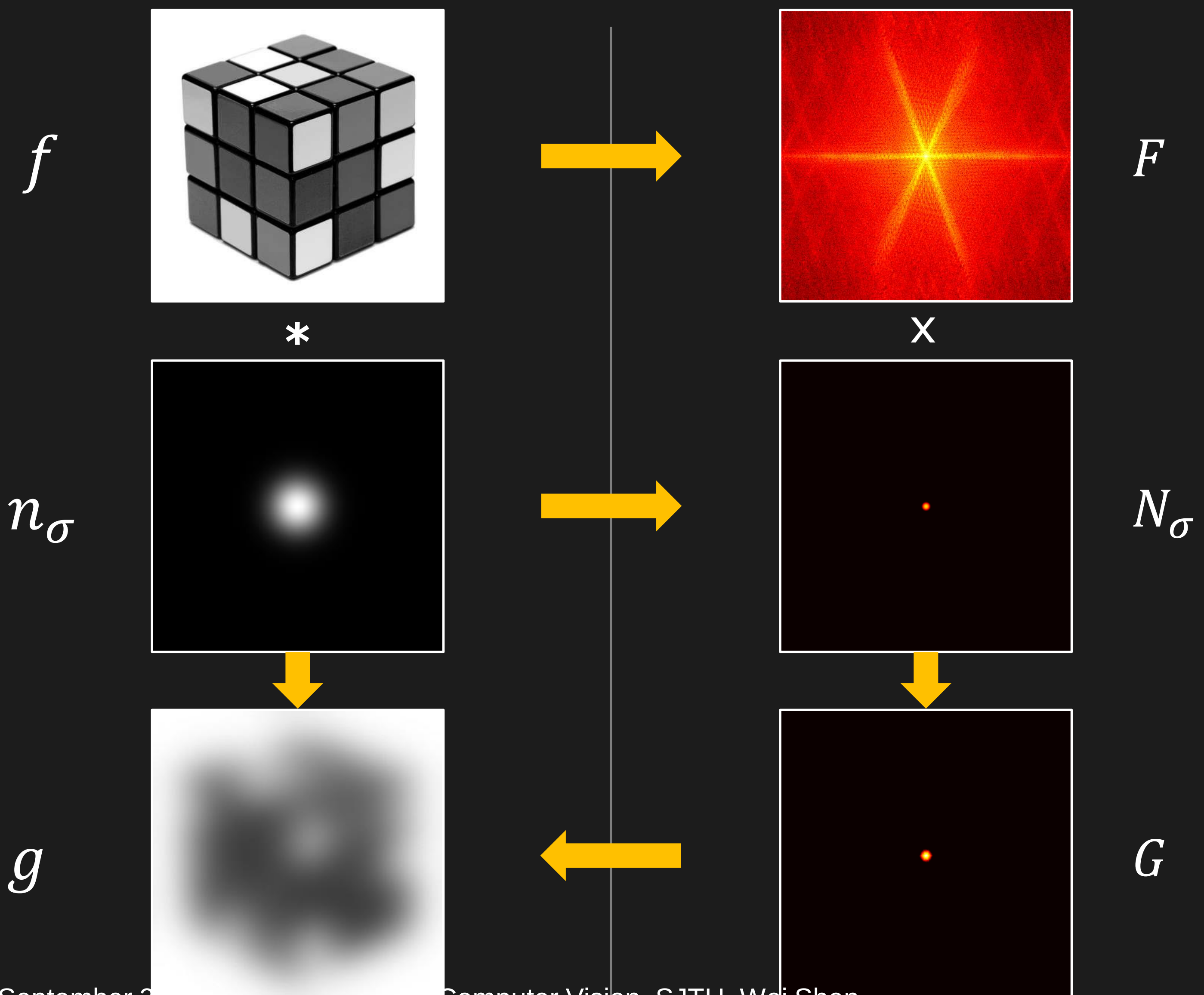
Gaussian Smoothing



Gaussian Smoothing

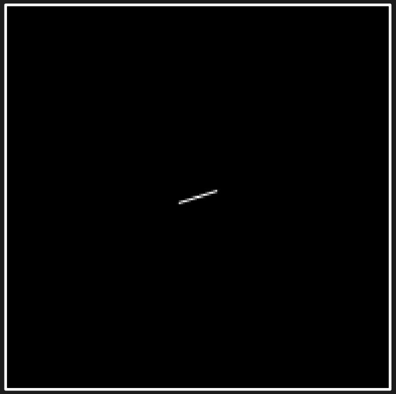


Gaussian Smoothing



Motion Blur

*



=

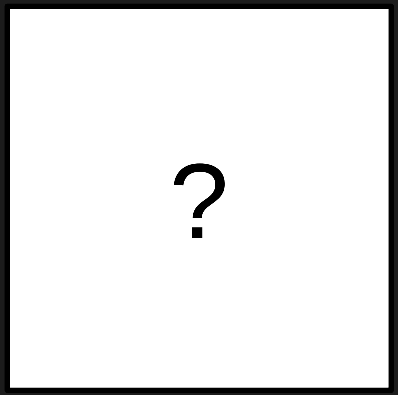
Scene $f(x, y)$

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

Image $g(x, y)$

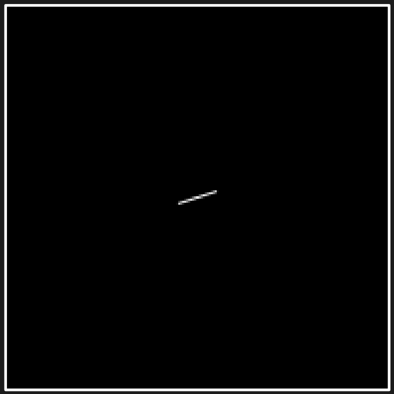
$$\boxed{f(x, y) * h(x, y) = g(x, y)}$$

Motion Blur



Scene $f(x,y)$

$*$



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

$=$

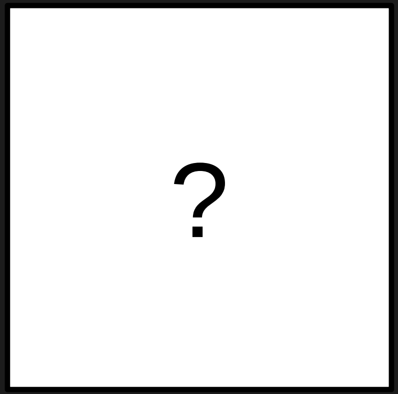
Image $g(x,y)$

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

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

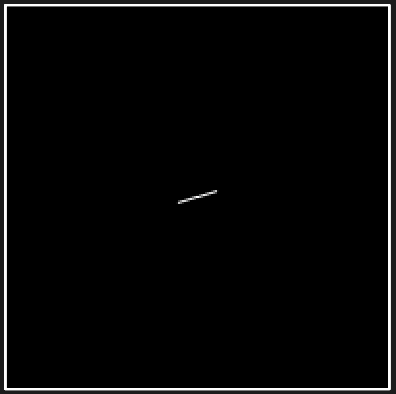
Fourier Transform To the Rescue!

Motion Deblur: Deconvolution



Scene $f(x,y)$

$*$



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

$=$

Image $g(x,y)$

Let f' be the recovered scene.

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

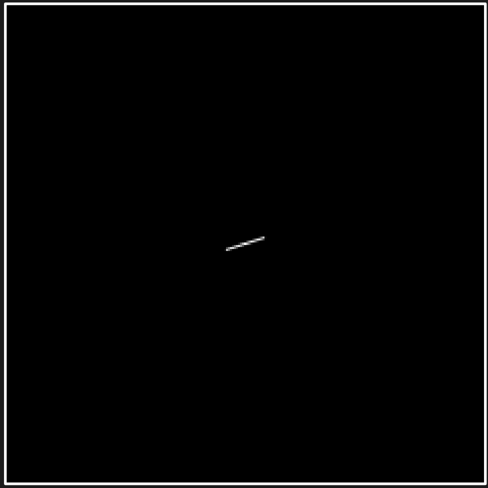
$$F'(u,v)H(u,v) = G(u,v)$$

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

Motion Deblur: Deconvolution

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

/



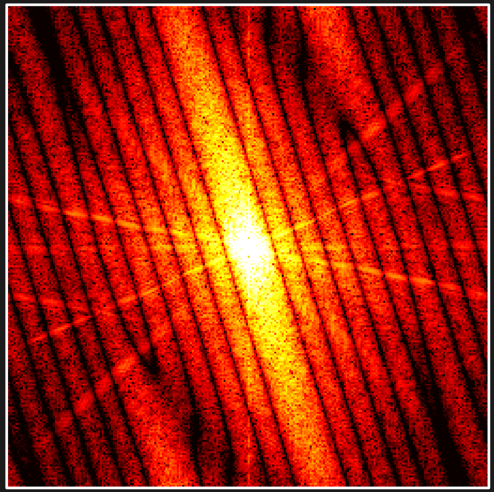
=

?

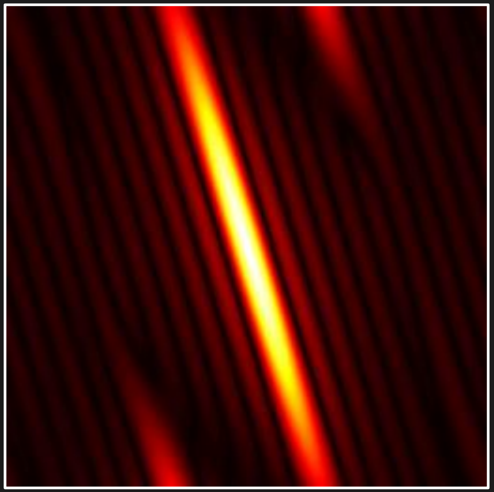
Image $g(x,y)$ PSF $h(x,y)$ Recovered $f'(x,y)$

Motion Deblur: Deconvolution

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



/



=

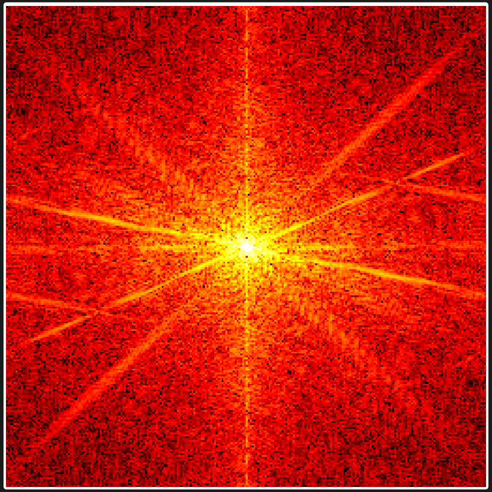


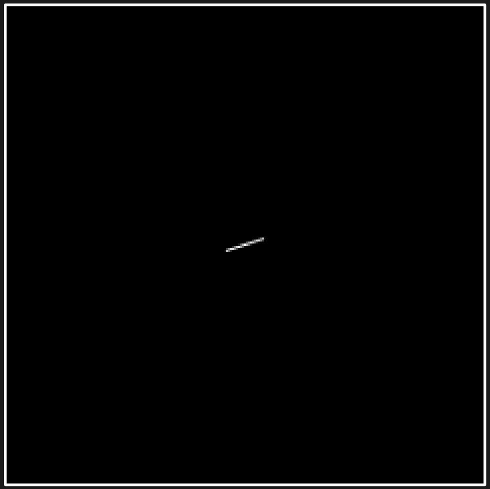
Image $G(u,v)$ PSF $H(u,v)$ Recovered $F'(u,v)$

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

Motion Deblur: Deconvolution

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

/



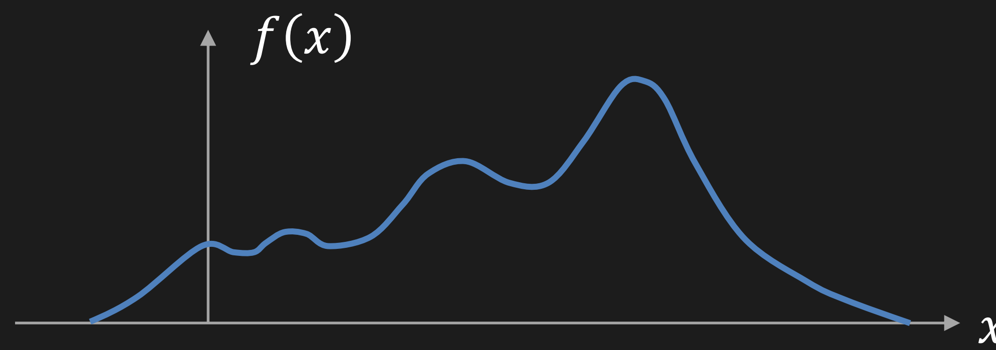
=

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

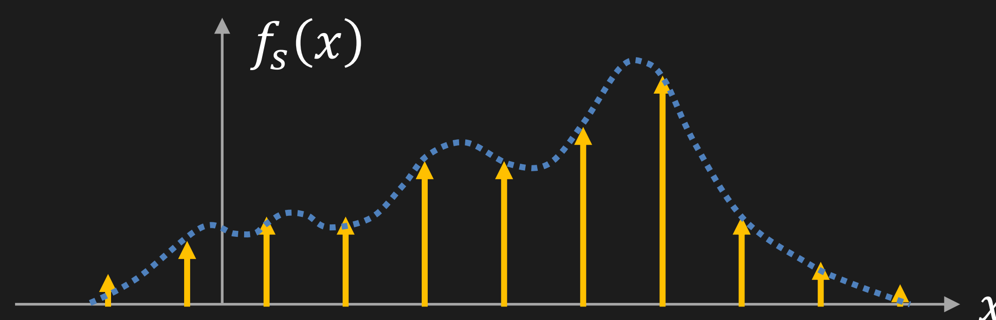
Step 2: Compute IFT of $F'(u,v)$ to recover scene

From Continuous to Digital Image

Continuous Signal:

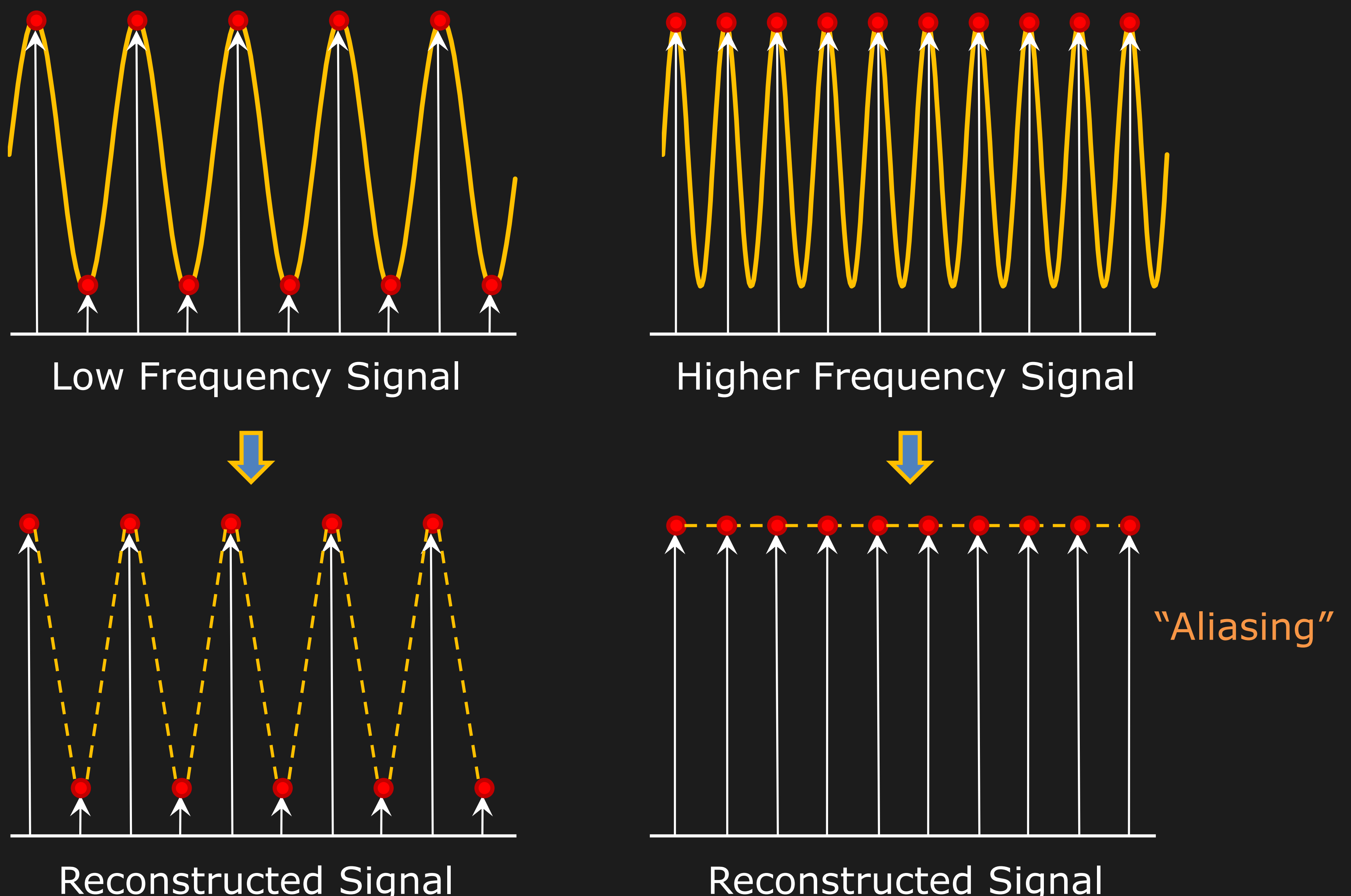


Digital Signal:



How “dense” should the samples be?

Sampling Problem



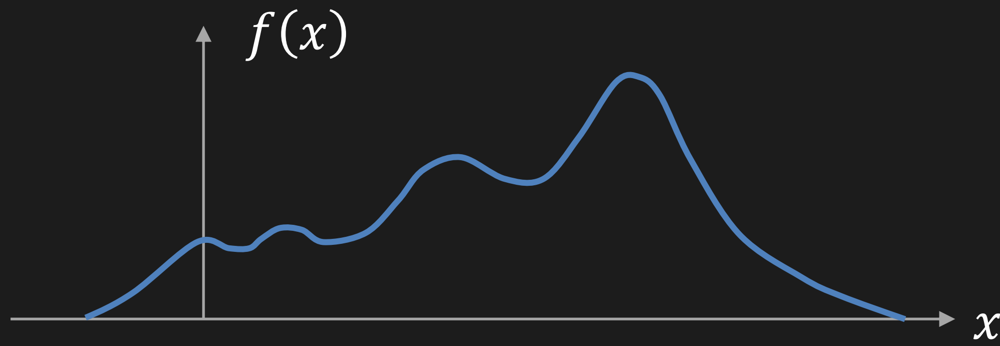
Sampling Problem

"Well sampled" image

"Under sampled" image
(visible aliasing artifacts)

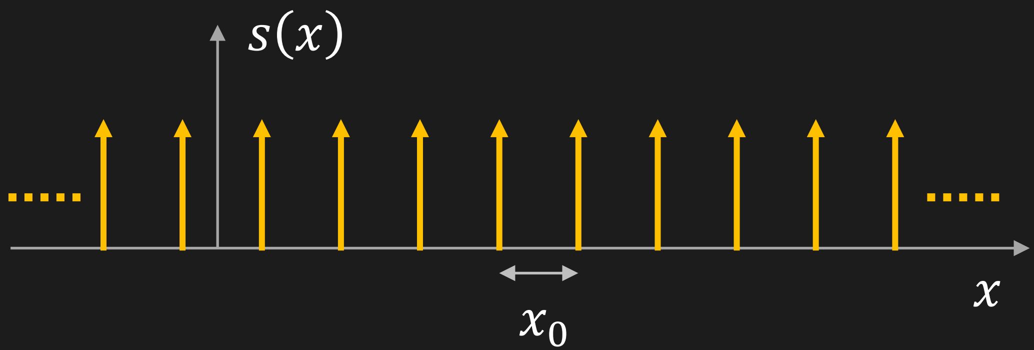
Sampling Theory

Continuous Signal:



Shah Function (Impulse Train):

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

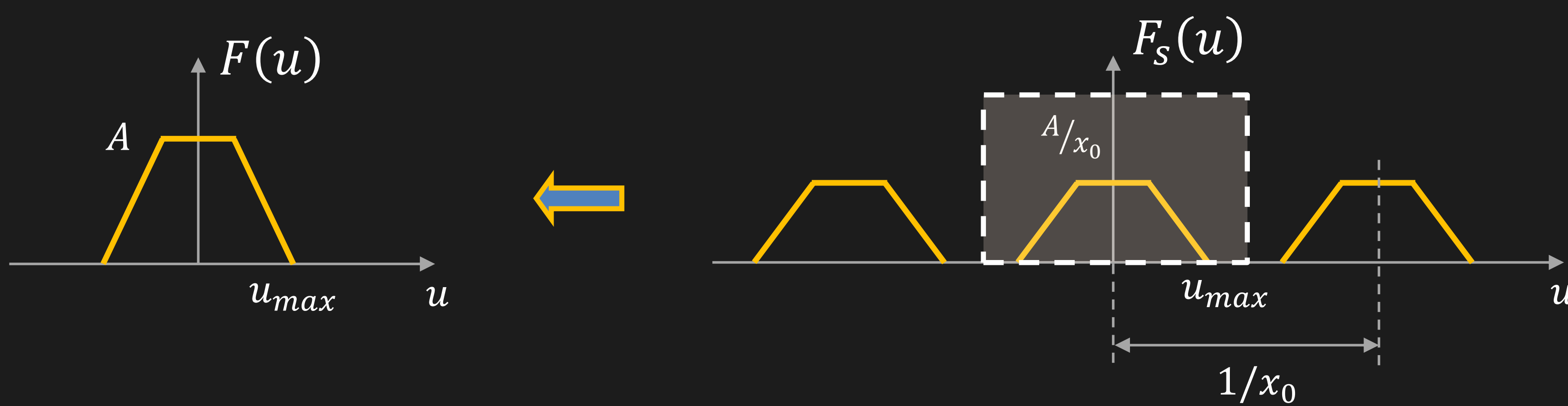


Sampled Function: $$\boxed{f_s(x) = f(x)s(x)}$$

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)



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

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

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

Aliasing in Digital Image Sensors

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

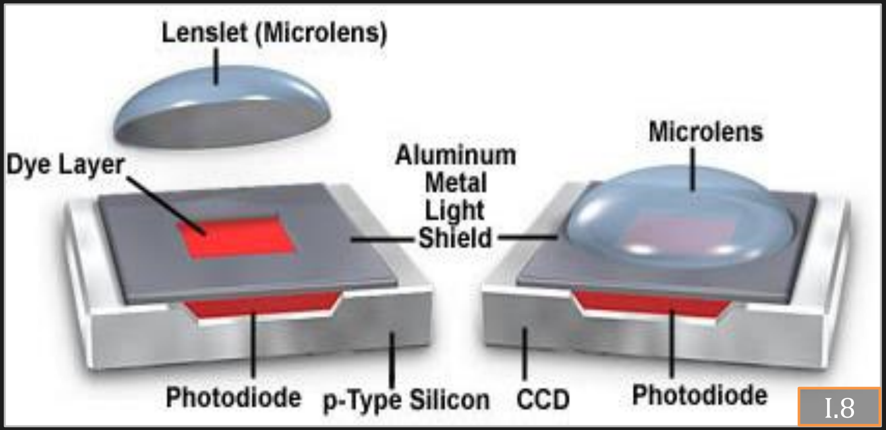
Aliasing artifacts usually occur in the form of Moiré patterns

Aliasing is unavoidable.
But its effects can be minimized.

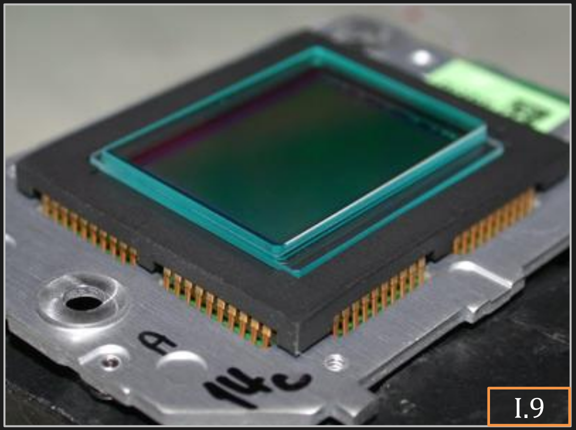
Minimizing the Effects of Aliasing

Band Limit: Clip the signal above the Nyquist frequency.

Effectively, “blur” the scene before sampling.



Pixels are area-samplers
(box-averaging filter)



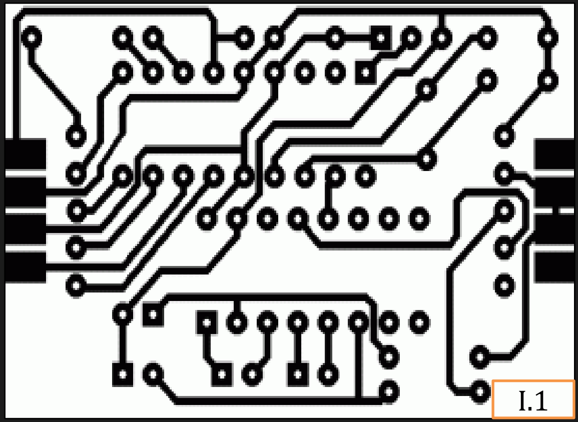
Use optical low-pass filter
(anti-aliasing filter)

BINARY IMAGE PROCESSING

What are Binary Images?

Binary Image: Can have only two values (0 or 1).

Simple to process and analyze.



I.1

I.2

I.3

Binary Images: Properties and Methods

Binary Image: Can have only two values (0 or 1). Simple to process and analyze.

Topics:

(1) Geometric Properties

(2) Discrete Binary Images

(3) Multiple Objects (Connectivity)

(4) Sequential and Iterative Processing

Representation

- A (grey) image I is a function $I : \begin{cases} \Omega \subset \mathbb{R}^2 & \longrightarrow & \mathbb{R} \\ p = (x, y) & \longmapsto & I(x, y) \end{cases}$

- Represented, after sampling and quantization, by a matrix

Representation

256x256 128x128 64x64 32x32 16x16

Number of pixels

32 16 8 4 2

Number of gray levels

Making Binary Images

Binary Image $b(x, y)$: Usually obtained from Gray-level (or other) image $g(x, y)$ by Thresholding.

Characteristic Function:

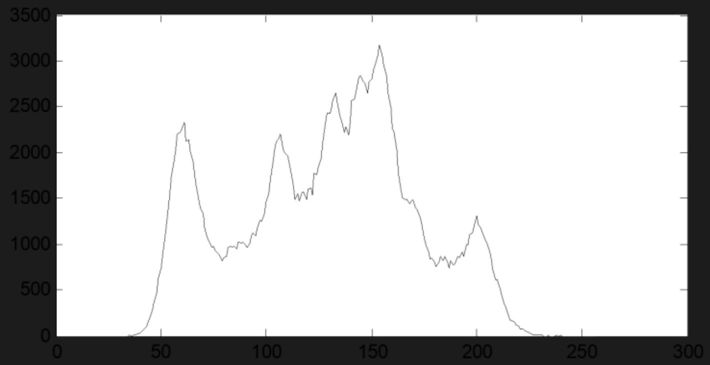
$$b(x, y) = \begin{cases} 0, & g(x, y) < T \\ 1, & g(x, y) \geq T \end{cases}$$

Histograms

I

512x512

H



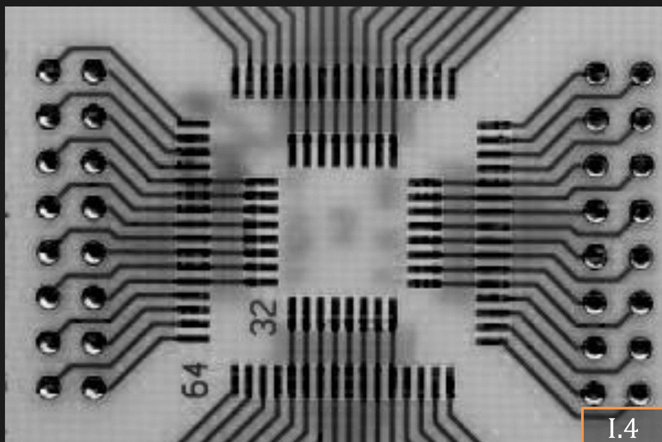
256 gray levels

- H(x) is the number of pixels in image I with grey value x
- Probability of observing grey value x ?

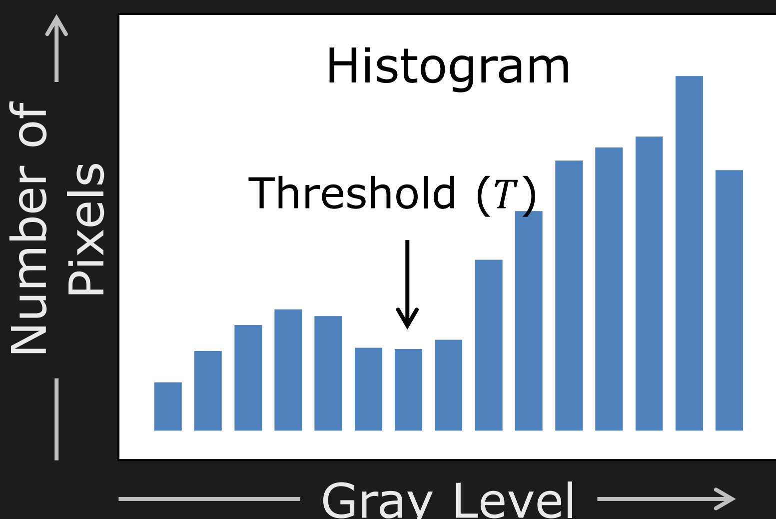
$$p(x) = \frac{H(x)}{s_x \times s_y}$$

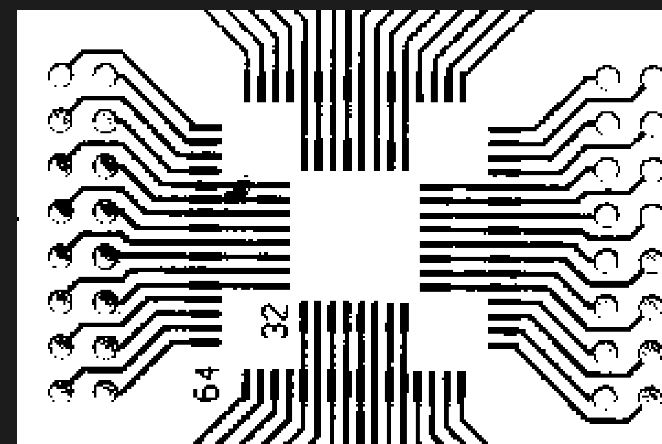
- Invariant to pixel permutations

Selecting a Threshold (T)



Gray Image $g(x, y)$



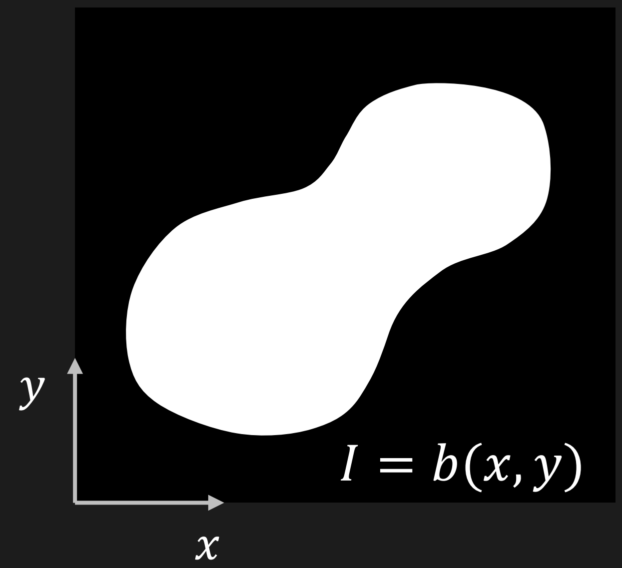


Binary Image $b(x, y)$

Geometric Properties of Binary Images

Assume:

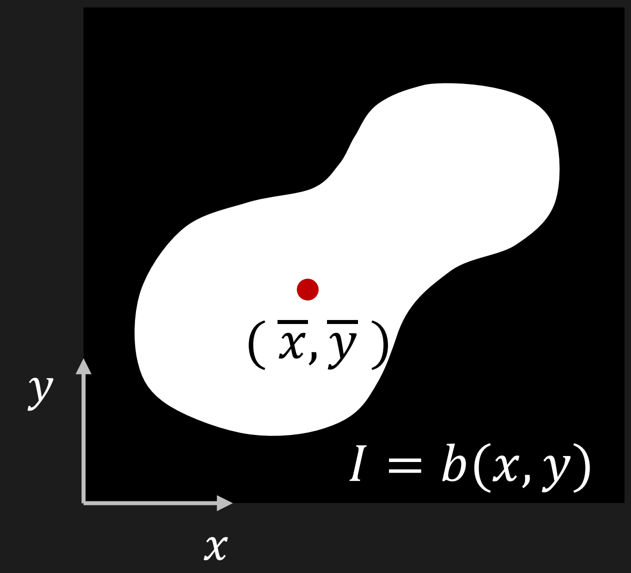
- $b(x, y)$ is continuous
- Only one object



Area and Position

Area: (Zeroth Moment)

$$A = \iint_I b(x, y)\, dx\, dy$$

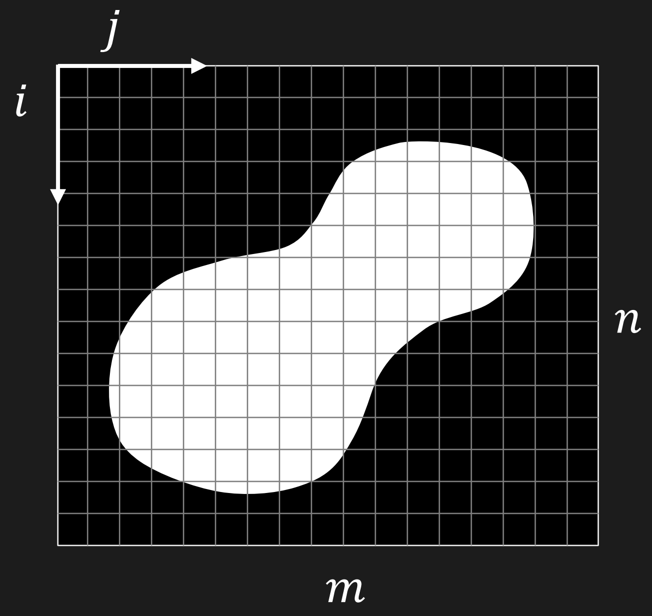


Position: Center of Area (First Moment)

$$\bar{x} = \frac{1}{A} \iint_I x\, b(x, y)\, dx\, dy \quad , \quad \bar{y} = \frac{1}{A} \iint_I y\, b(x, y)\, dx\, dy$$

Discrete Binary Images

b_{ij}: Value at cell (pixel) in row i and column j.
Assume pixel area = 1.



Area:

$$A = \sum_{i=1}^{n} \sum_{j=1}^{m} b_{ij}$$

Position: Center of Area (First Moment)

$$\bar{x} = \frac{1}{A} \sum_{i=1}^{n} \sum_{j=1}^{m} j b_{ij} \qquad \bar{y} = \frac{1}{A} \sum_{i=1}^{n} \sum_{j=1}^{m} i b_{ij}$$

Discrete Binary Images

Second Moments:

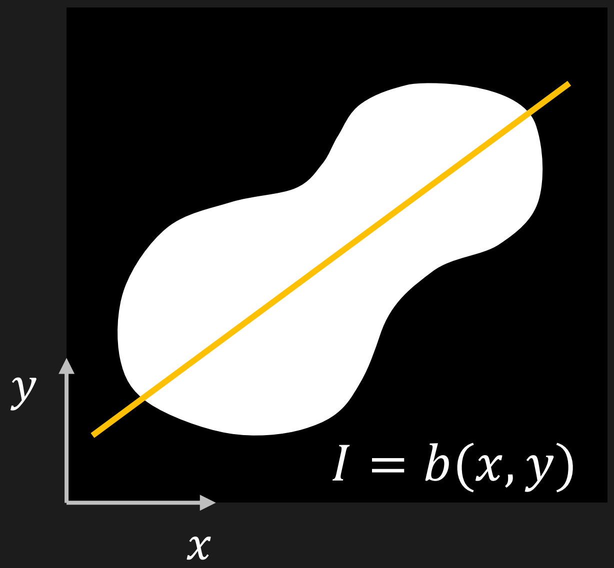
$$a' = \sum_{i=1}^{n}\sum_{j=1}^{m} i^2 b_{ij} \qquad b' = 2\sum_{i=1}^{n}\sum_{j=1}^{m} ij b_{ij} \qquad c' = \sum_{i=1}^{n}\sum_{j=1}^{m} j^2 b_{ij}$$

Note: a', b', c' are second moments w.r.t origin.
a, b, c (w.r.t. center) can be found from a', b', c', $\bar{x}$, $\bar{y}$, A

Hint: Expand $a = \sum_{i=1}^{n}\sum_{j=1}^{m}(i-\bar{y})^2 b_{ij}$ and represent in terms of a', $\bar{y}$, A.

Orientation

Difficult to define!

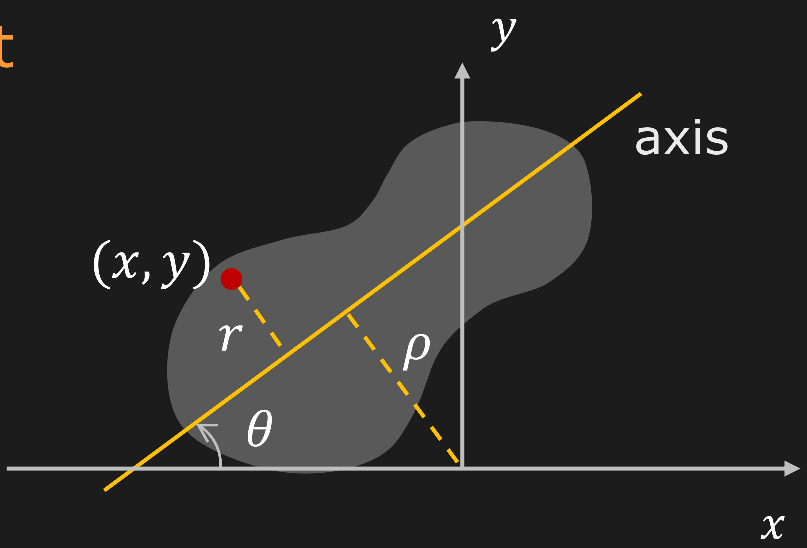


Use: Axis of Least Second Moment

Orientation

Axis of Least Second Moment minimizes:

$$E = \iint_I r^2 \, b(x,y) \, dx \, dy$$



Which equation to use for axis?

$$y = mx + b \ ? \qquad -\infty \leq m \leq \infty$$

Use: $\boxed{x \sin\theta - y \cos\theta + \rho = 0}$ $\qquad \rho,\ \theta$ are finite

Find ρ and θ that minimize E for given $b(x,y)$

Recall Polar Coordinates

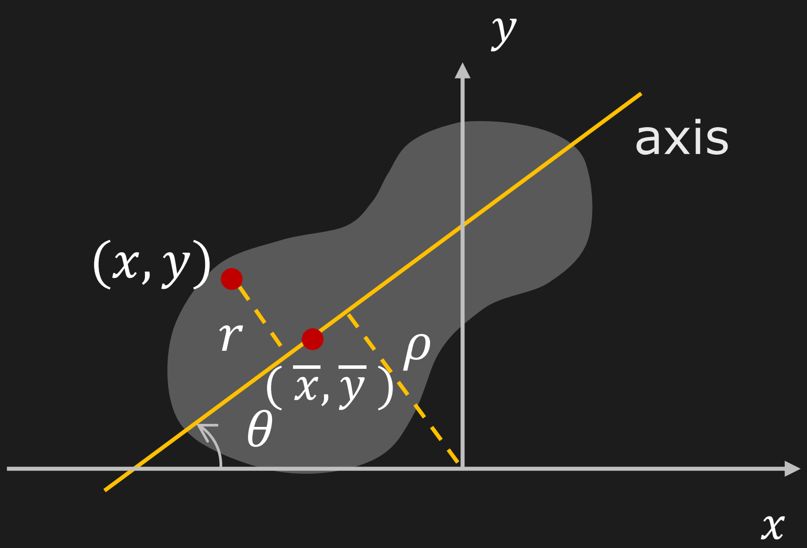
$$x = r * \cos t$$
$$y = r * \sin t$$

$$r = \mathrm{sqrt}(x*x + y*y)$$
$$t = \mathrm{atan2}(y,x)$$

Minimizing Second Moment

We can show that for any point (x, y):

$$r = x \sin\theta - y \cos\theta + \rho$$



So, minimize:

$$E = \iint_I (x \sin\theta - y \cos\theta + \rho)^2 b(x, y)\, dx\, dy$$

Using $\dfrac{\partial E}{\partial \rho} = 0$ we get: $A(\underline{\bar{x} \sin\theta - \bar{y} \cos\theta + \rho}) = 0$

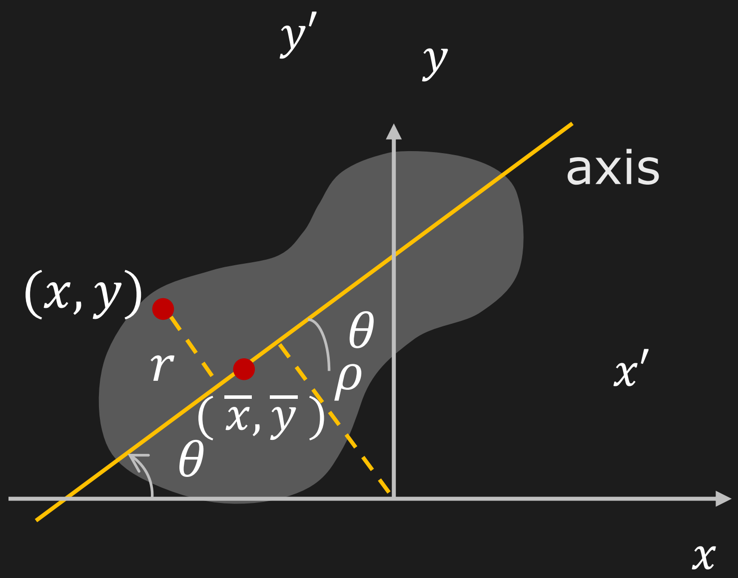
Axis passes through center $(\bar{x}, \bar{y})$!

Shift the Coordinate System

Change coordinates:

$$x' = x - \bar{x},\ y' = y - \bar{y}$$

$$x \sin\theta - y\cos\theta + \rho$$
$$= x' \sin\theta - y' \cos\theta$$



Therefore, we can rewrite E as:

$$\boxed{E = a\sin^2\theta - b\sin\theta\cos\theta + c\cos^2\theta}$$

where: $\begin{cases} a = \iint_{I'} (x')^2\, b(x,y)\, dx'\, dy' \\ c = \iint_{I'} (y')^2\, b(x,y)\, dx'\, dy' \end{cases}$

$$b = 2\iint_{I'} (x'y')\, b(x,y)\, dx'\, dy'$$

(a, b, c are easy to compute)

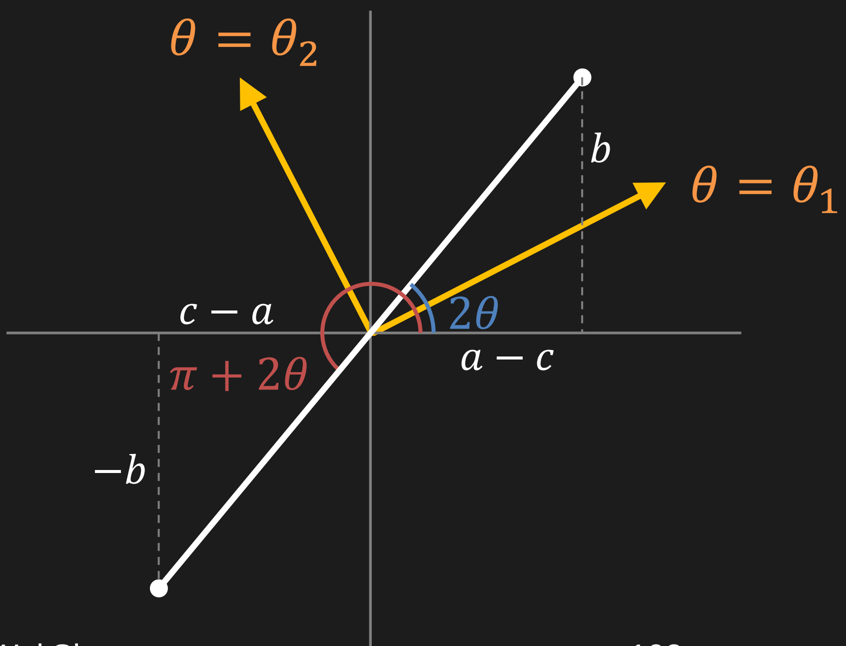
Finally, Minimize E

Using $\frac{dE}{d\theta} = (a - c)\sin 2\theta - b\cos 2\theta = 0$ we get: $\boxed{\tan 2\theta = \frac{b}{a - c}}$

We know that: $\tan 2\theta = \tan(2\theta + \pi) = \frac{-b}{c - a}$

θ has two solutions.

1. $\theta = \theta_1$
2. $\theta = \theta_2 = \theta_1 + \frac{\pi}{2}$



One gives Minimum of E
and the other Maximum of E

Which One To Use?

Using second derivative test:

If $\frac{d^2E}{d\theta^2} = (a-c)\cos 2\theta + b\sin 2\theta \begin{cases} > 0 \text{ then Minimum} \\ < 0 \text{ then Maximum} \end{cases}$

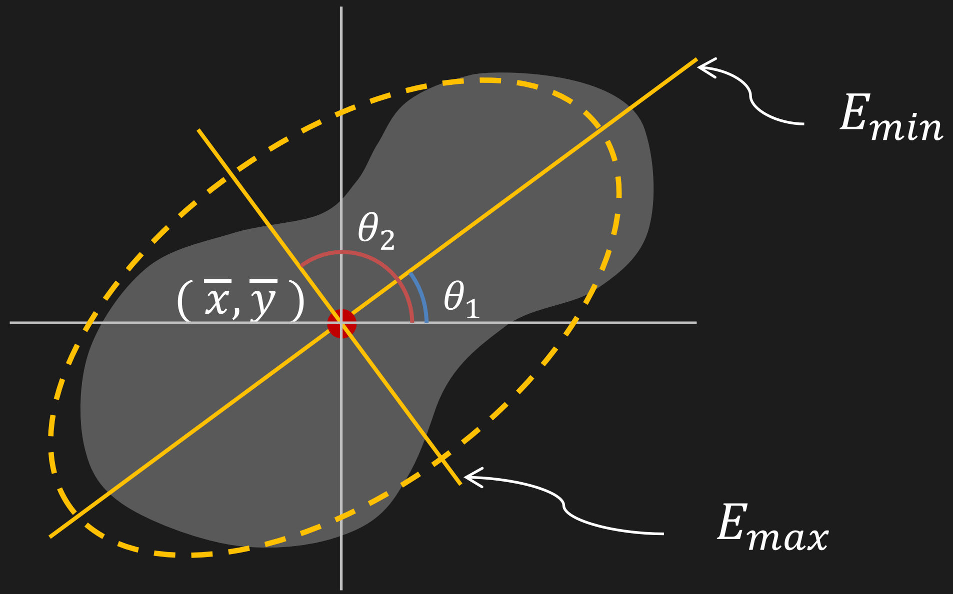
Substituting $\cos 2\theta_1$, $\sin 2\theta_1$, $\cos 2\theta_2$ and $\sin 2\theta_2$:

$$\frac{d^2E}{d\theta^2}(\theta_1) > 0 \quad \text{and} \quad \frac{d^2E}{d\theta^2}(\theta_2) < 0$$

Therefore,

Orientation: $\boxed{\theta = \theta_1 = \frac{atan2\left(\frac{b}{a-c}\right)}{2}}$

Roundedness

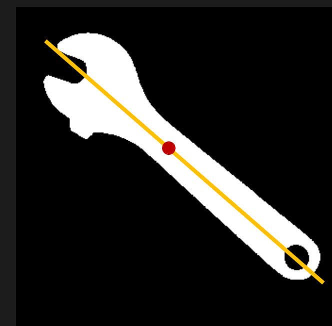
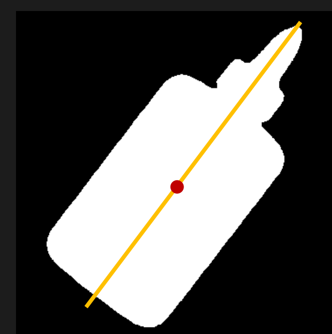
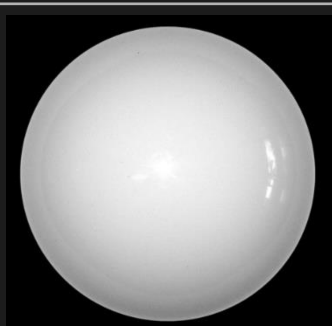
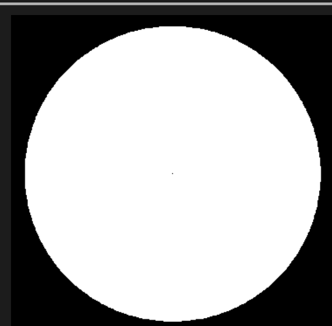
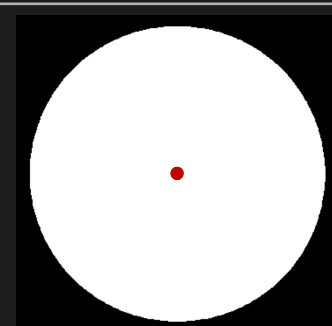


$$E = a\sin^2\theta - b\sin\theta\cos\theta + c\cos^2\theta$$

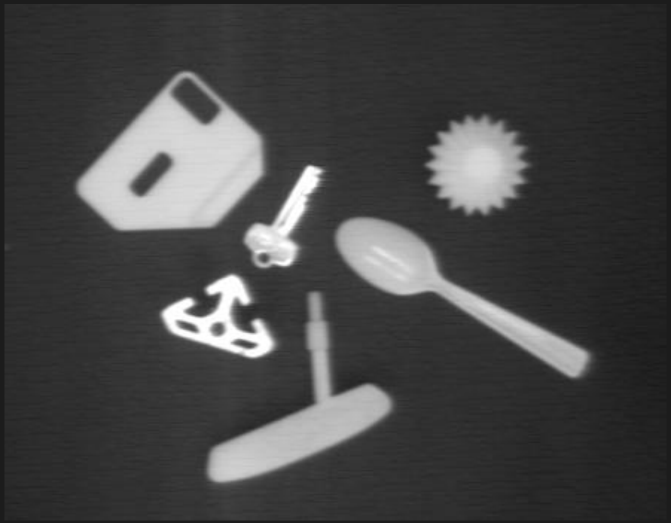
$$\boxed{\text{Roundedness} = \frac{E_{min}}{E_{max}}}$$

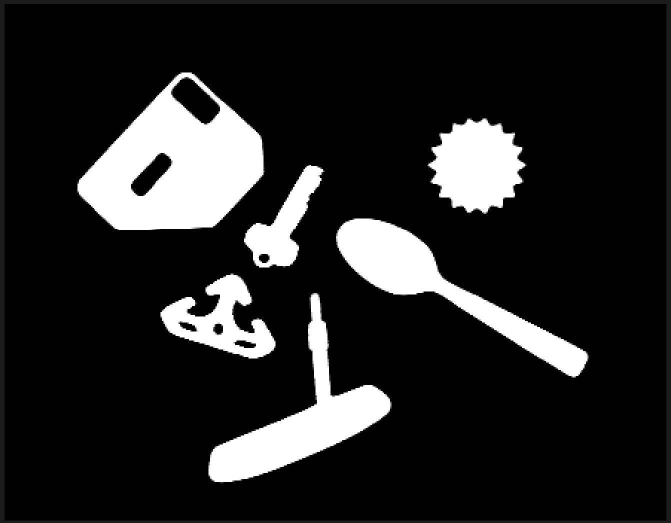
where: $E_{min} = E(\theta_1)$ and $E_{max} = E(\theta_2)$

Examples

Gray Image	Binary Image	Orientation	Roundedness
			0.19
			0.49
			1.0

Multiple Objects



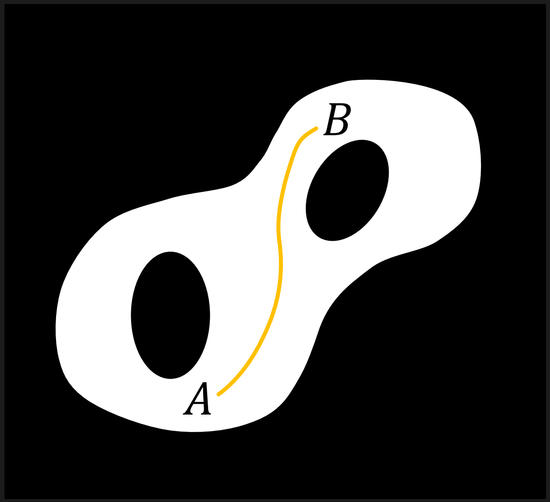


Need to Segment image into separate Components

Non-Trivial!

Connected Component

Maximal Set of Connected Points



A and B are connected if path exists between A and B along which $b(x, y)$ is constant.

Connected Component Labeling

Region Growing Algorithm

(a) Find Unlabeled "Seed" point with $b = 1$.
If not found, Terminate.

(b) Assign New Label to seed point

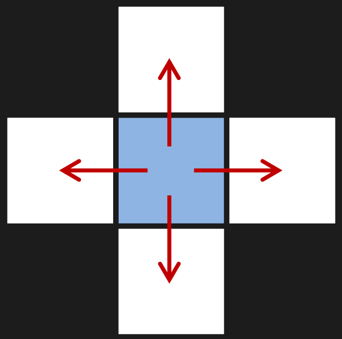
(c) Assign Same Label to its Neighbors with $b = 1$

(d) Assign Same Label to Neighbors of Neighbors with $b = 1$. Repeat until no more Unlabeled Neighbors with b=1.

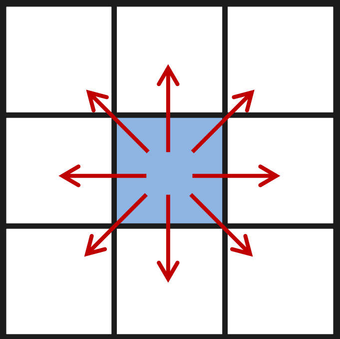
(e) Go to (a)

What do we mean by Neighbors?

Connectedness



4-Connectedness
4-C



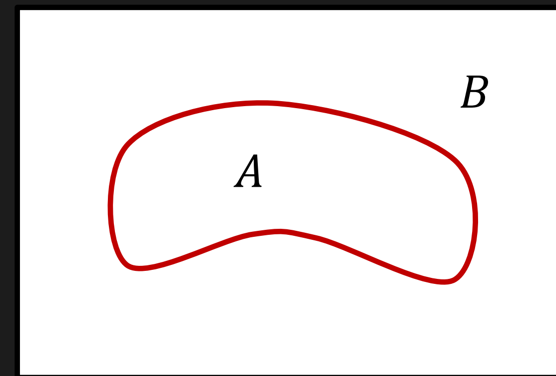
8-Connectedness
8-C

Neither is Perfect!

Connectedness

Jordan's Curve Theorem

Closed curve
→ 2 Connected Regions



Consider

0	1	0
1	0	1
0	1	0

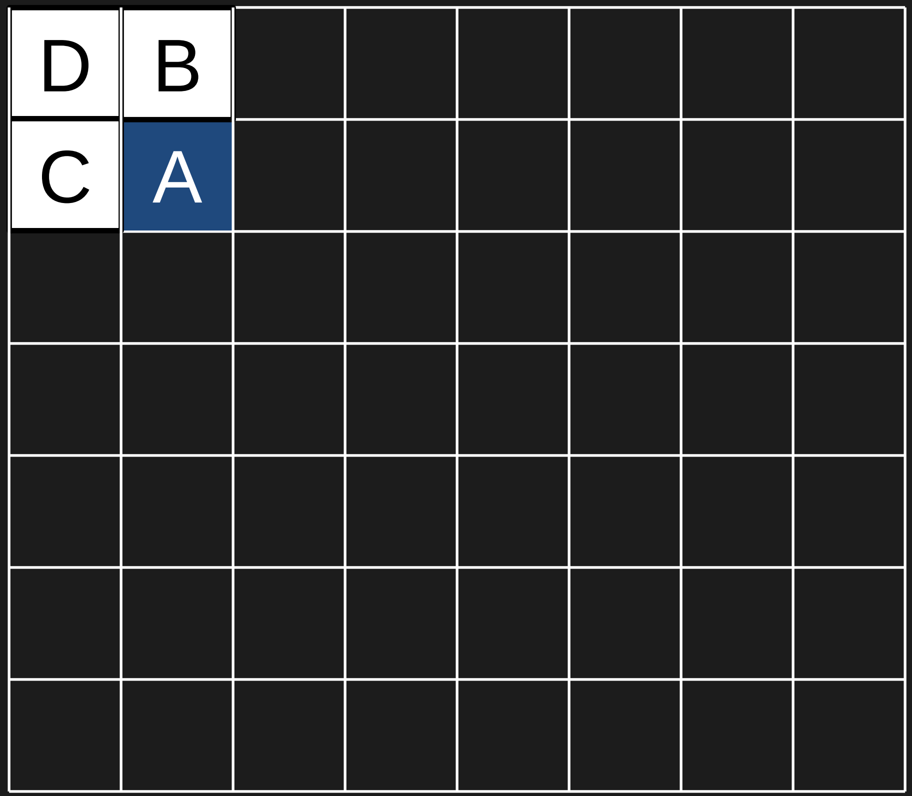
B1	O1	B1
O4	B2	O2
B1	O3	B1

4-C
Hole without a closed loop!

B	O	B
O	B	O
B	O	B

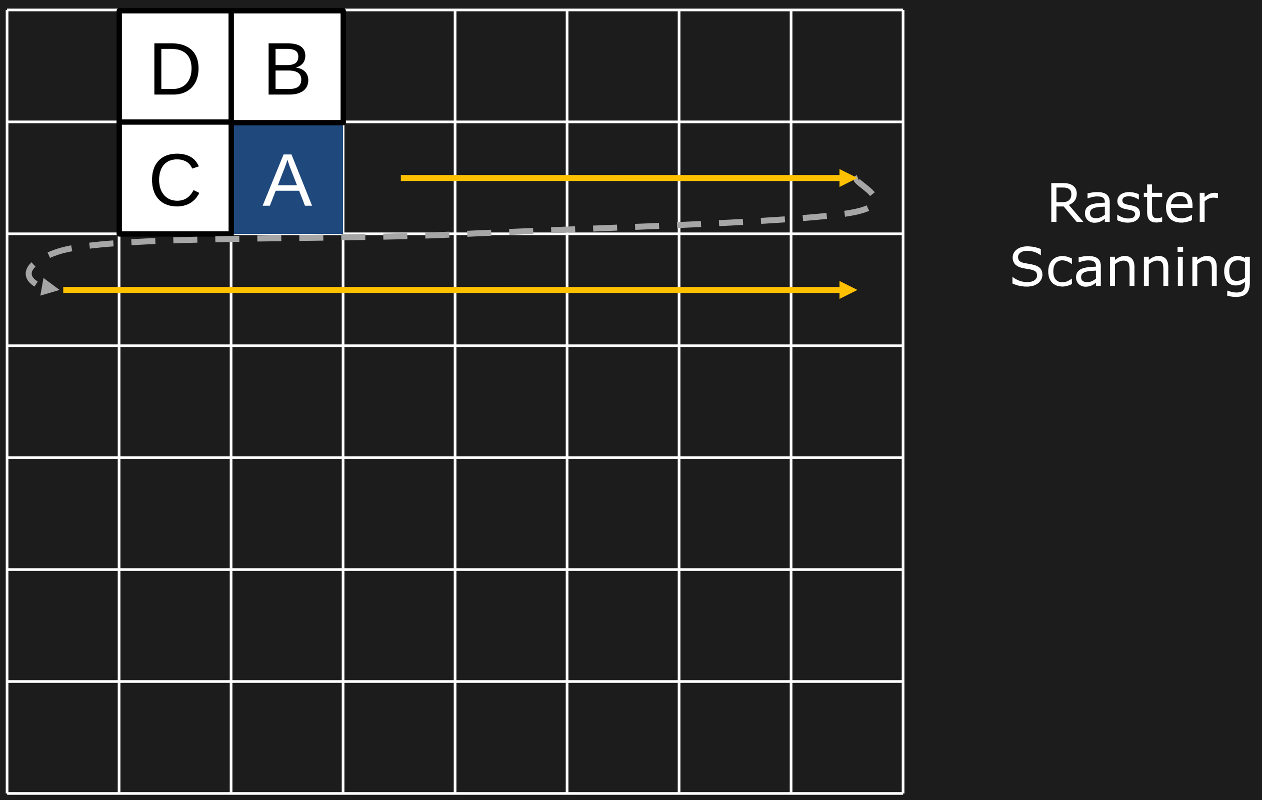
8-C
Connected backgrounds with a closed loop!

Sequential Labeling Algorithm



We want to label A.
B, C, D are already labeled.

Sequential Labeling Algorithm



We want to label A.
B, C, D are already labeled.

Sequential Labeling Algorithm

X	X
X	0

→ label(A) = "background"

0	0
0	1

→ label(A) = new label

D	X
X	1

→ label(A) = label(D)

0	0
C	1

→ label(A) = label(C)

0	B
0	1

→ label(A) = label(B)

0	B
C	1

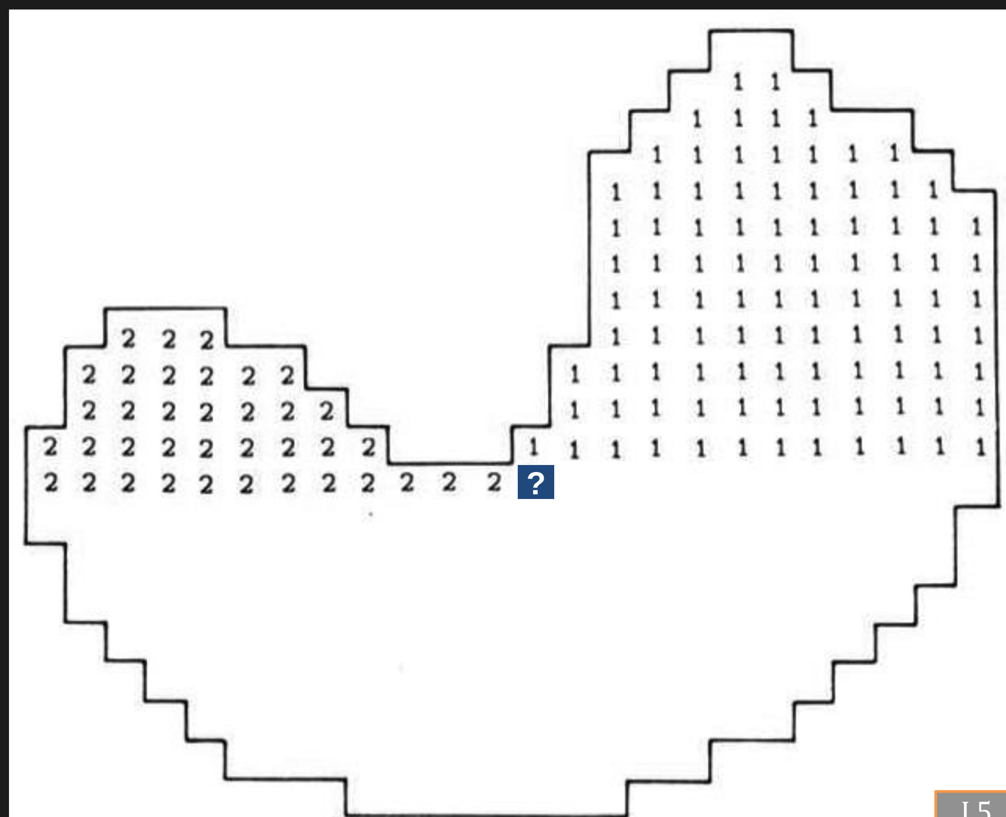
→ If
    label(B) = label(C)
then,
    label(A) = label(B)

X: Value does not matter (Can be 0 or 1)

Sequential Labeling Algorithm

0	B
C	1

→ What if label(B) not equal to label(C)?



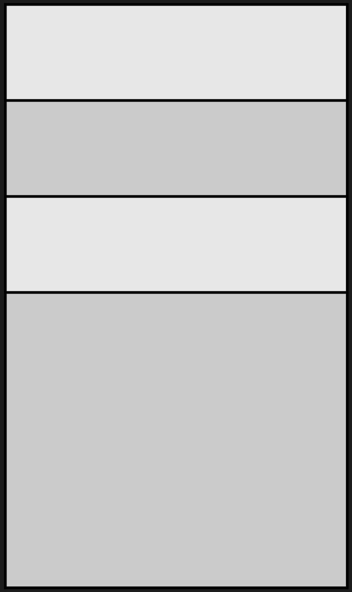
Sequential Labeling Algorithm

0	B
C	1

→ What if label(B) not equal to label(C)?

Solution: Create Equivalence Table

- Note down that label(B) ≡ label(C)
- Assign label(A) = label(B)

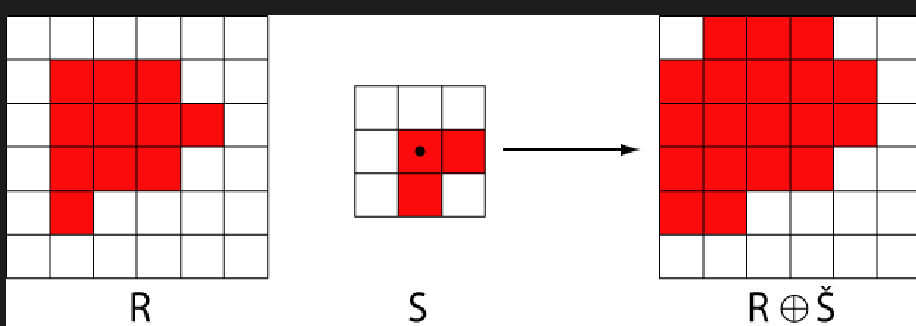


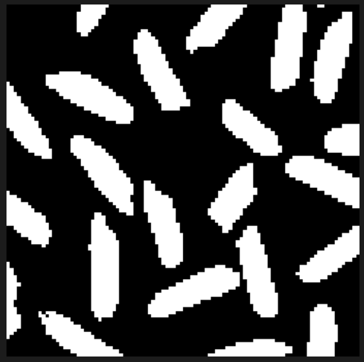
Resolve Equivalence in Second Pass

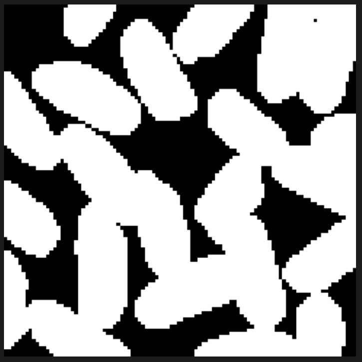
Morphological operators

Binary dilation

- Defined by a Morphological structuring element ***S*** (a binary template)
- Images are represented by the sets $(\subset Z^2)$ containing the positions of their non-zero elements
- Binary dilatation $D(R,S) = R \oplus S = \{u - v | u \in R, v \in S\}$
- (Intuitively: set of all possible positions of the center of ***S*** such that the two patterns overlap by at least one element)




Original binary image

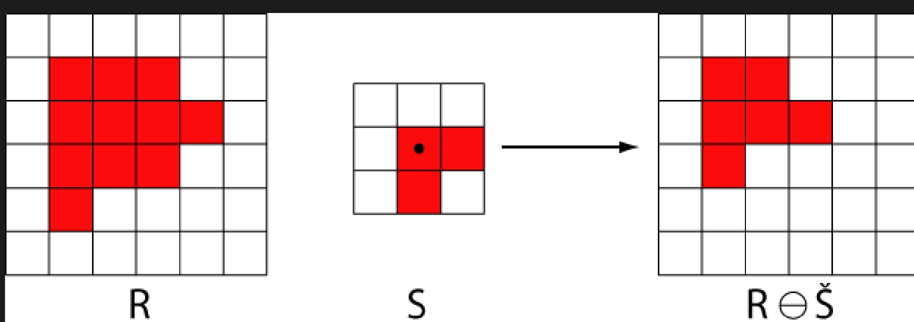

Dilated image

$S = \{(0,0),(1,0),(0,1)\}$

$R = \{(1,1),(1,2),(1,3),(2,1),(2,2),(2,3),(2,4),(3,1),(3,2),(3,3),(4,1)\}$

Binary erosion

- Defined by a Morphological structuring element ***S***
- Binary erosion $E(R,S) = R \ominus S = \{u \mid \forall v \in S, u+v \in R\}$
- (Intuitively: all positions of the center of S such that pattern ***S*** is contained in pattern R)




Original binary image


Eroded image

$S = \{(0,0),(1,0),(0,1)\}$

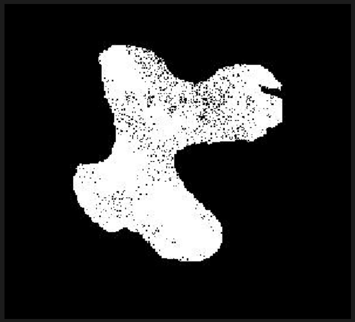
$R = \{(1,1),(1,2),(1,3),(2,1),(2,2),(2,3),(2,4),(3,1),(3,2),(3,3),(4,1)\}$

Binary Closing

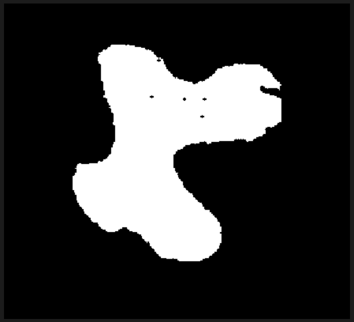
- Defined by a Morphological structuring element S
- Binary closing $C(R,S) = E(D(R,S),S)$
  - Properties:

    Fill the **holes smaller** than the structuring elements

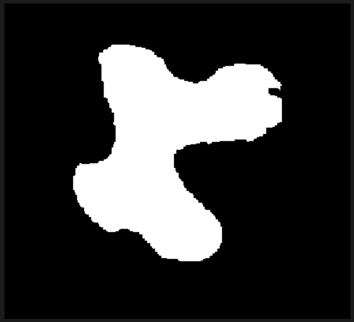
    Smooth the contours by filling the cavities



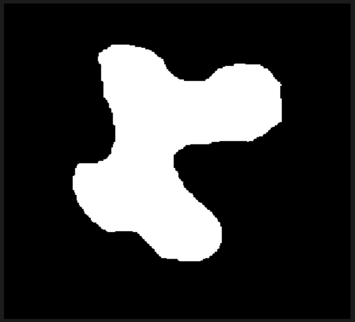
Original binary image



Radius of the structuring element R = 1



R = 3



R = 10

Binary opening

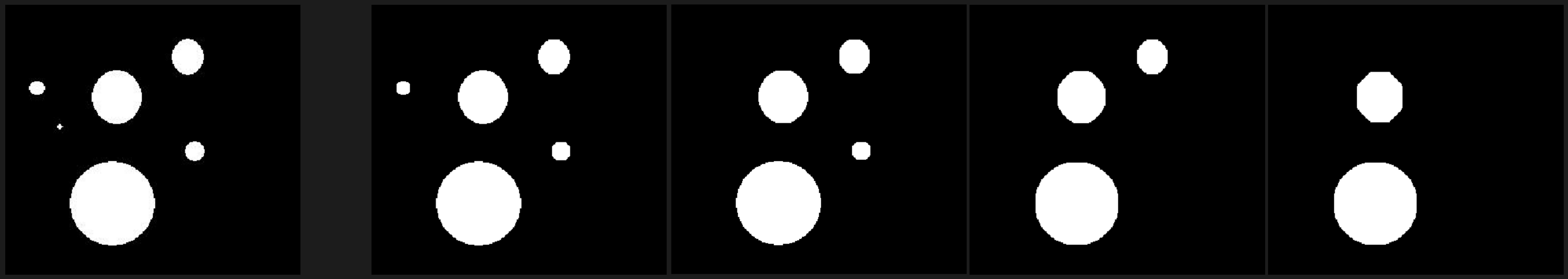
- Defined by a Morphological structuring element ***B***
- Binary opening $O(R,S) = D(E(R,S),S)$
  - Properties:

    Suppress the **structures smaller** than the structuring elements

    Delete the link between weak connected components

    Smooth the contours by deleting the outgrowths

    Applications: **Granulometry**



Original binary image

R = 5

R = 7

R = 10

R = 20

Examples

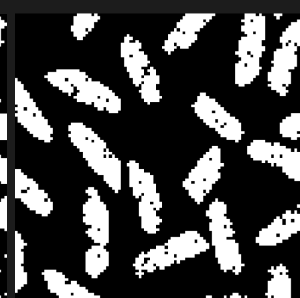
- Removing the noise perturbation

  - Close-open operation: $O\left(C(R,S),S\right)$

  - Open-close operation: $C\left(O(R,S),S\right)$


Original binary image


Close


Open


Close-open


Open-Close

- Note: morphological operations can be generalized to grey value images

References: Textbooks

Computer Vision: Algorithms and Appications (Chapter 3.3-3.4)
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