CSET340

Advanced Computer Vision and Video Analytics

2nd Feb. to 06th Feb. 2026

Overall Course Coordinator-
Dr. Gaurav Kumar Dashondhi
Gaurav.dashondhi@bennett.edu.in

Note : Any query related to course then first connect with overall course coordinator.

Fourier Transform

Sampling theory and aliasing

Fourier Series

Fourier Transform

Convolution theorem

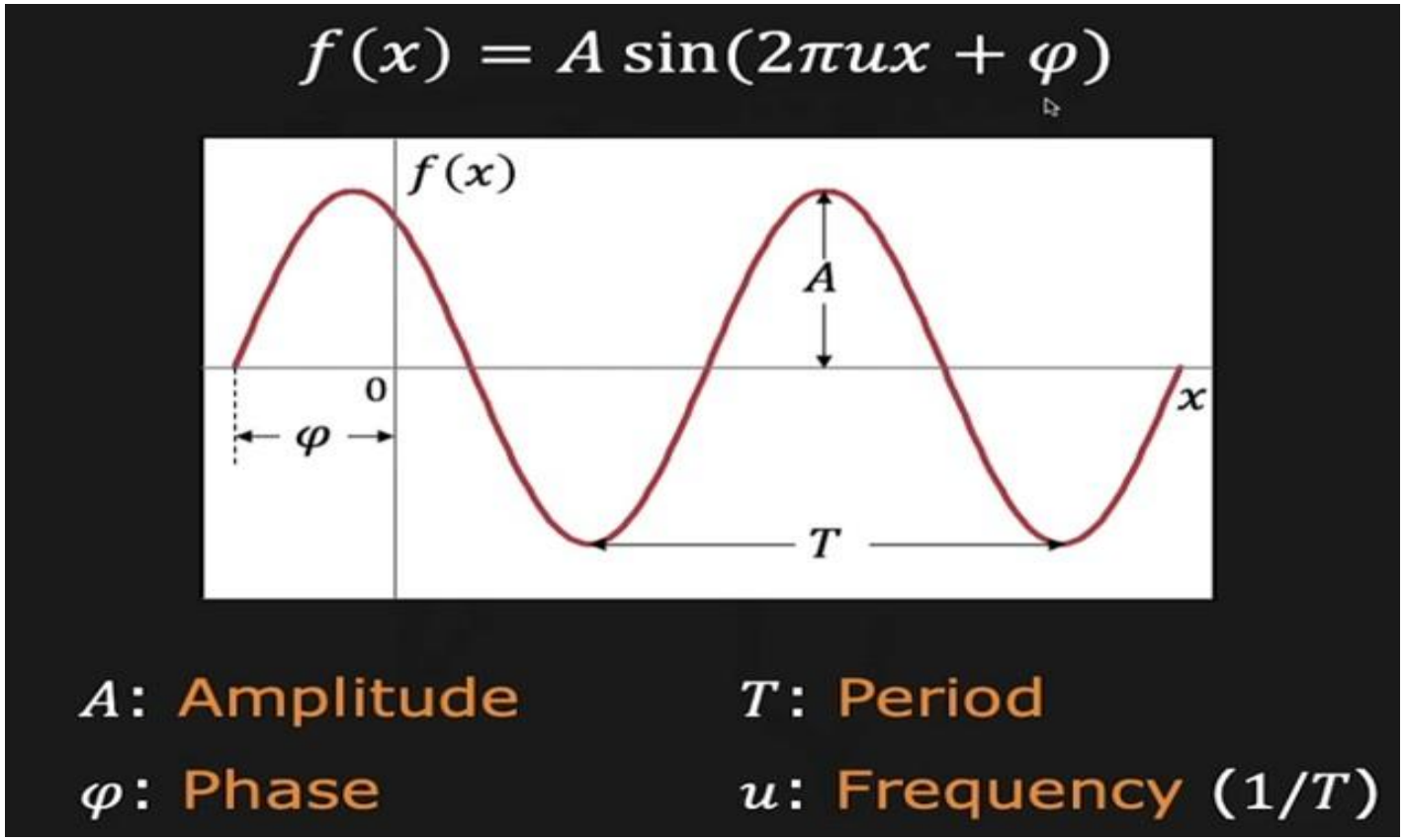
Frequency domain Filtering

Importance of the phase information

Application of FT

Deconvolution in Frequency domain

Math Primer.



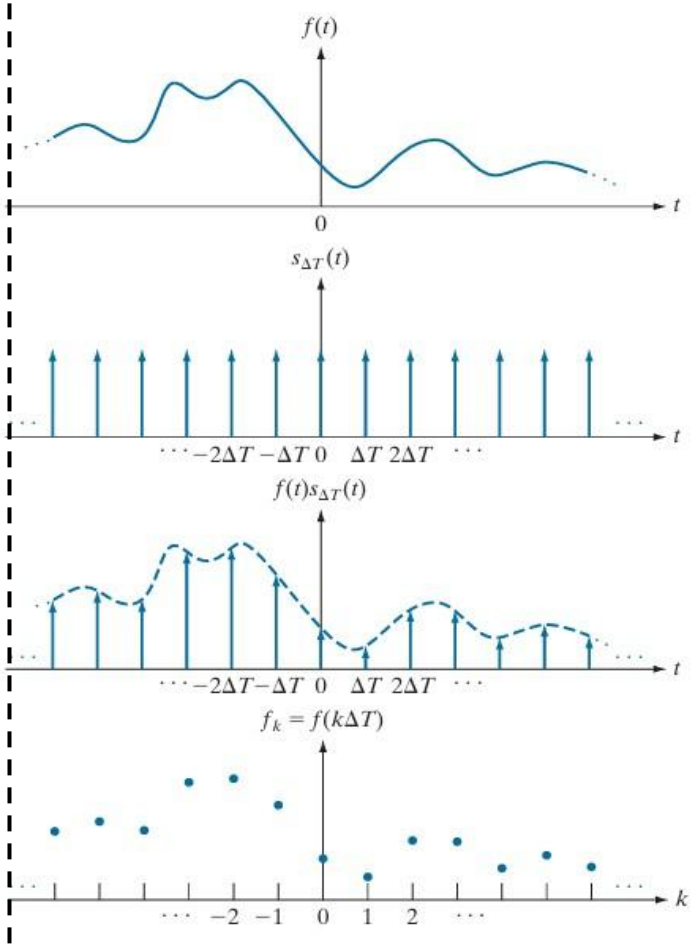
Sampling

Motivation:
As we know, lens of the camera creates the continuous optical image on the image plane.

Image sensor convert this continuous image into discrete digital image.

How densely we should sample the continuous image so that we do not lose important information in it. At the same time, sampling not introduced any artifacts or any undesirable effects in the final captured image.

Sampling: A continuous functions has to be converted into the sequence of discrete values before they processed in the computers.



(a) Continuous Function

(b) Train of Impulses used to model sampling.

(c) Sampled function formed as product of **a** and **b**.

Aliasing in 2D image

If sampling not done properly then some undesirable artifacts added in the image

Frequency Domain Fundamentals: Sampling theorem, Nyquist rate

Sampling Theorem: A continuous bandlimited signal can be recovered completely from the set of its samples if the samples are acquired at a rate exceeding at least twice the highest frequency content of the function.

Fs > 2fm (Oversampled)

Fs = 2fm (Perfect sampling or Nyquist rate)

A sampling rate exactly equal to twice the highest frequency is called the **Nyquist rate.**

Fs < 2fm (Under sampling or **aliasing** effect)

Where,

Fs = sampling frequency

fm = highest frequency component present in the signal

Nyquist Frequency = Nyquist rate / 2

Example -

$m(t) = \sin 2\Pi t + \sin 3\Pi t + \sin 4\Pi t$

$W = 2\Pi$

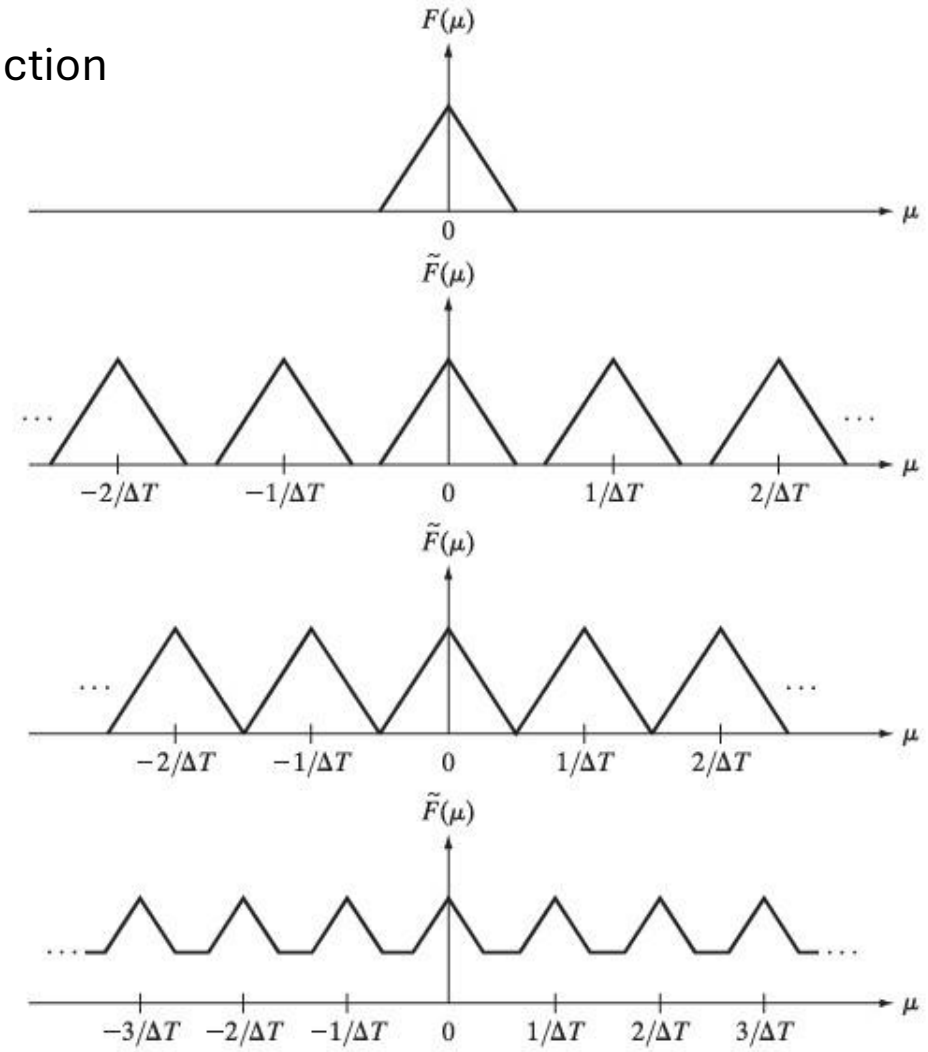
$2\Pi f = 2\Pi$

$f = 1$

Similarly f = 1.5 and 2 for other two cases respectively.

Frequency Domain Fundamentals: Sampling theorem, Nyquist rate

FT of band limited function



Oversampling

Fs > 2fm (Oversampled)

Critical sampling

Fs = 2fm (Perfect sampling or Nyquist rate)

A sampling rate exactly equal to twice the highest frequency is called the **Nyquist rate.**

Under sampling

Fs < 2fm (Under sampling or **aliasing** effect)

Where,
Fs = sampling frequency, fm = highest frequency component present in the signal

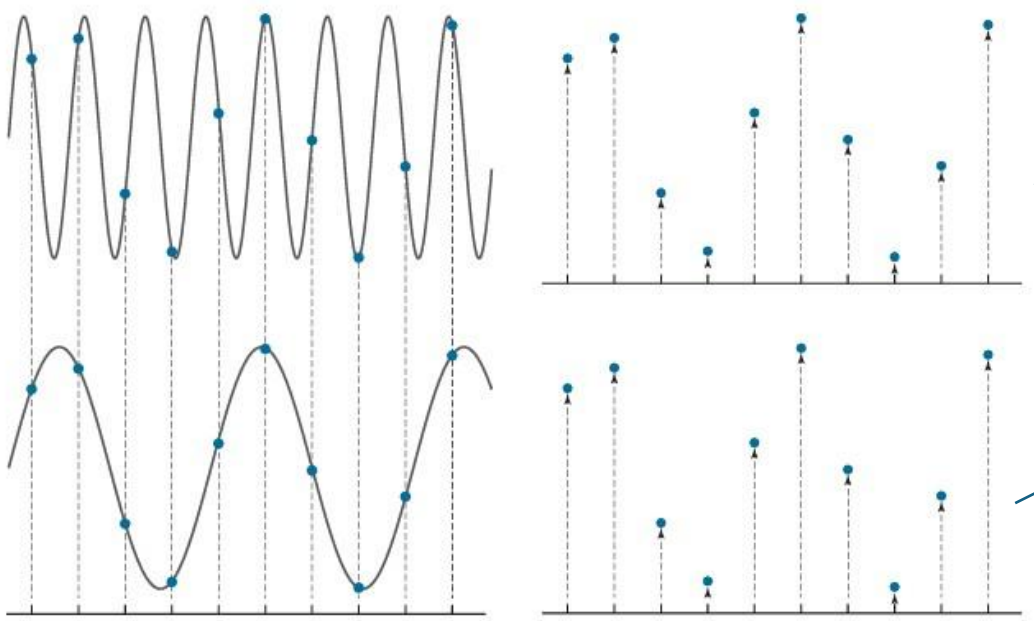
Frequency Domain Fundamentals: Aliaising

What happened if during sampling of continuous function, someone is not following the sampling theorem ?

Answer: Aliasing or false identity

Aliasing: It's a phenomena where different signals are indistinguishable from one another after sampling.

Fs < 2fm (Under sampling or **aliasing** effect)



Two different function but their digitization is same.

From here, it is very difficult to recover back the original signal because digitization of Two different signal is same.

Solution

Anti Aliasing :

Aliasing can be reduced by smoothening (Low Pass filter) on the input function or image to attenuate the higher frequencies.

This process has to be done before the function is sampled because aliasing is an sampling issue that cannot be undone.

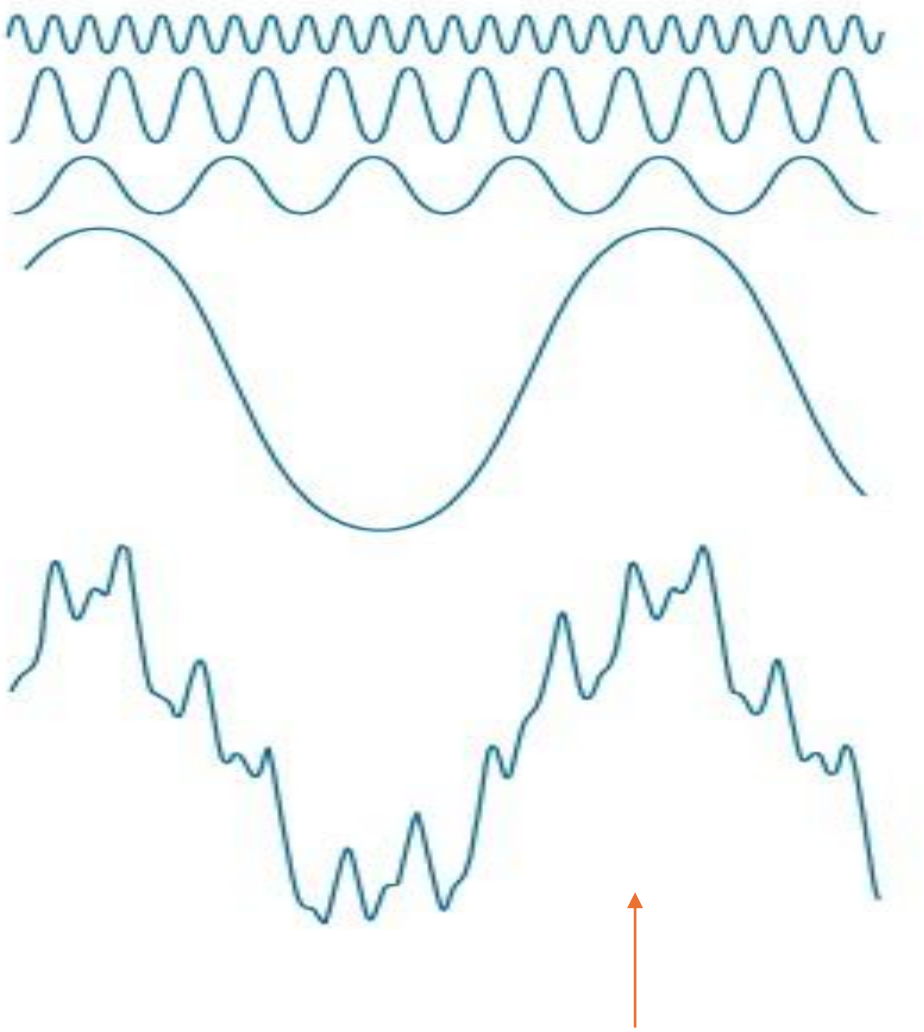
Frequency Domain Fundamentals: Fourier series

1. **Fourier Series:** Any periodic function can be expressed as a sum of sines and/or cosines of different frequencies, each multiplied by a different co-efficient.

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{j\frac{2\pi n}{T}t}$$

$$c_n = \frac{1}{T}\int_{-T/2}^{T/2} f(t) e^{-j\frac{2\pi n}{T}t} dt \qquad \text{for } n = 0, \pm 1, \pm 2, \ldots$$

Where –
A function f(t) of a continuous variable, t that is periodic with period T.



This function is the sum of all the four above functions

Note: Here, Sine and cosines are the basis functions.

Frequency Domain Fundamentals: Fourier Transform

2. **Fourier Transform** : Functions those are not periodic (But whose area under the curve is finite) can be expressed as the integrals of sines and/or cosines of different frequencies each multiplied by a weighting function.

Fourier Transform Formula

$$F(\mu) = \int_{-\infty}^{\infty} f(t) e^{-j2\pi\mu t} dt$$

f(t) is a continuous function.

Inverse Fourier Transform Formula

$$f(t) = \int_{-\infty}^{\infty} F(\mu) e^{j2\pi\mu t} d\mu$$

$$f(t) \Leftrightarrow F(\mu)$$

It's a Fourier transform pair which indicate forward and Inverse Fourier transform is possible.

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

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

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

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

Note: Here, Sine and cosines are the basis functions.

Fourier Transform Working Example

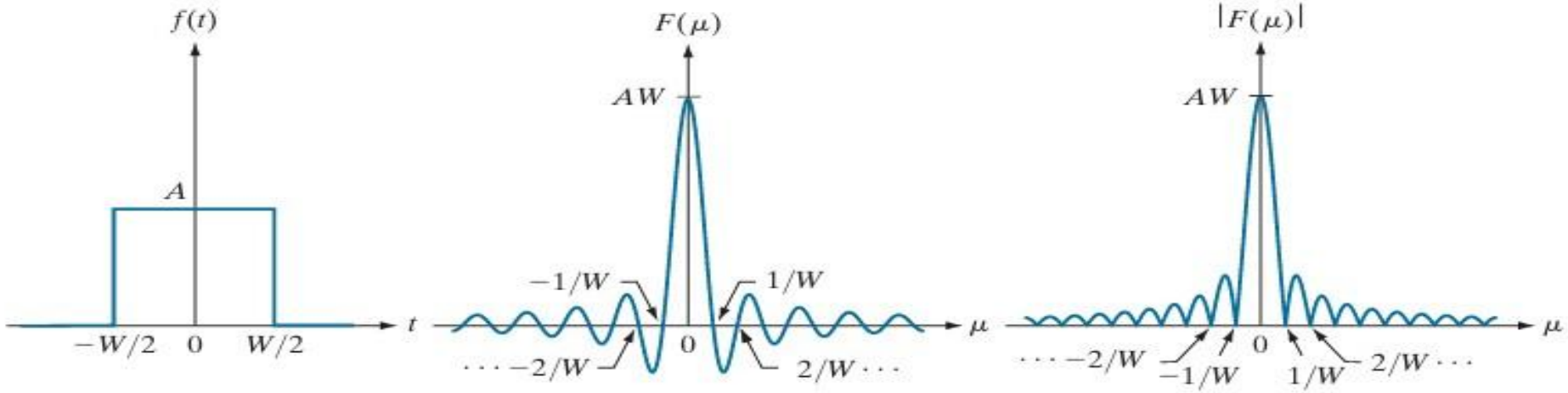
$$F(\mu) = \int_{-\infty}^{\infty} f(t)e^{-j2\pi\mu t}dt = \int_{-W/2}^{W/2} Ae^{-j2\pi\mu t}dt$$

$$= \frac{-A}{j2\pi\mu}\left[e^{-j2\pi\mu t}\right]_{-W/2}^{W/2} = \frac{-A}{j2\pi\mu}\left[e^{-j\pi\mu W} - e^{j\pi\mu W}\right]$$

$$= \frac{A}{j2\pi\mu}\left[e^{j\pi\mu W} - e^{-j\pi\mu W}\right]$$

$$= AW\frac{\sin(\pi\mu W)}{(\pi\mu W)}$$ (Signum function)

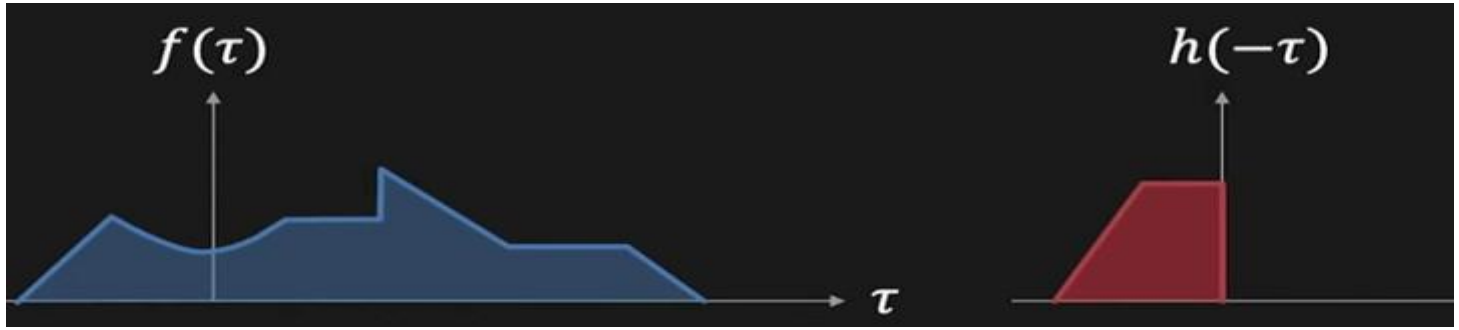
Note: A Functions that is expressed by either Fourier series or Fourier Transform could be reconstructed or recovered completely via an inverse process.



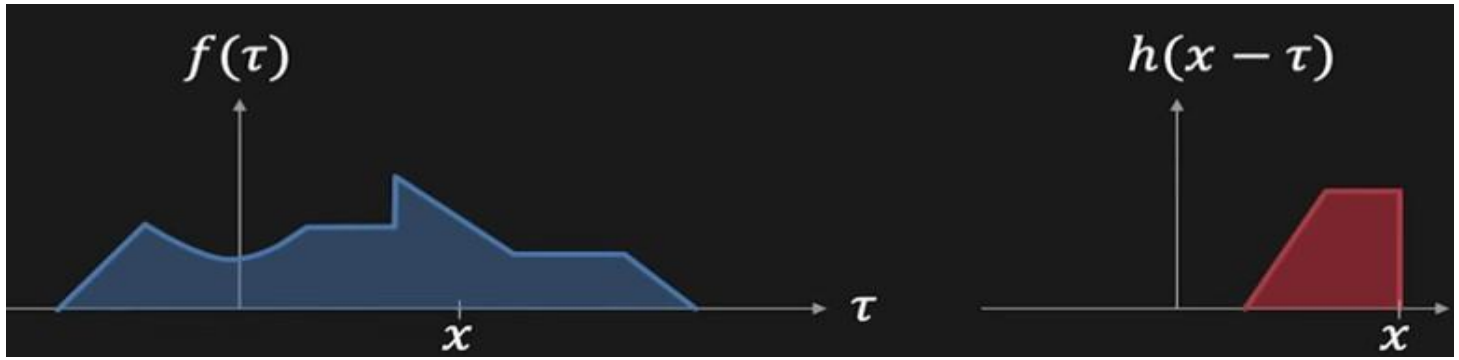
(a) A Box function (b) Fourier Transform (c) Spectrum

Convolution Theorem:

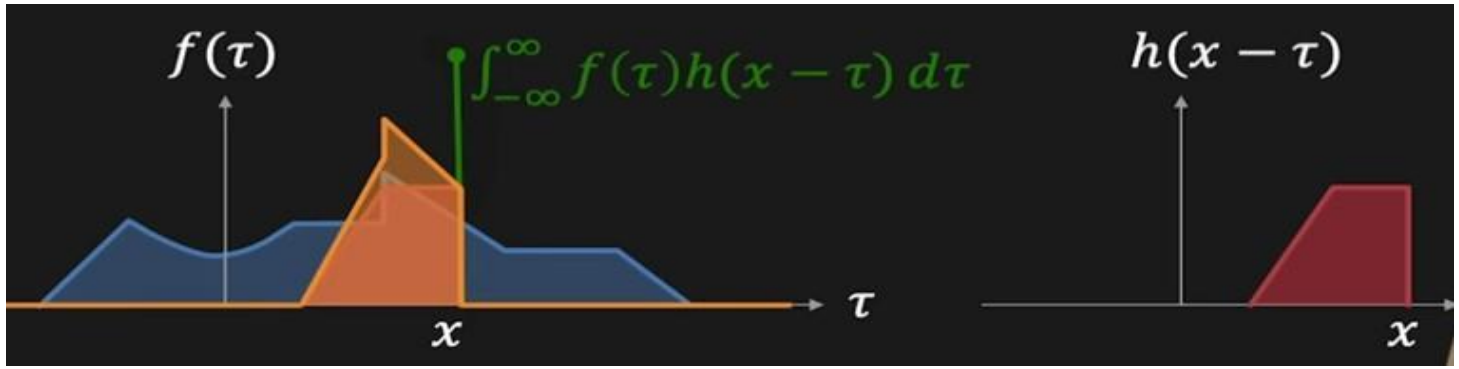
Revisit, what is the convolution.



Flip h(x)



shift h(x)



Overlay it on the f(x). Multiply f(x) and h(x)and perform integration, it will provide a single value to a particular point. This is know as convolution.

Convolution Theorem

Convolution Theorem: Convolution of two signal or image in time domain will become the multiplication in the frequency domain.

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

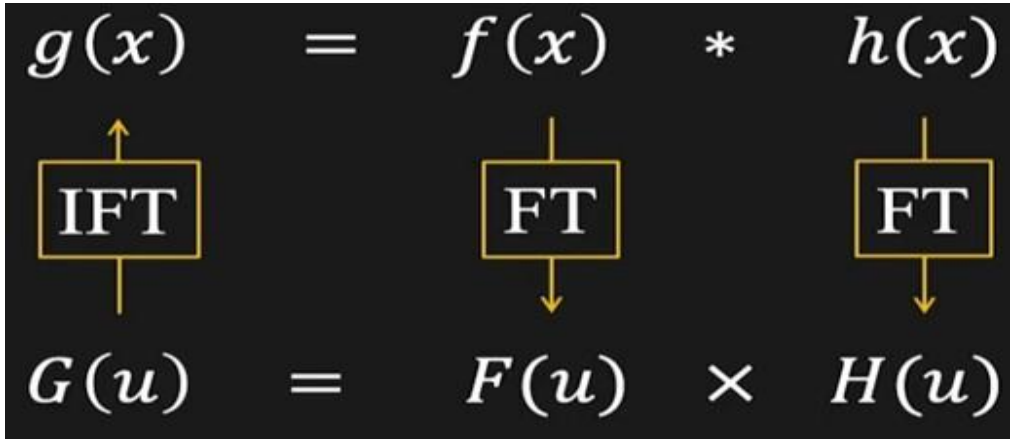
Fourier Transform of $g(x)$:

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

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

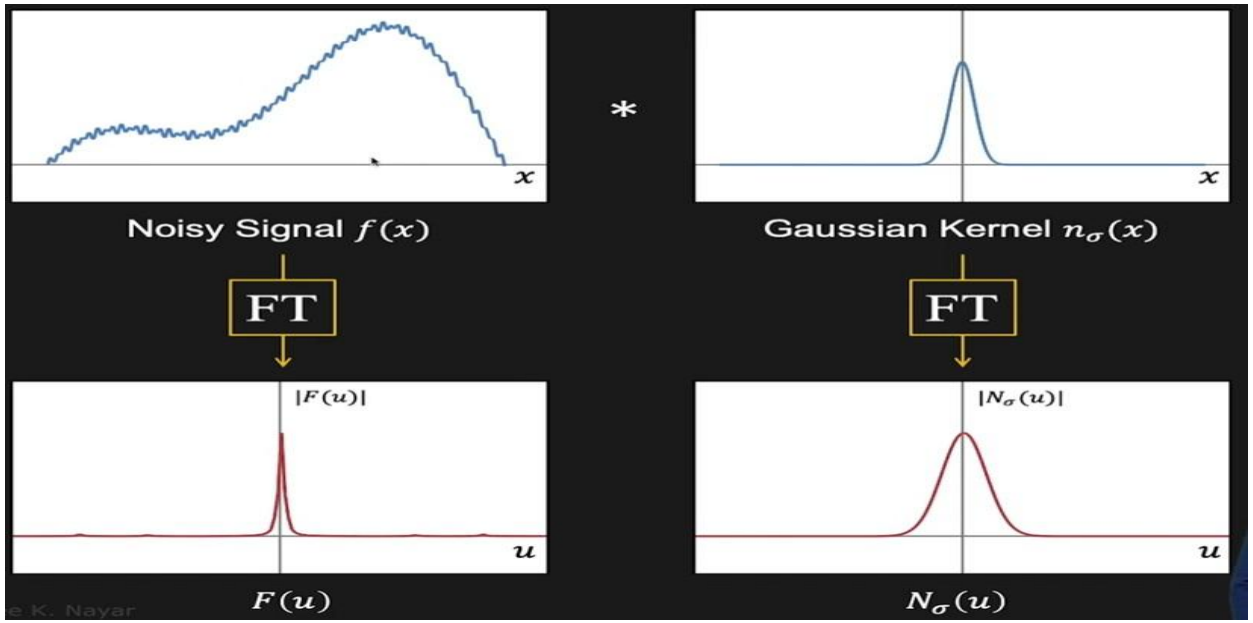
$$G(u) = \underbrace{\int_{-\infty}^{\infty} f(\tau)e^{-i2\pi u\tau}d\tau}_{F(u)} \underbrace{\int_{-\infty}^{\infty} h(x-\tau)e^{-i2\pi u(x-\tau)}dx}_{H(u)}$$

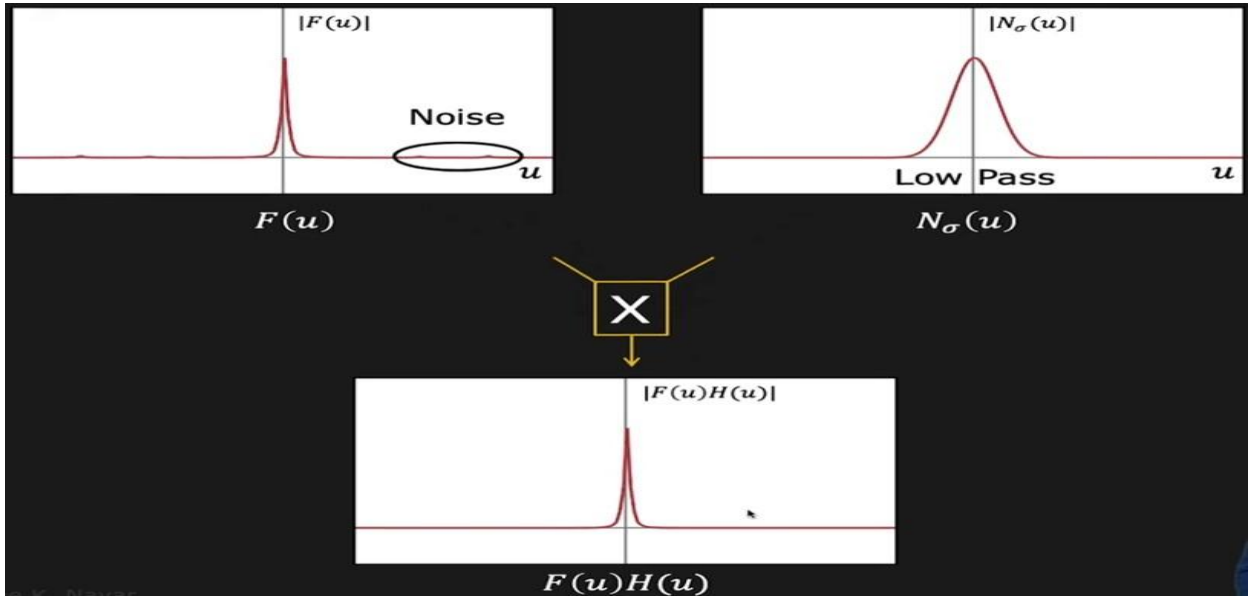
Spatial Domain		Frequency Domain
$g(x) = f(x) * h(x)$ Convolution	$\longleftrightarrow$	$G(u) = F(u)\,H(u)$ Multiplication
$g(x) = f(x)\,h(x)$ Multiplication	$\longleftrightarrow$	$G(u) = F(u) * H(u)$ Convolution

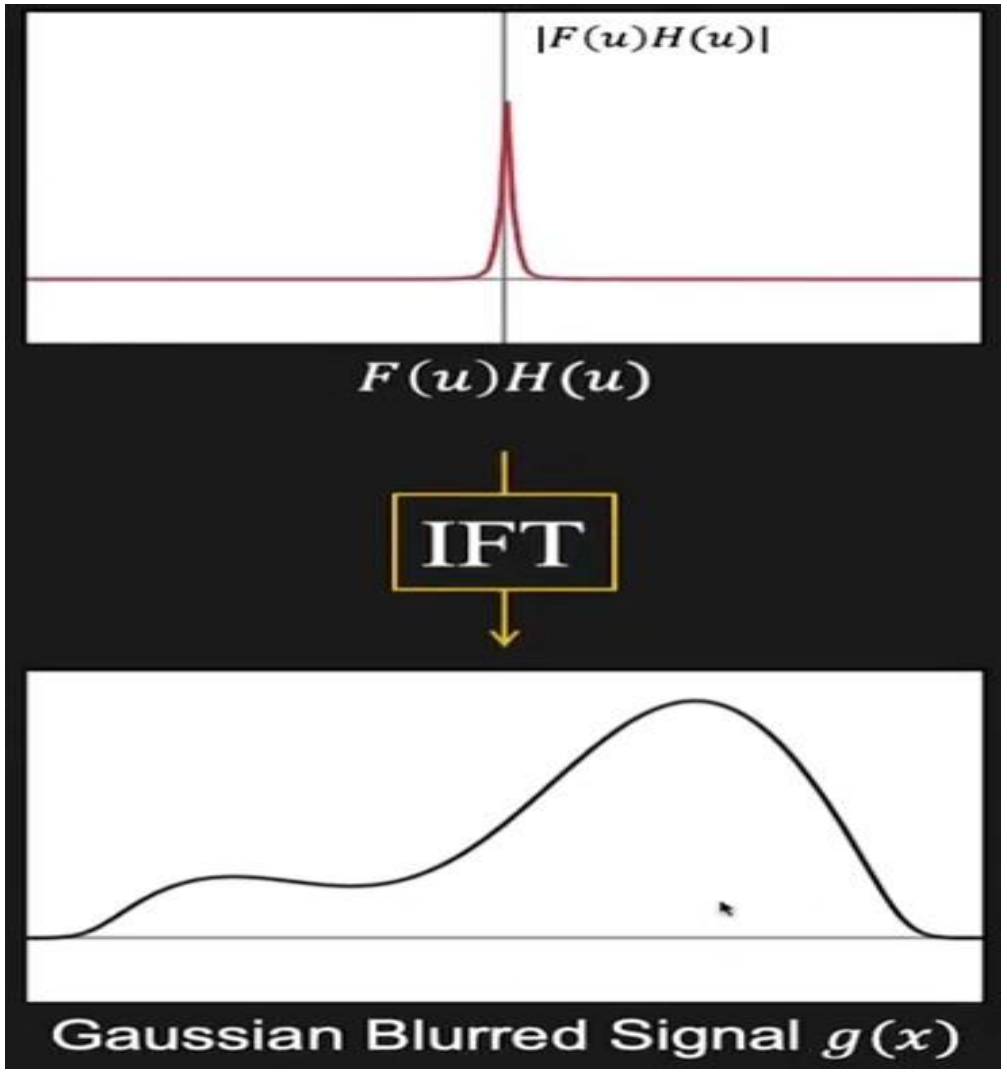


Convolution Theorem

Fourier Transform Working Example





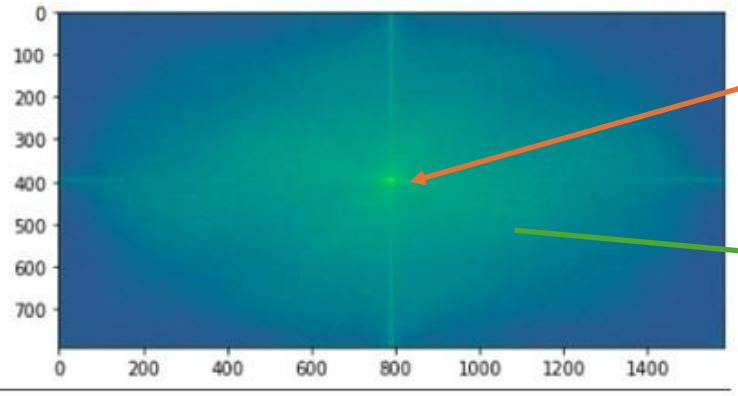


Interpretation image in the Frequency Domain: Fourier Transform

Fourier Transform: It's an image enhancement tool, which is used to decompose an image into its sine and cosine components.

Low frequencies : It represent smooth transition in the image.

High frequencies : It capture the rapid changes in the image. Like edges and texture.



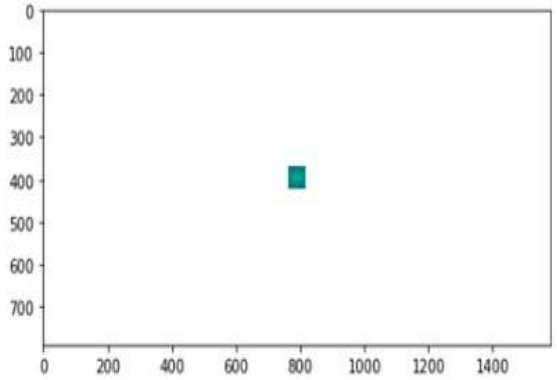
Center shows low Frequency component

High Frequency component

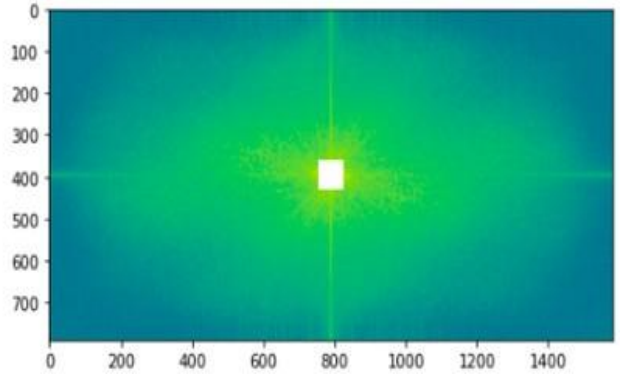
Input Image

F.T. of input Image with heatmap

Keep only low frequency components: image is blurred

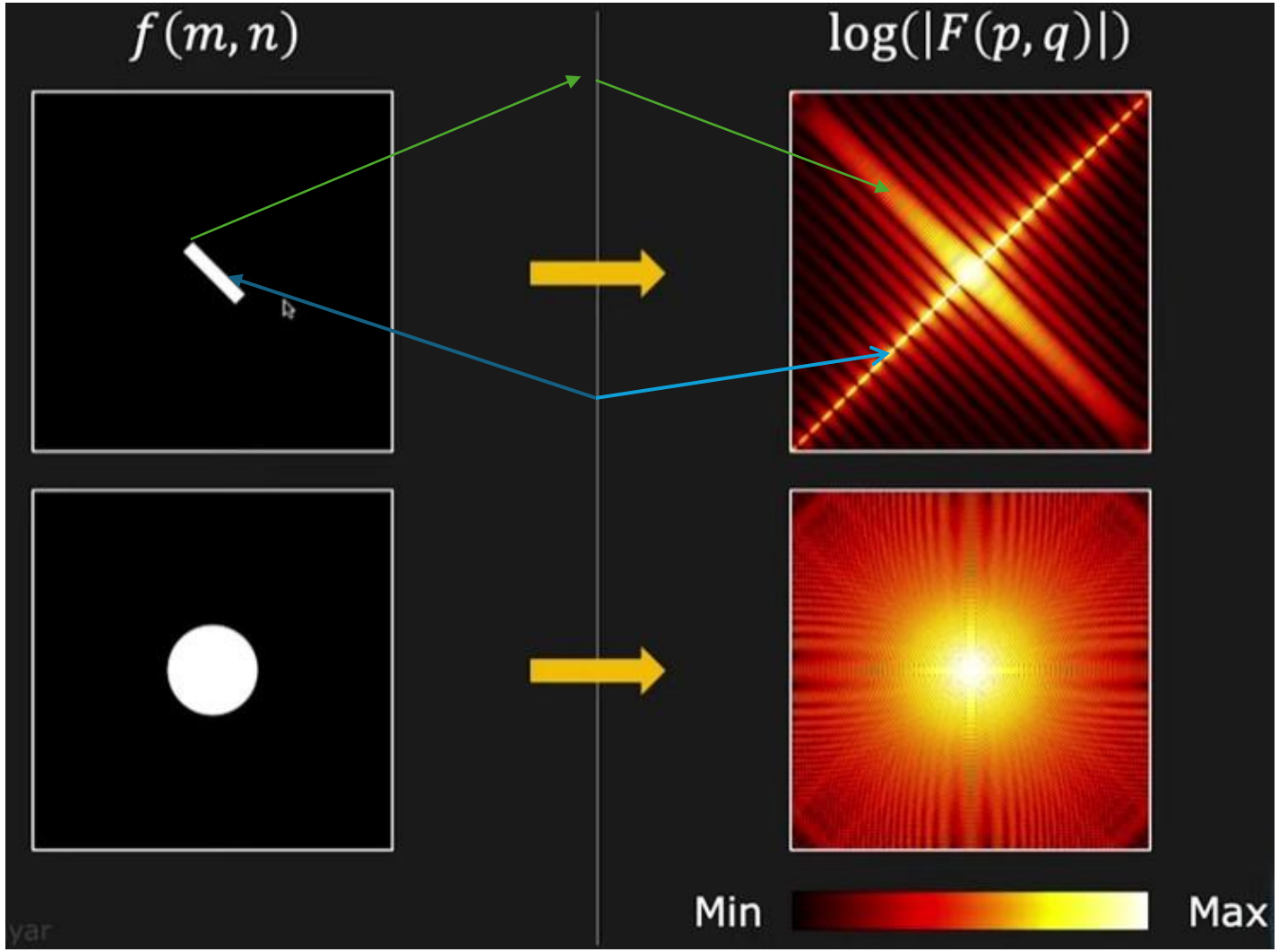


Keep only high frequency components

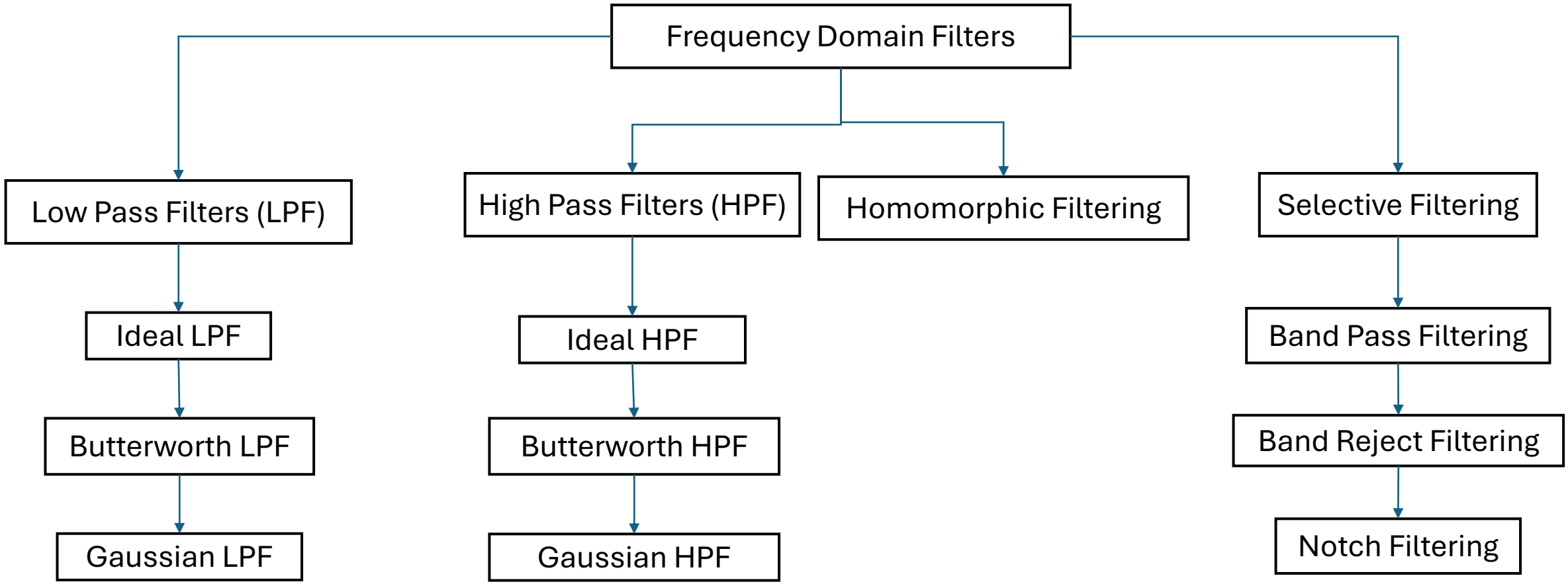


Frequency Domain Filtering: Application

Heatmap of original image and its fourier transform



Frequency Domain Filters



Frequency Domain Filters: Ideal Low Pass Filter (ILPF)

A 2D filter which passes all the frequencies within a circle of radius from the origin and cut off or attenuate all the frequencies which are outside to this circle.

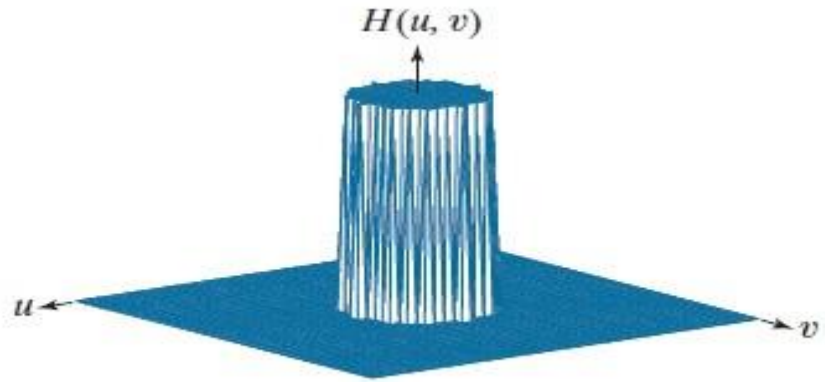
It is specified by the transfer function H(u,v)

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

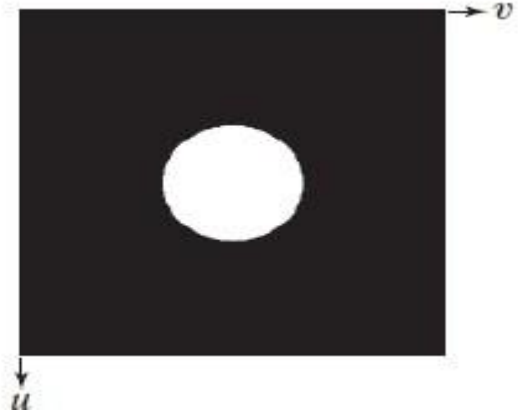
Where - Do = Positive Constant

D(u,v) = distance between a point (u , v) in the frequency domain and the center.

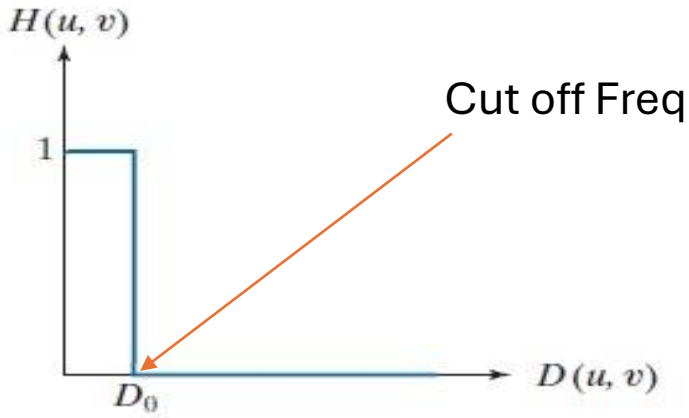
Frequency Domain Filters: Ideal Low Pass Filter (ILPF)



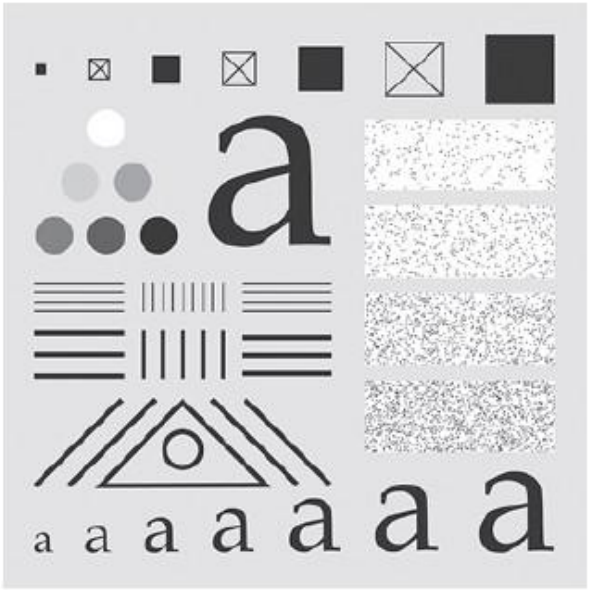
(a) Ideal LPF Transfer function Plot



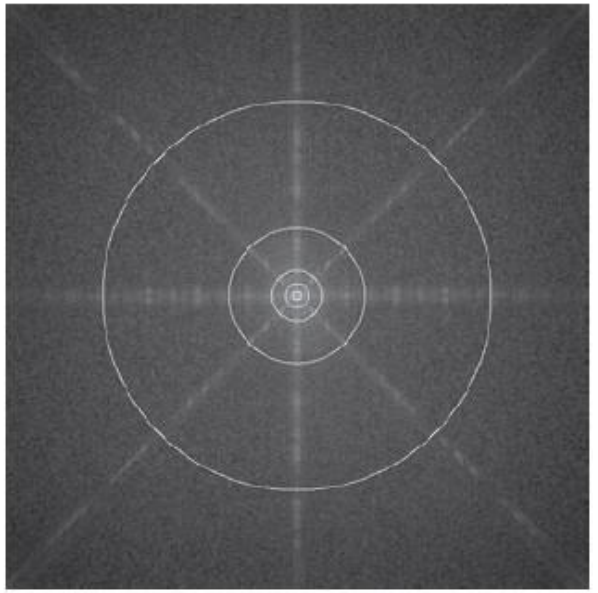
(b) Function displayed as an image



(c) Radial Cross Section

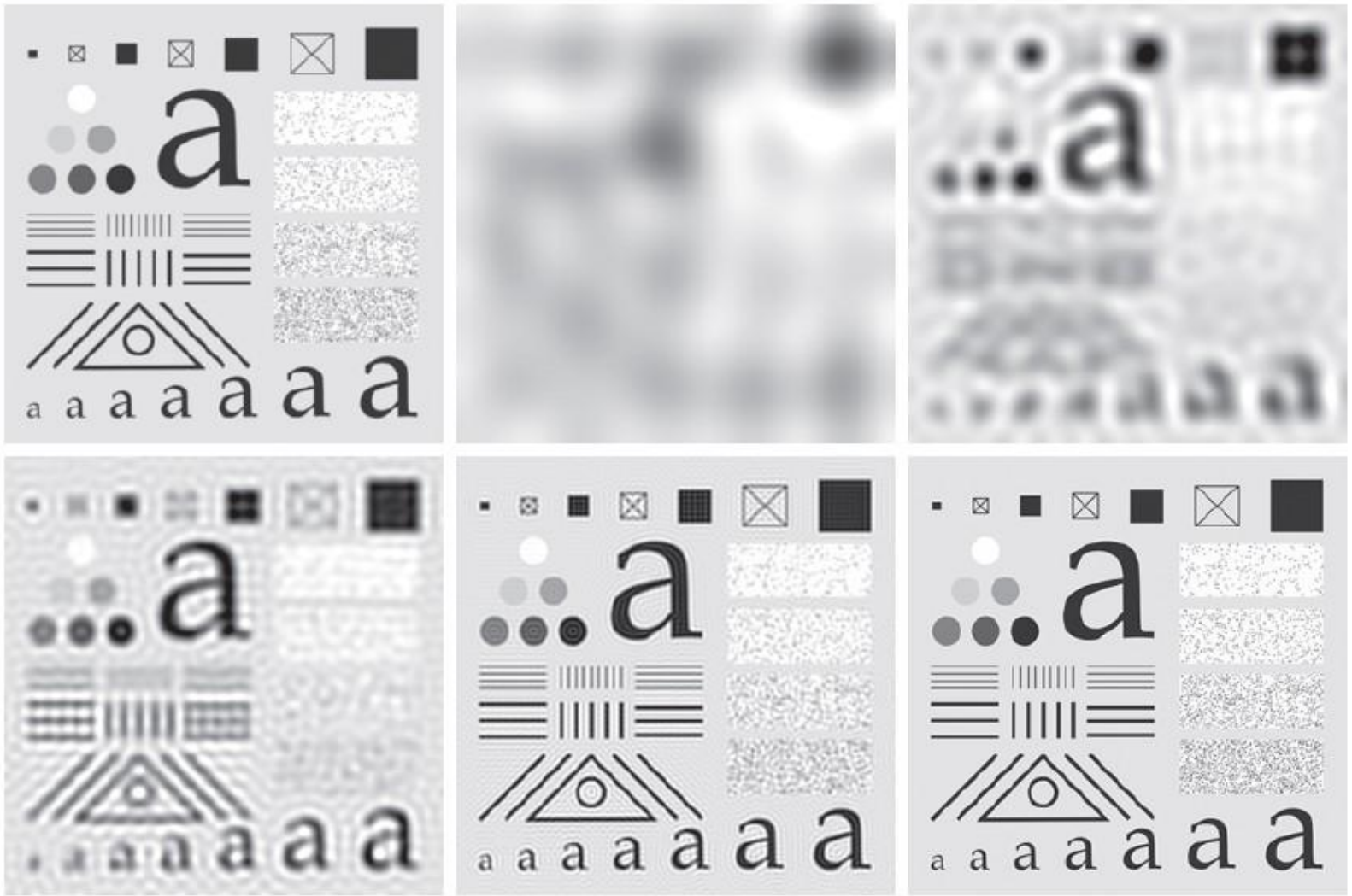


Test Pattern image



Circle with radi 10,30,60,160,460

Frequency Domain Filters: Ideal Low Pass Filter (ILPF)



a – Original Image

b – ILPF with cut off Frequency set at radii value - **10**

c – ILPF with cut off Frequency set at radii value - **30**

d – ILPF with cut off Frequency set at radii value - **60**

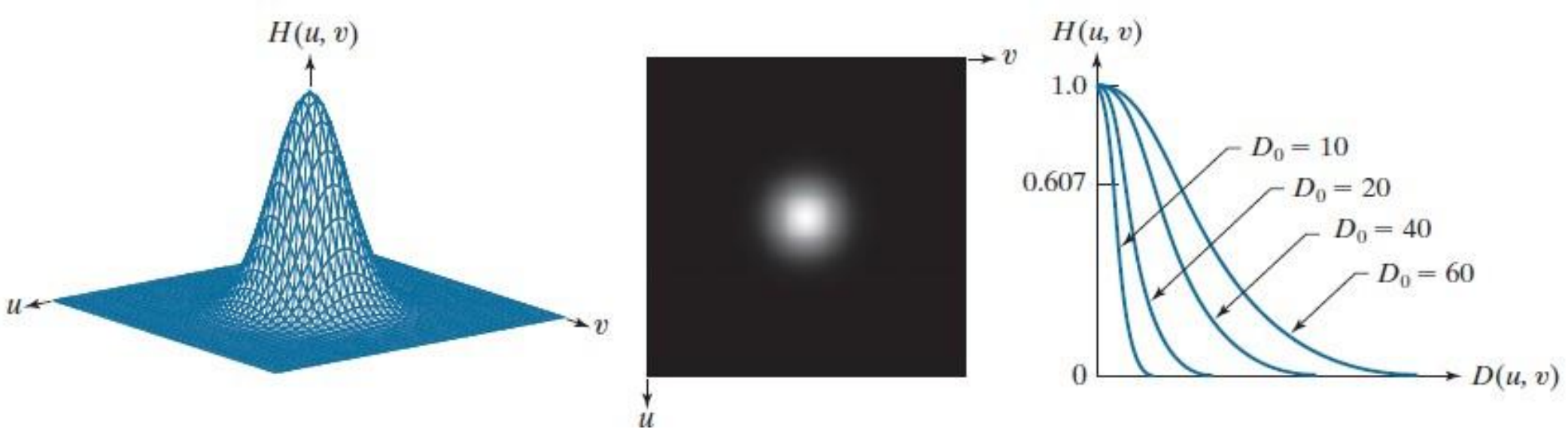
e – ILPF with cut off Frequency set at radii value - **160**

f – ILPF with cut off Frequency set at radii value - **460**

Frequency Domain Filters: Gaussian Low Pass Filter (GLPF)

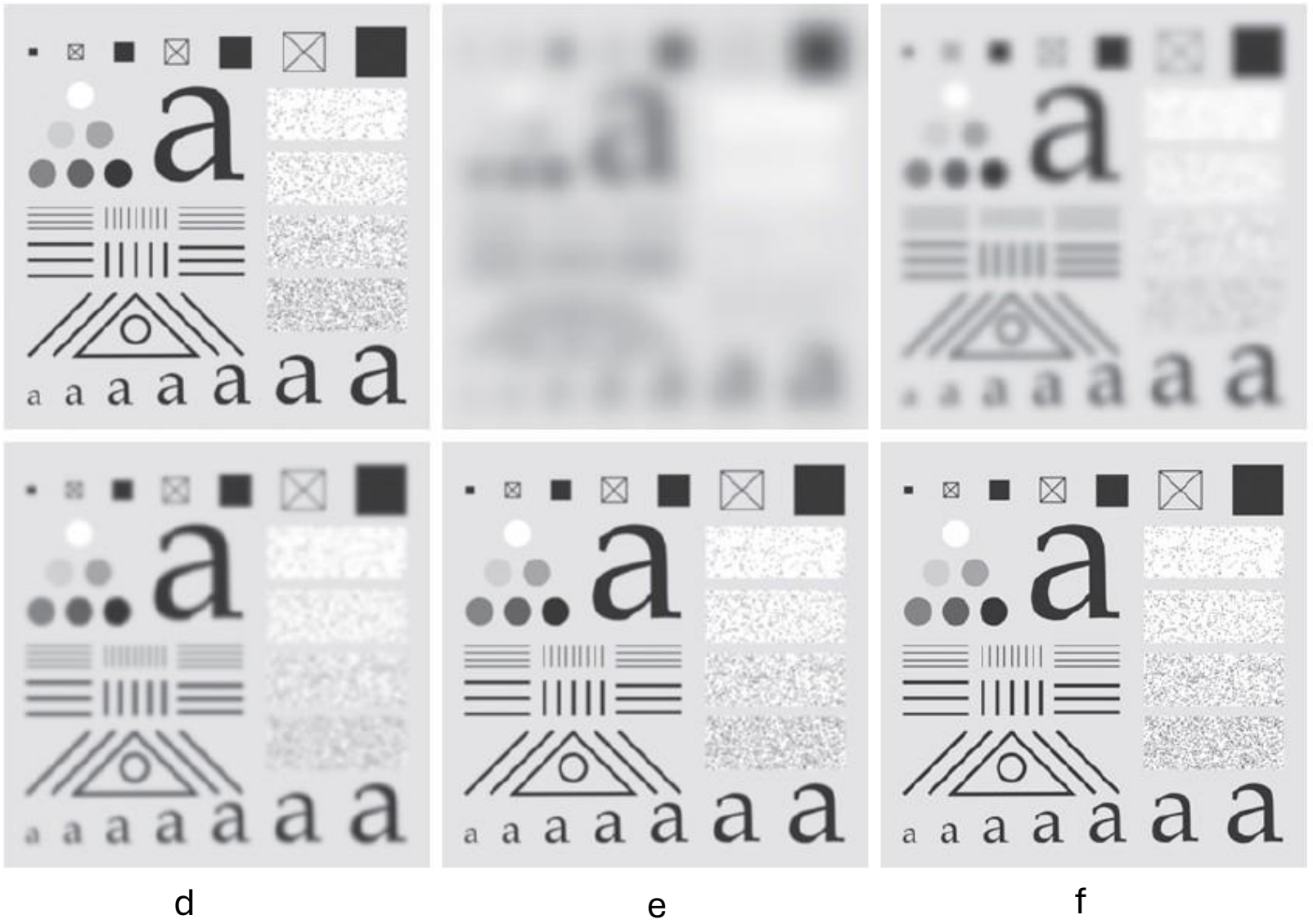
It is specified by the transfer function H(u,v)

$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$



(a) Gaussian LPF Transfer function Plot (b) Function displayed as an image (c) Radial Cross Section with various value of D0

Frequency Domain Filters: Gaussian Low Pass Filter (GLPF)



a – Original Image

b – GLPF with cut off Frequency set at radii value - **10**

c – GLPF with cut off Frequency set at radii value - **30**

d – GLPF with cut off Frequency set at radii value - **60**

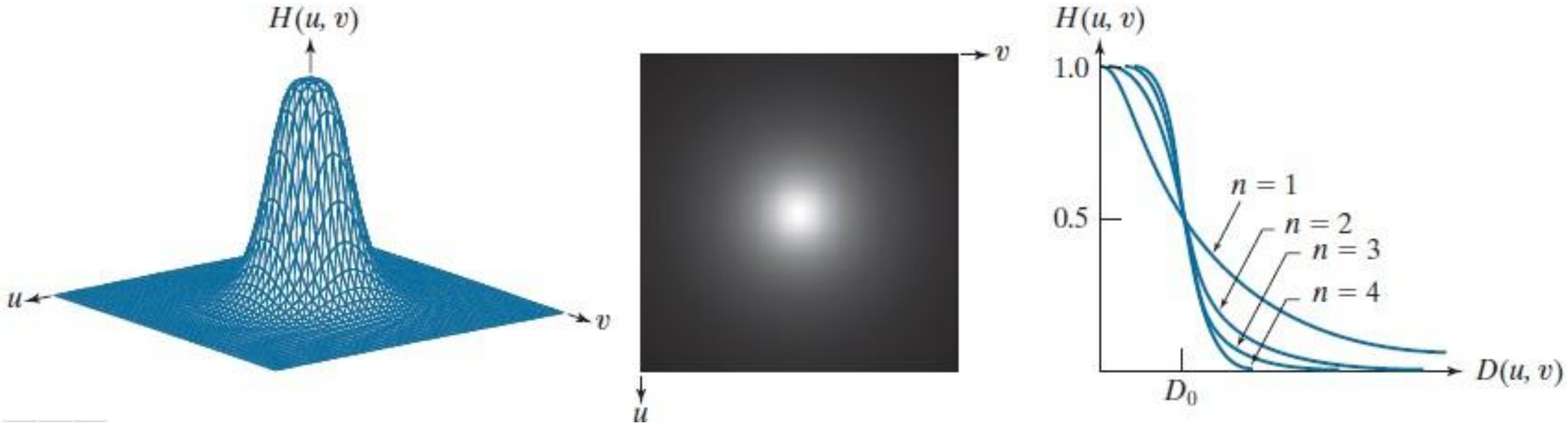
e – GLPF with cut off Frequency set at radii value - **160**

f – GLPF with cut off Frequency set at radii value - **460**

Frequency Domain Filters: Butterworth Low Pass Filter (BLPF)

It is specified by the transfer function H(u,v)

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

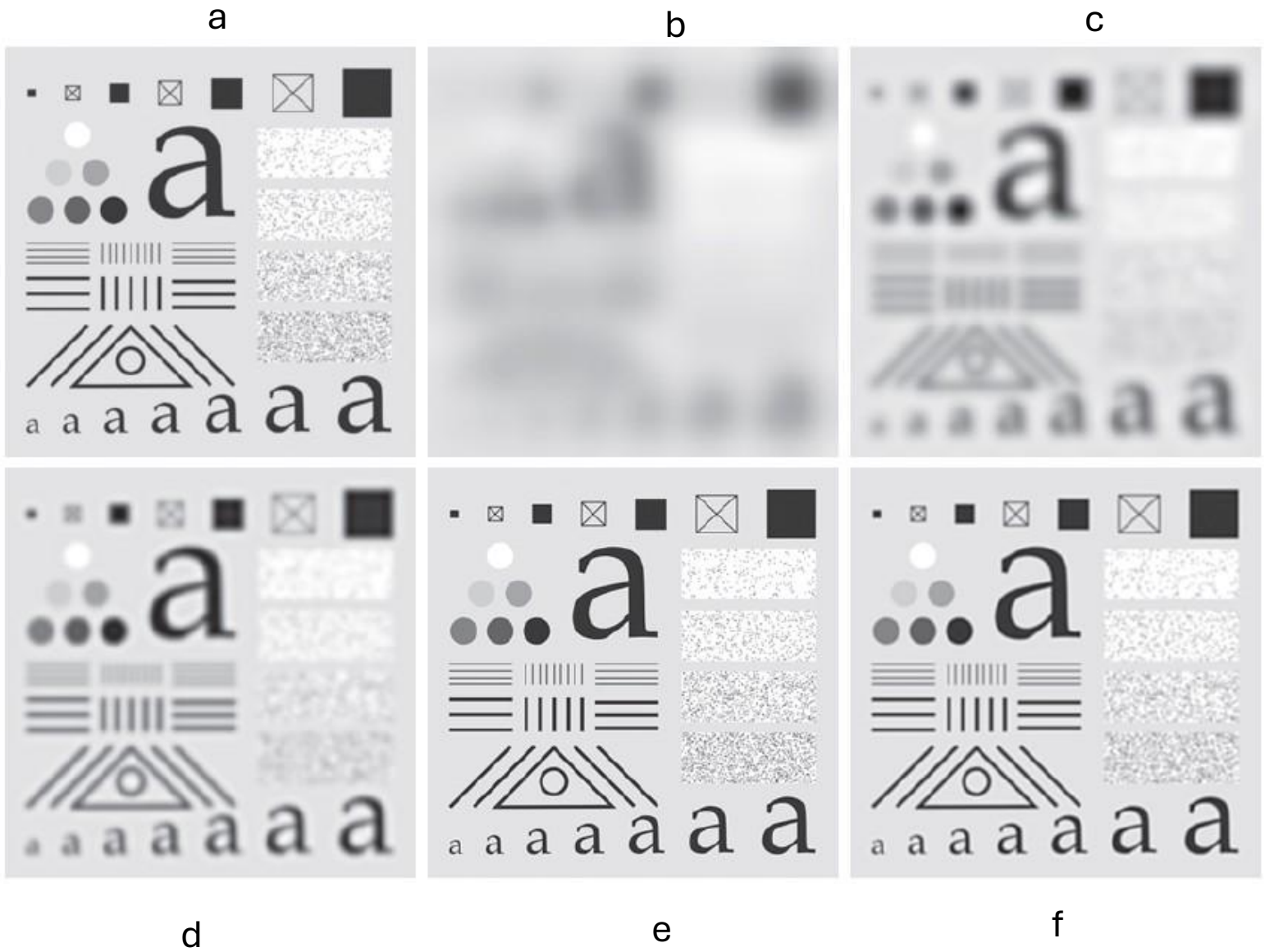


(a) Gaussian LPF Transfer function Plot

(b) Function displayed as an image

(c) Radial Cross Section with orders 1 to 4.

Frequency Domain Filters: Butterworth Low Pass Filter (BLPF)



a – Original Image

b – BLPF with cut off Frequency set at radii value - **10**

c – BLPF with cut off Frequency set at radii value - **30**

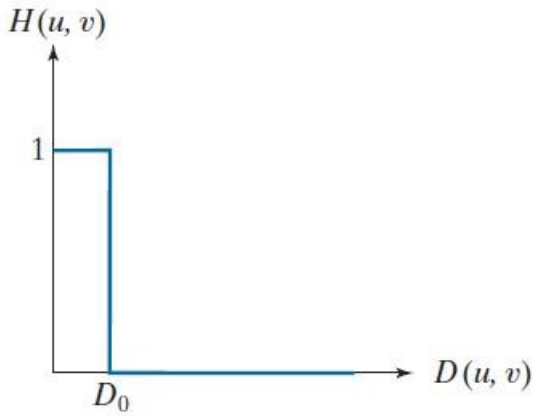
d – BLPF with cut off Frequency set at radii value - **60**

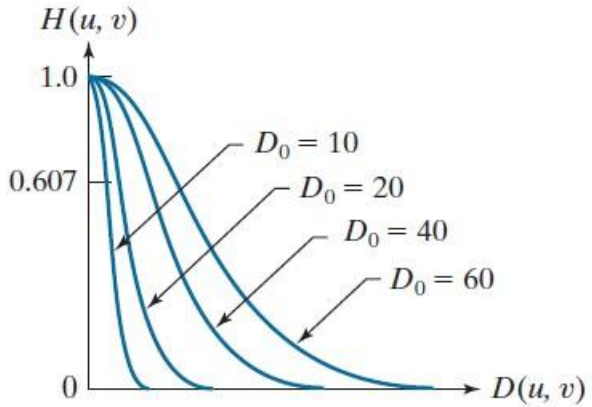
e – BLPF with cut off Frequency set at radii value - **160**

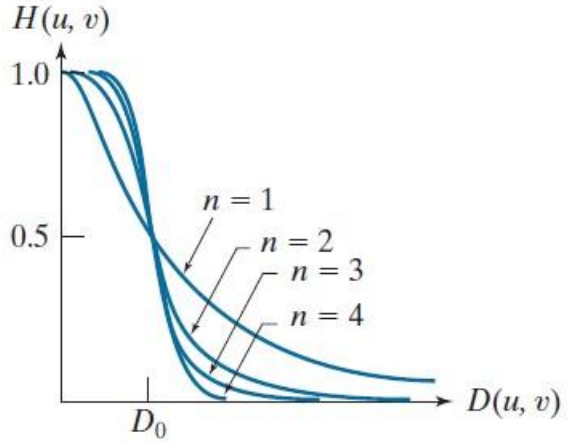
f – BLPF with cut off Frequency set at radii value - **460**

Comparative Analysis Between ILPF, GLBF and BLPF

Ideal	Gaussian	Butterworth
$H(u,v)=\begin{cases}1 & \text{if } D(u,v)\le D_0\\ 0 & \text{if } D(u,v)>D_0\end{cases}$	$H(u,v)=e^{-D^2(u,v)/2D_0^2}$	$H(u,v)=\dfrac{1}{1+\left[D(u,v)/D_0\right]^{2n}}$







Note:

The shape of the Butterworth filter is controlled by the filter order.

If n is large then Butterworth filter approaches to ILPF.

if n is small then Butterworth filter approaches to GLPF.

Frequency Domain Filters: Image Sharpening using High Pass Filter

High Pass Filter

Image Sharpening can be achieved in the frequency domain by passing the high frequency components (i.e Edges or other sharp transitions) and attenuate the low frequency components.

Ideal	Gaussian	Butterworth
$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$	$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$	$H(u,v) = \frac{1}{1 + [D_0/D(u,v)]^{2n}}$

Where

D_0 = Cut off frequency

n = order

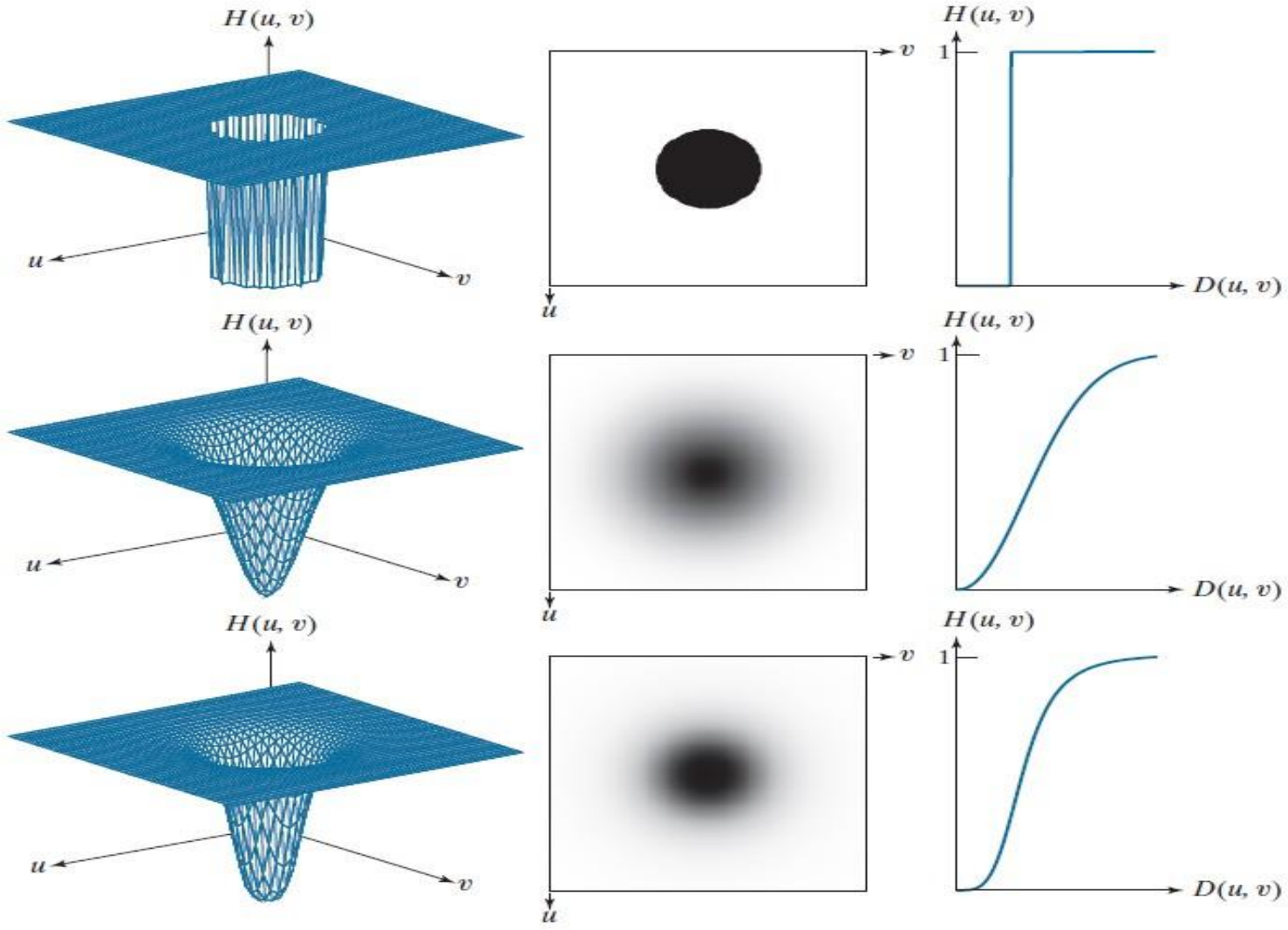
Frequency Domain Filters: Image Sharpening using High PF

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

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

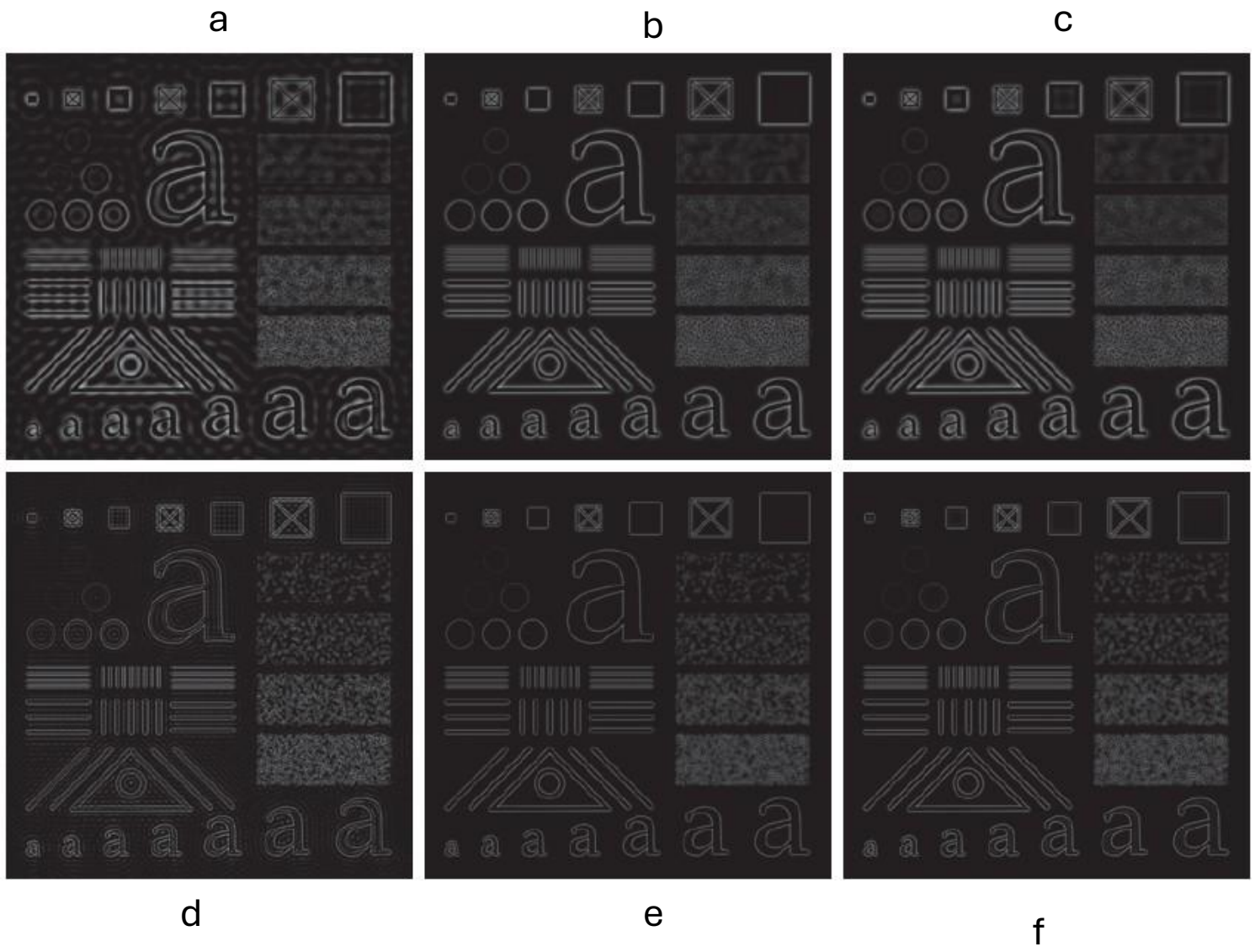
$$H(u,v) = \frac{1}{1 + [D_0/D(u,v)]^{2n}}$$

Transfer Function



(a) HPF Transfer function Plot (b) Function displayed as an image (c) Radial Cross Section

Frequency Domain Filters: Image Sharpening using High PF



Filtered with (a) IHPF, (b) GHPF, (c) BHPF with D0 =60

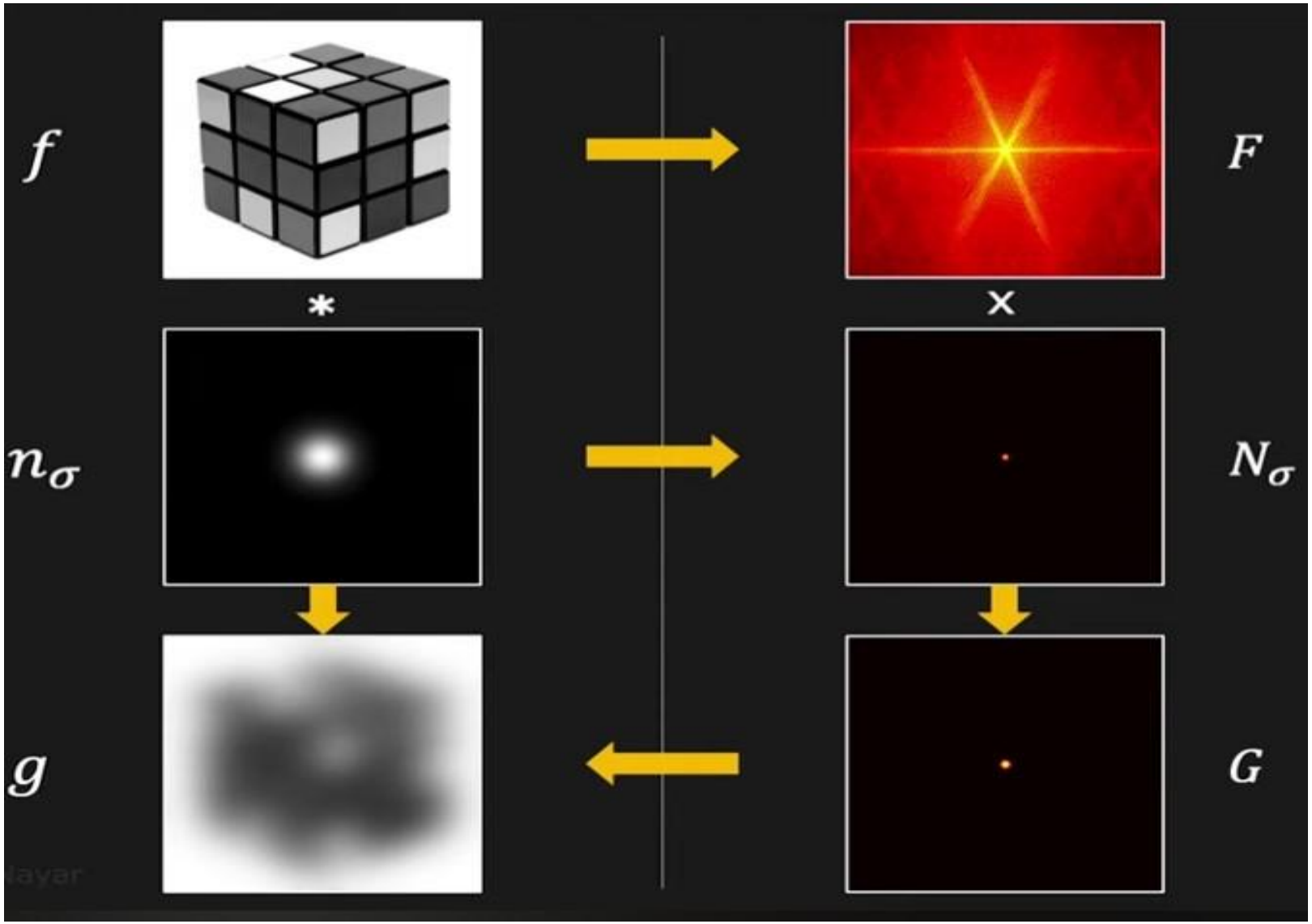
Filtered with (d) IHPF, (e) GHPF, (f) BHPF with D0 =160

Gaussian Smoothening

Original Image

Gaussian Smoothening kernel

Inverse FT



Original Image FT

Gaussian Smoothening kernel FT

Multiplication of Original image FT and Kernel FT.
Here output only contains low frequency terms.

Frequency Domain Filters: Working Examples

1-D Discrete Fourier Transform (DFT) and Inverse Discrete Fourier Transform (IDFT)

DFT

$$F(u) = \sum_{x=0}^{M-1} f(x) e^{-j2\pi ux/M} \qquad u = 0, 1, 2, \ldots, M-1$$

IDFT

$$f(x) = \frac{1}{M} \sum_{u=0}^{M-1} F(u) e^{j2\pi ux/M} \qquad x = 0, 1, 2, \ldots, M-1$$

1-D DFT and Inverse DFT working example

DFT

$$F(u)=\sum_{x=0}^{M-1} f(x)e^{-j2\pi ux/M} \qquad u=0,1,2,\ldots,M-1$$

IDFT

$$f(x)=\frac{1}{M}\sum_{u=0}^{M-1} F(u)e^{j2\pi ux/M} \qquad x=0,1,2,\ldots,M-1$$

$f(k) = \sum_{x=0}^{N-1} f(x)\, e^{-j2\pi kx/N}$

where $k = 0, 1, 2, \ldots N-1$

IDFT

$f(x) = \frac{1}{N}\sum_{k=0}^{N-1} F(k)\, e^{j2\pi kx/N}$

where $x = 0, 1, 2, \ldots N-1$

Q.1 Calculate DFT $f(x) = \{1, 0, 0, 1\}$

↳ 0 1 2 3 → Indices.

$f(k) = \sum_{x=0}^{N-1} f(x)\, e^{-j2\pi kx/N}$

Total 4 values so that $N = 4$

$f[k] = \sum_{x=0}^{3} f(x)\, e^{-j2\pi kx/4}$

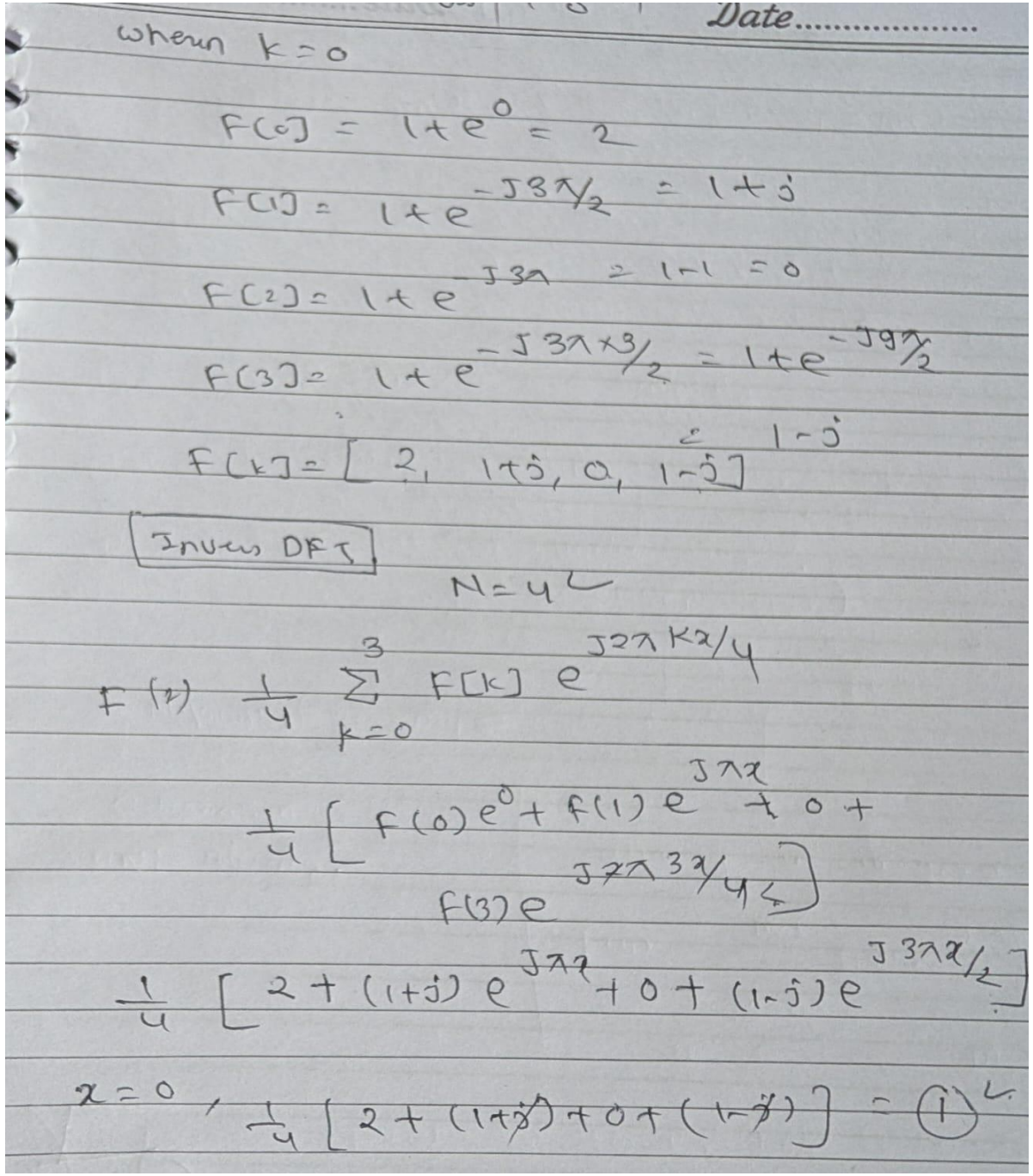
$f(0)\, e^{0} + f(1)\, e^{-j2\pi k/4} + f(2)\, e^{-j2\pi k\times 2/4} + f(3)\, e^{-j2\pi k\times 3/4}$

$1 + 0 + 0 + 1\, e^{-j3\pi k/2}$

$f(k) = 1 + e^{-j3\pi k/2}$

Teacher's Sign .....

1-D DFT and Inverse DFT working example



when k = 0

$F[0] = 1 + e^{0} = 2$

$F[1] = 1 + e^{-J3\pi/2} = 1 + j$

$F[2] = 1 + e^{-J3\pi} = 1 - 1 = 0$

$F[3] = 1 + e^{-J3\pi \times 3/2} = 1 + e^{-J9\pi/2}$

$\therefore 1 - j$

$F[k] = [2,\ 1+j,\ 0,\ 1-j]$

Inverse DFT

$N = 4$

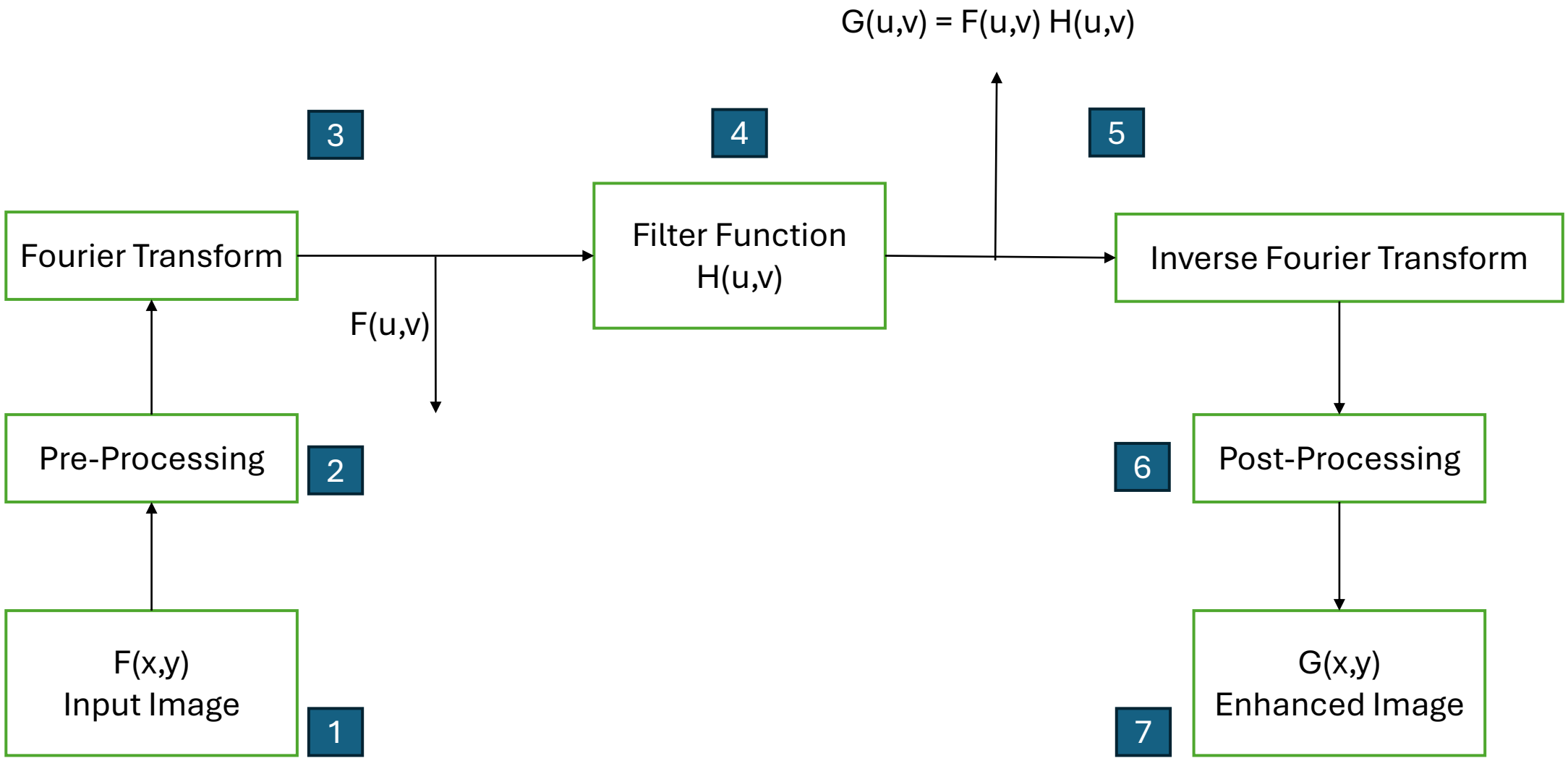
$$f(x) \frac{1}{4} \sum_{k=0}^{3} F[k]\, e^{J2\pi kx/4}$$

$$\frac{1}{4}\left[f(0)e^{0} + f(1)e^{J\pi x} + 0 + f(3)e^{J2\pi 3x/4} \right]$$

$$\frac{1}{4}\left[2 + (1+j)e^{J\pi x} + 0 + (1-j)e^{J3\pi x/2} \right]$$

$x = 0$, $\frac{1}{4}[2 + (1+j) + 0 + (1-j)] = (1)$

Frequency domain filtering : Flow Chart



Frequency Domain Filters: Working Examples

Question

For the spatial domain image, perform the frequency domain filtering using Ideal High pass filter with cut off frequency 0.5.

Input Image

1	0	1	0
1	0	1	0
1	0	1	0
1	0	1	0

Step 1: Multiply the input image by (-1)^x+y to shift the Centre from (0,0) to (2,2).

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

Note:
Input image is 4x4. so the centre is row/2, column/2 i.e 4/2 = 2, 4/2 = 2.

Frequency Domain Filters: Working Examples

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

(-1)^0,0 = 1, (-1)^0,1 = -1 (-1)^0,2 = 1 (-1)^0,3 = -1

(-1)^1,0 = -1, (-1)^1,1 = 1 (-1)^1,2 = -1 (-1)^1,3 = 1

(-1)^2,0 = 1, (-1)^2,1 = -1 (-1)^2,2 = 1 (-1)^2,3 = -1

(-1)^3,0 = -1, (-1)^3,1 = 1 (-1)^3,2 = -1 (-1)^3,3 = 1

Input Image

1	0	1	0
1	0	1	0
1	0	1	0
1	0	1	0

X

(-1)^(x+y)

1	-1	1	-1
-1	1	-1	1
1	-1	1	-1
-1	1	-1	1

=

1	0	1	0
-1	0	-1	0
1	0	1	0
-1	0	-1	0

Note: Its pixel-to-pixel multiplication **NOT** a matrix multiplication.

Frequency Domain Filters: Working Examples

Step 2: Compute the DFT of the image.

F(U,V) DFT = Kernel x f(x,y) x Kernel^{T}

Kernel

1	1	1	1
1	-j	-1	j
1	-1	1	-1
1	j	-1	-j

X

Input Image

1	0	1	0
-1	0	-1	0
1	0	1	0
-1	0	-1	0

X

Kernel Transpose

1	1	1	1
1	-j	-1	j
1	-1	1	-1
1	j	-1	-j

=

F(U,V) DFT =

0	0	0	0
0	0	0	0
16	0	16	0
0	0	0	0

Note: Matrix multiplication. Consider only real values.

Frequency Domain Filters: Working Examples

Step 2: Compute the distance between each value and the centre.

$\sqrt{(x-u)^2+(y-v)^2}$ u, v = 2,2

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

D(u,v) =

2.82	2.23	2	2.23
2.23	1.14	1	1.41
2	1	0	1
2.23	1.41	1	1.41

Filter Function H(u,v)

H(u,v) =

D0 = 0.5

1	1	1	1
1	1	1	1
1	1	0	1
1	1	1	1

Note: IHPF – Any value greater than 0.5 will be 1 else is 0.

Frequency Domain Filters: Working Examples

Step 3,4: G(u,v) = F(u,v) x H(u,v)

0	0	0	0
0	0	0	0
16	0	16	0
0	0	0	0

X

1	1	1	1
1	1	1	1
1	1	0	1
1	1	1	1

=

0	0	0	0
0	0	0	0
16	0	0	0
0	0	0	0

Note: Its pixel-to-pixel multiplication not a matrix multiplication.

Frequency Domain Filters: Working Examples

Step 5: Compute the IDFT of the image.

DFT = 1/4 (Kernel) x f(x,y) x $1/4(\text{Kernel}^{T})$

1	1	1	1
1	j	-1	-j
1	-1	1	-1
1	-j	-1	j

X

0	0	0	0
0	0	0	0
16	0	0	0
0	0	0	0

X

1	1	1	1
1	j	-1	-j
1	-1	1	-1
1	-j	-1	j

=

1/16

16	16	16	16
-16	-16	-16	-16
16	16	16	16
-16	-16	-16	-16

=

1	1	1	1
-1	-1	-1	-1
1	1	1	1
-1	-1	-1	-1

Note: Matrix multiplication.

Frequency Domain Filters: Working Examples

Step 6: Multiply the output image by (-1)^x+y to shift the Centre from (2,2) to (0,0)

1	1	1	1
-1	-1	-1	-1
1	1	1	1
-1	-1	-1	-1

X

(-1)^(x+y)

1	-1	1	-1
-1	1	-1	1
1	-1	1	-1
-1	1	-1	1

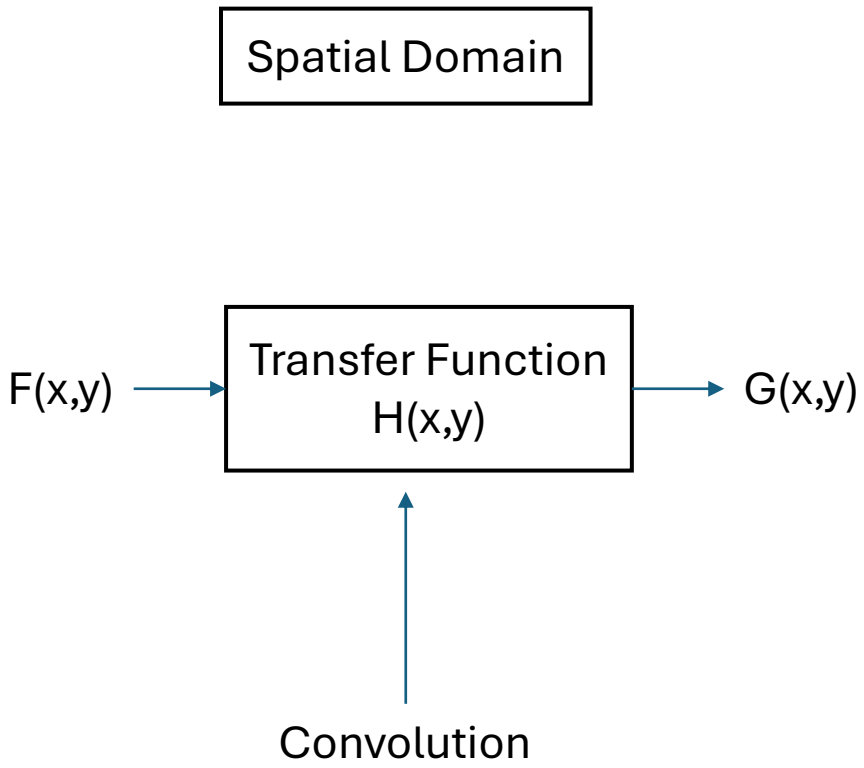
=

Final Output =

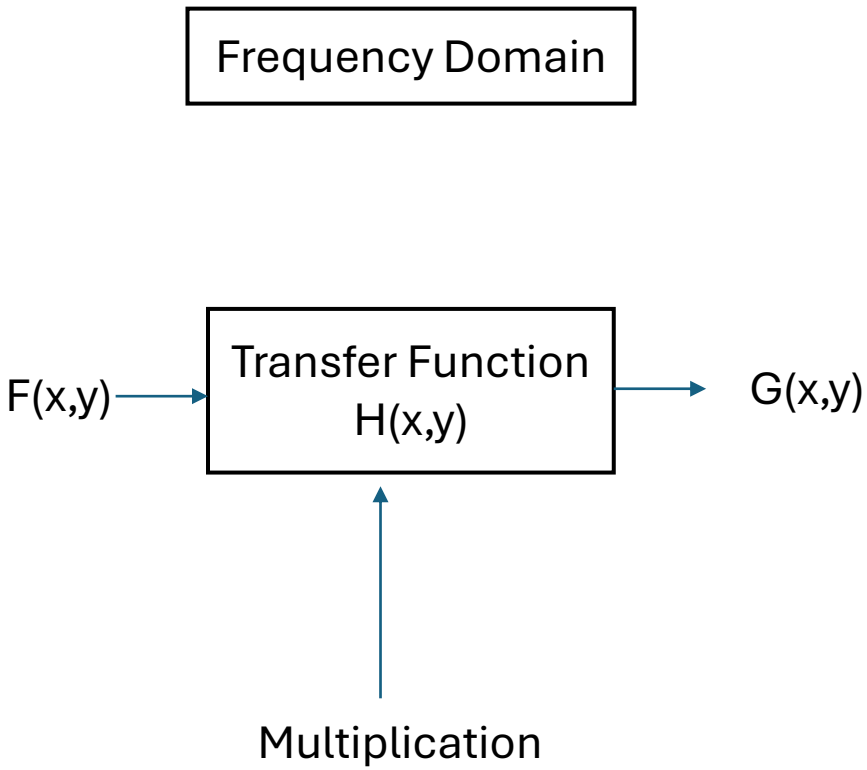
1	-1	1	-1
1	-1	1	-1
1	-1	1	-1
1	-1	1	-1

Note: Its pixel-to-pixel multiplication **NOT** a matrix multiplication.

Spatial domain and Frequency Domain Filters Analogy



G(x,y) = F(x,y) convolution H(x,y)



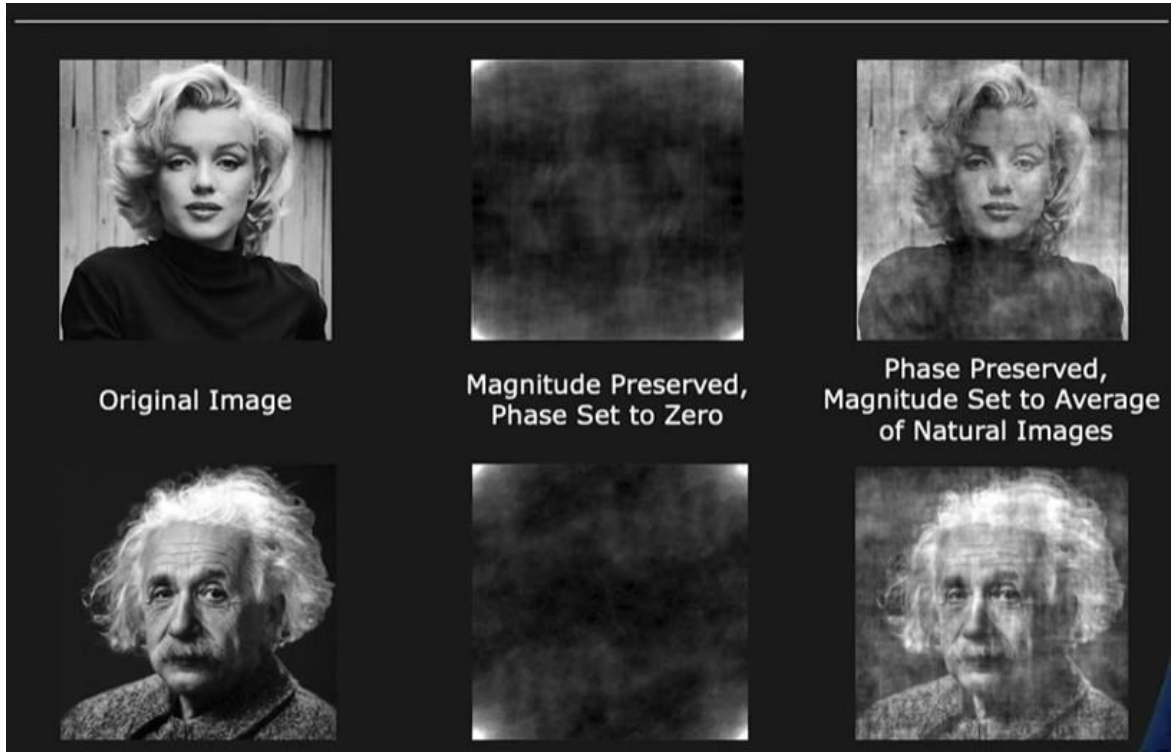
G(x,y) = F(x,y) Multiplication H(x,y)

Difference between spatial domain and frequency domain.

Feature	Spatial Domain Enhancement	Frequency Domain Enhancement
Domain of Operation	Image plane (pixels)	Frequency components (Fourier transform)
Primary Operations	Direct pixel manipulation, neighborhood filtering	Filtering in the frequency spectrum
Common Techniques	Histogram equalization, spatial filtering (smoothing, sharpening)	Low-pass filtering (blurring), high-pass filtering (sharpening), homomorphic filtering
Best For	Localized tasks (e.g., edge detection)	Global tasks (e.g., removing periodic noise)
Computational Speed	Fast for small kernels, slow for large kernels	Slower due to transforms for small kernels, faster for large kernels
Intuition	High (direct pixel manipulation)	Low (abstract frequency components)

Frequency domain: The process requires two computationally intensive steps: a forward Fourier transform and an inverse Fourier transform.

Frequency Domain Filtering: Application

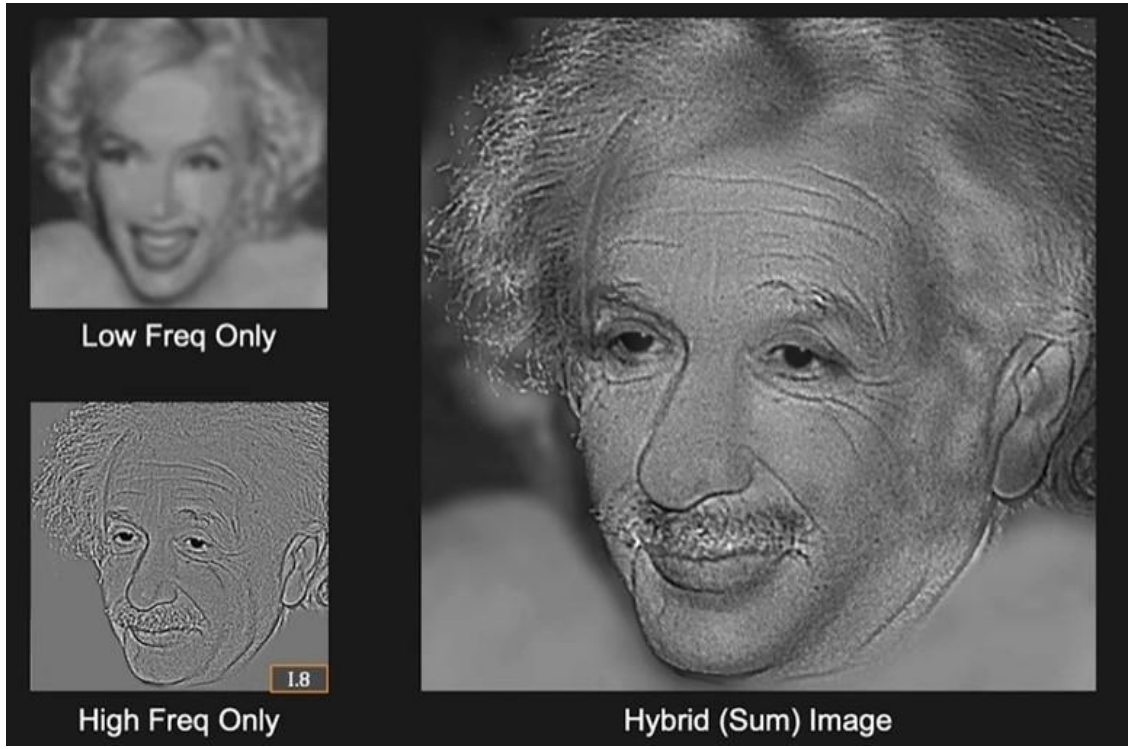


Importance of phase information

Image Morphing

Blur

Edges

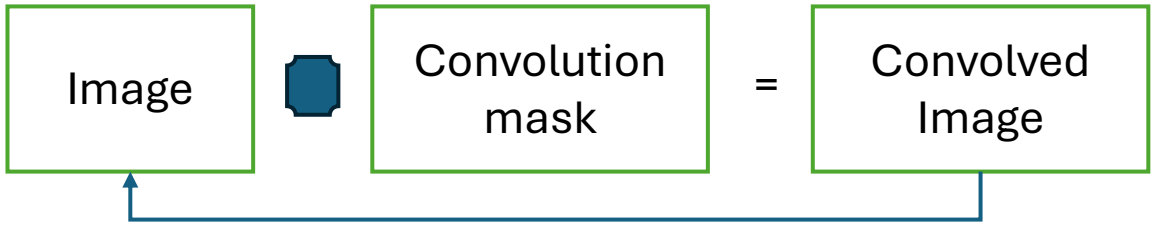


From closer look you will see albert Einstein

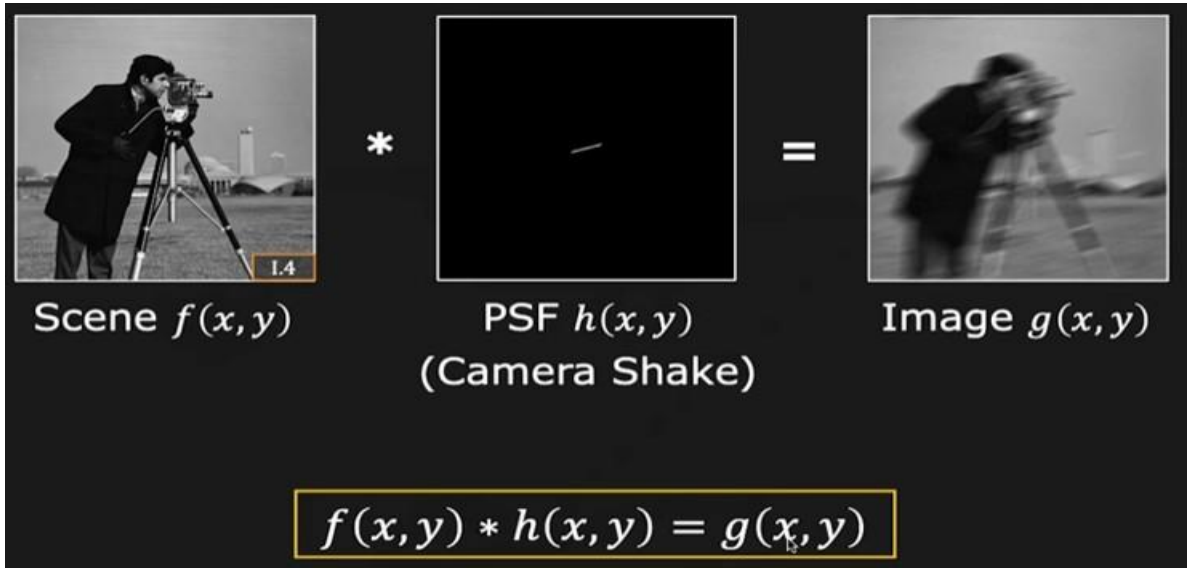
From far away you will see Marilyne.

When you close to the image, you could able to see high Details inside the image but as you see the same image from Far away you may loose high detail and you can see only low Frequency component.

Deconvolution in the frequency domain

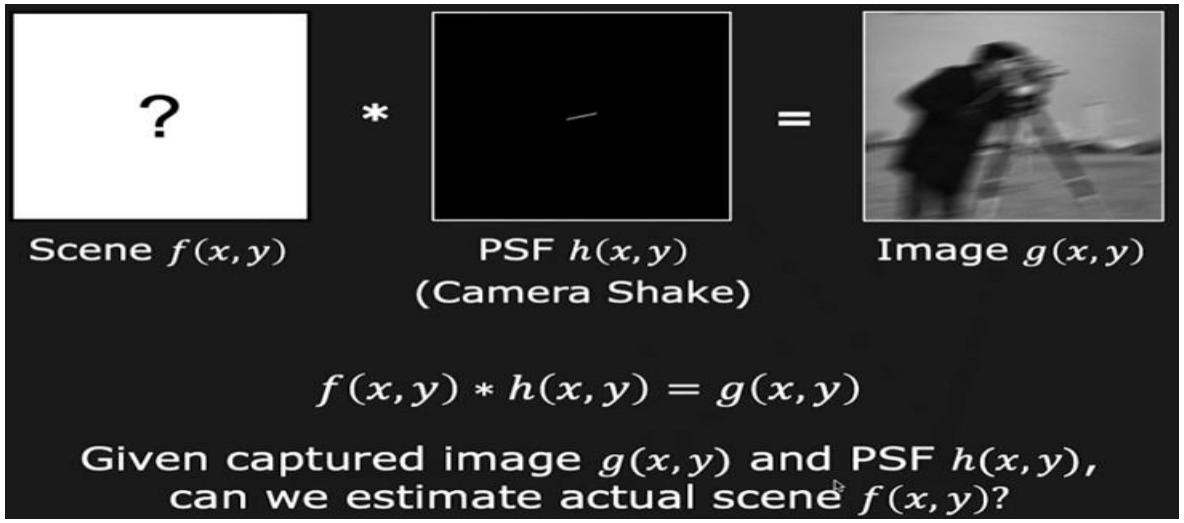


Deconvolution is the process to undo the convolution.



Take an example of the motion blur.

A original image convolved with point Spread function to get the blurred image.



How to find the PSF ?

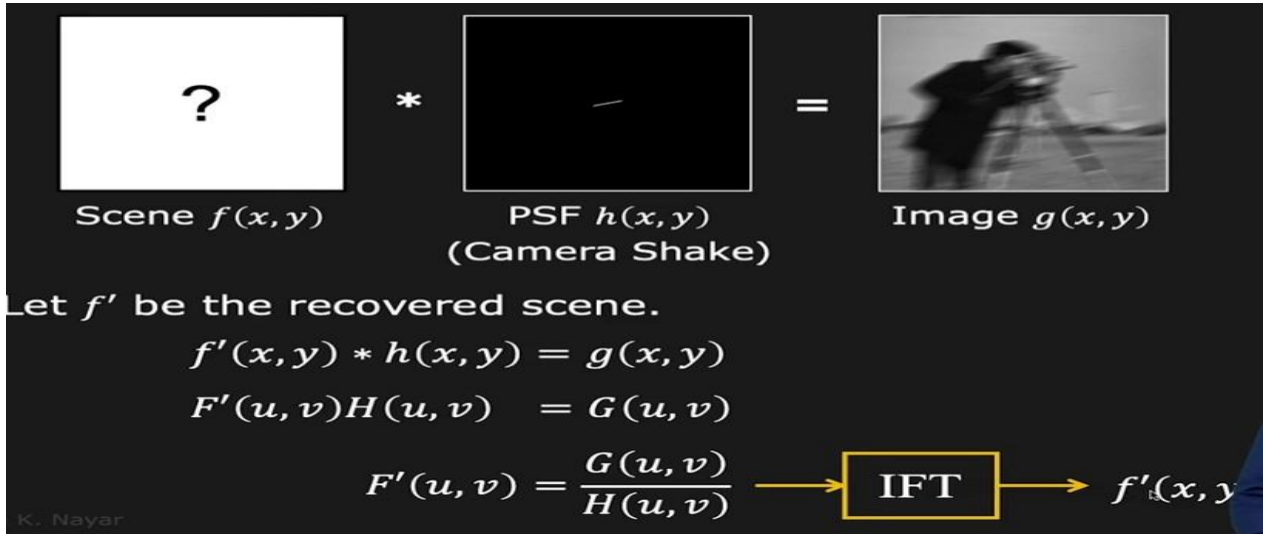
Smartphone camera contains some kind of measurement units -
Like -
Inertial measurement unit (IMU)
Accelerometer.
Gyroscope.
Use all of them to capture motion of the camera and with the help of all these parameters, generate the PSF.

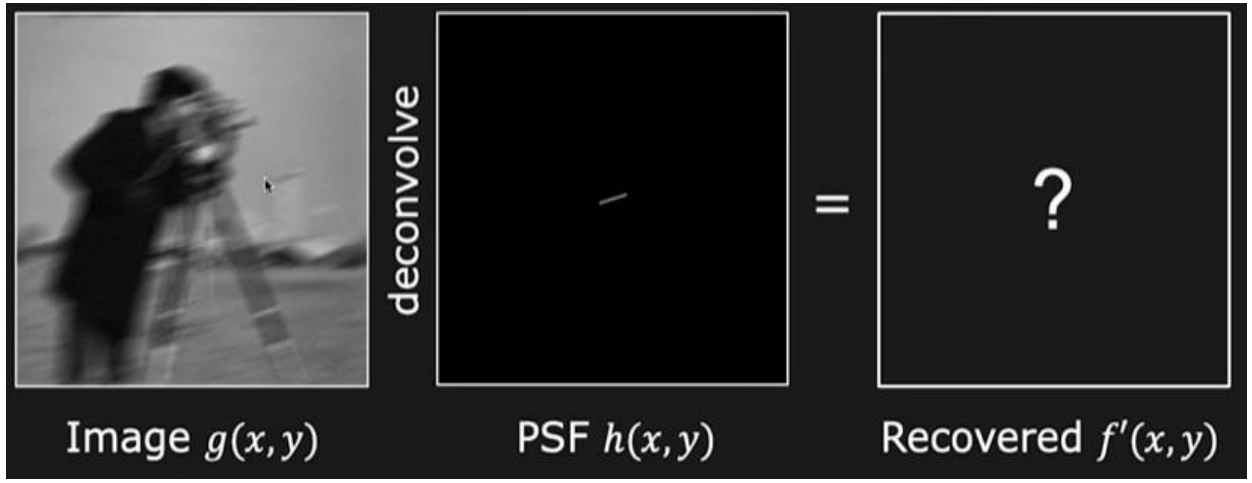
Deconvolution can be performed in better way in the frequency domain.

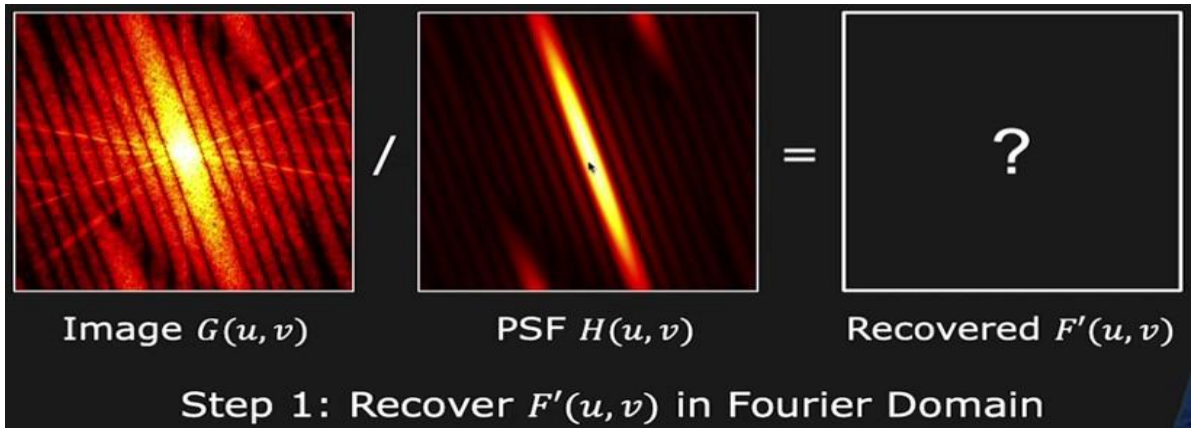
In the frequency domain, Fourier transform can help to recover the original image.

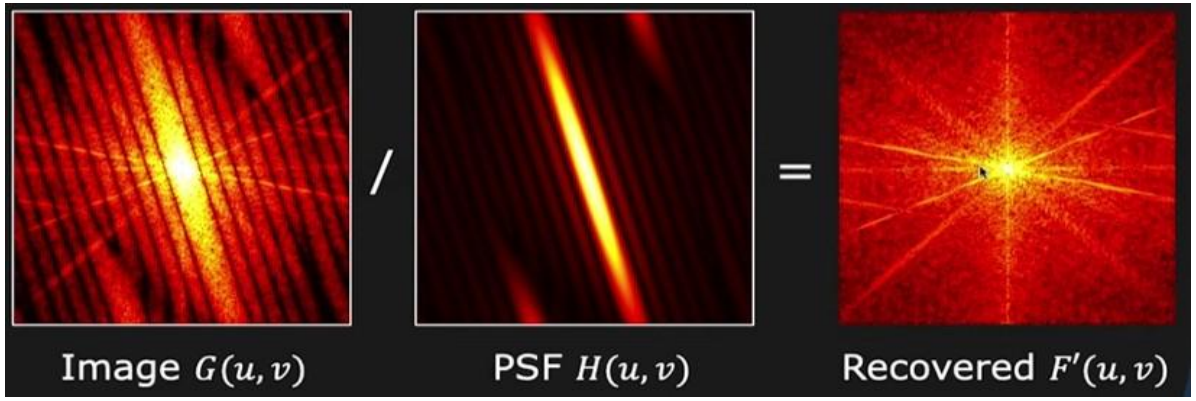
Deconvolution in the frequency domain

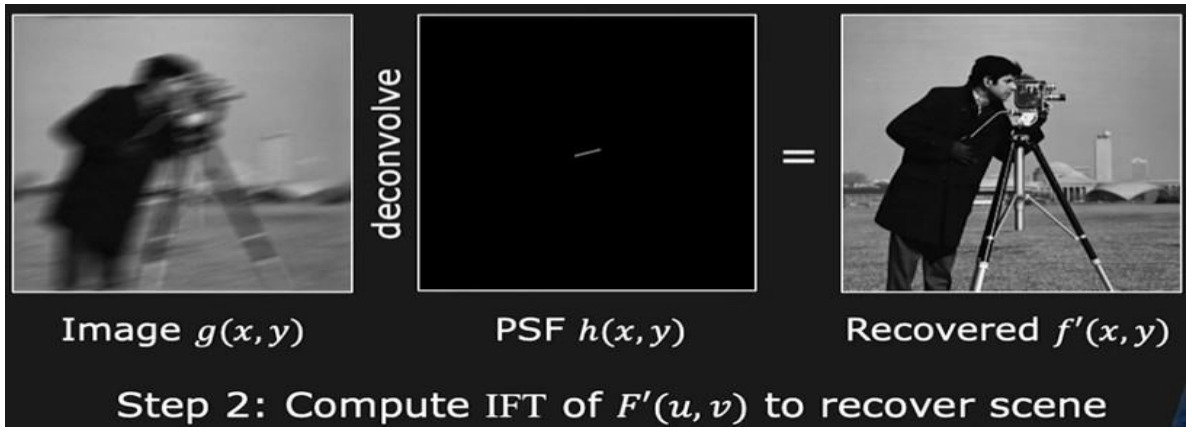
Here, you want to recover f'.
Take the FT and then find the F' and by performing IFT, you will get the original image i.e. (f').





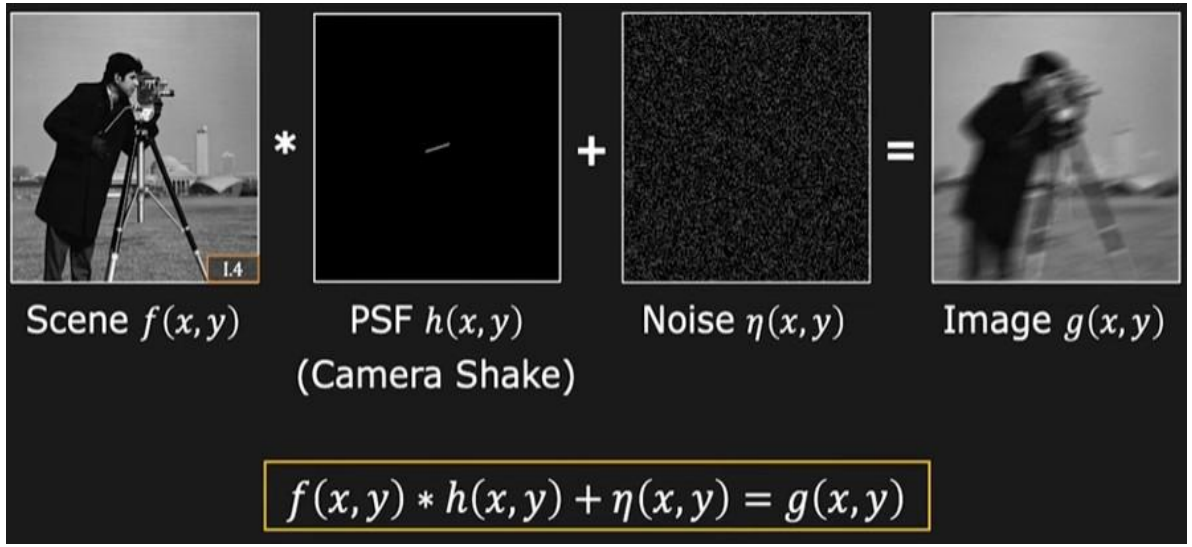




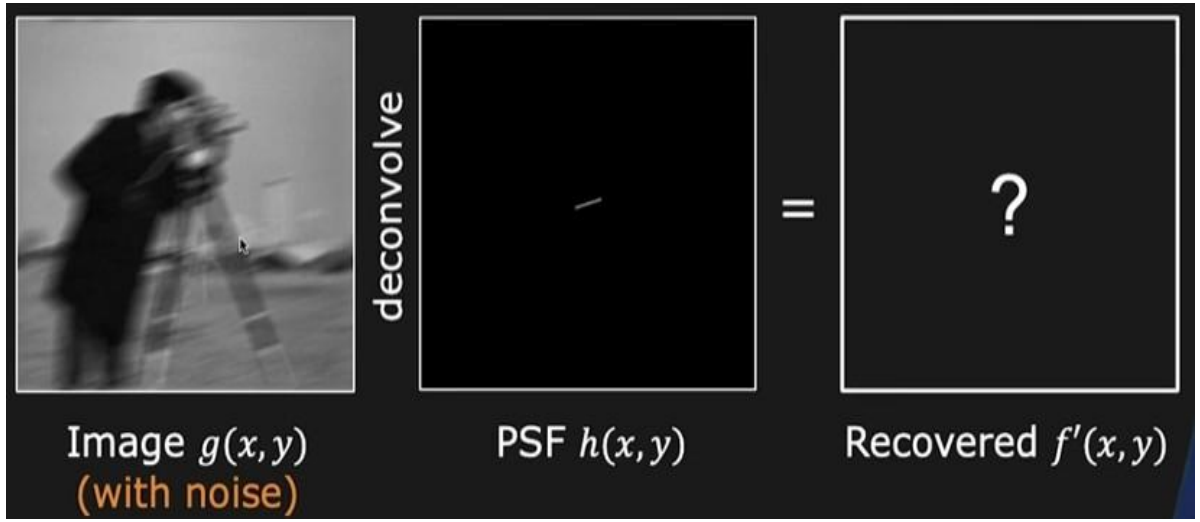


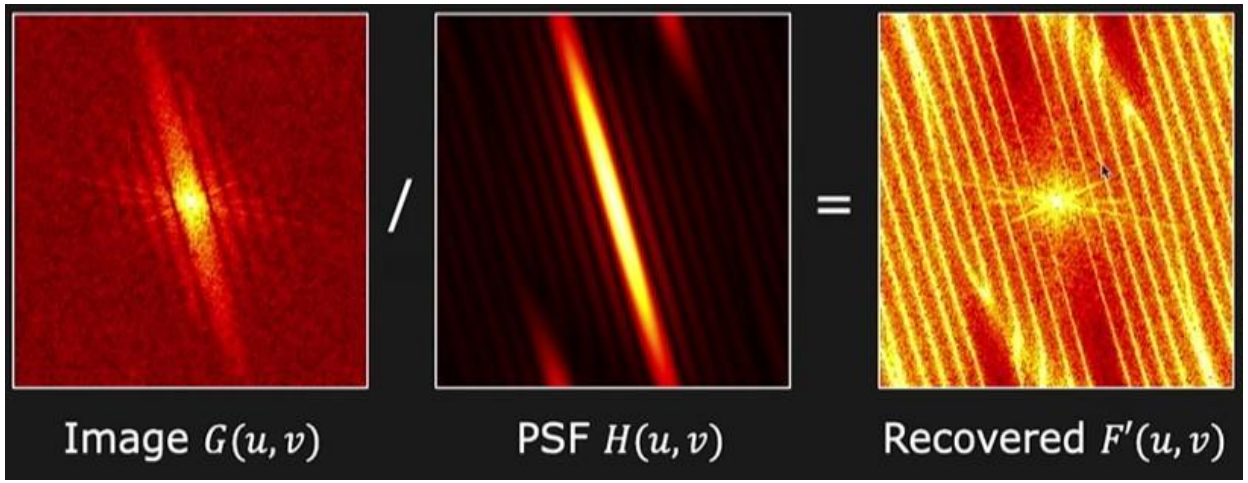
Deconvolution in the frequency domain

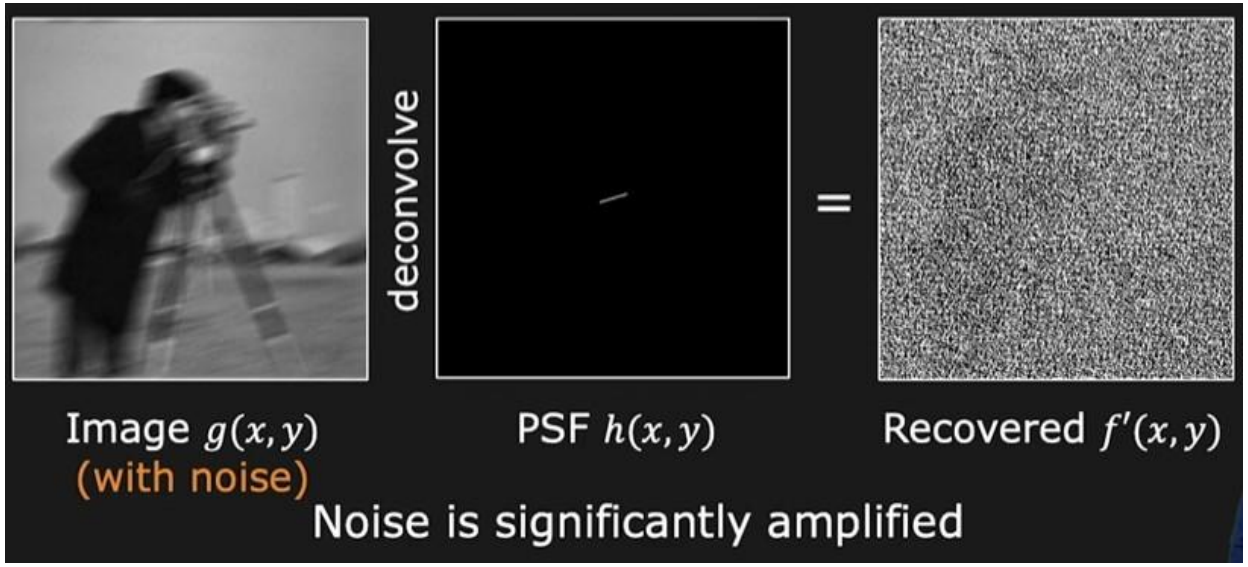
But we have some problem. The problem is related to the noise. After camera shake a noise has been added in the image.



Can we afford to avoid noise ?







Our original image is destroyed because of noise.

Deconvolution: Issues

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

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

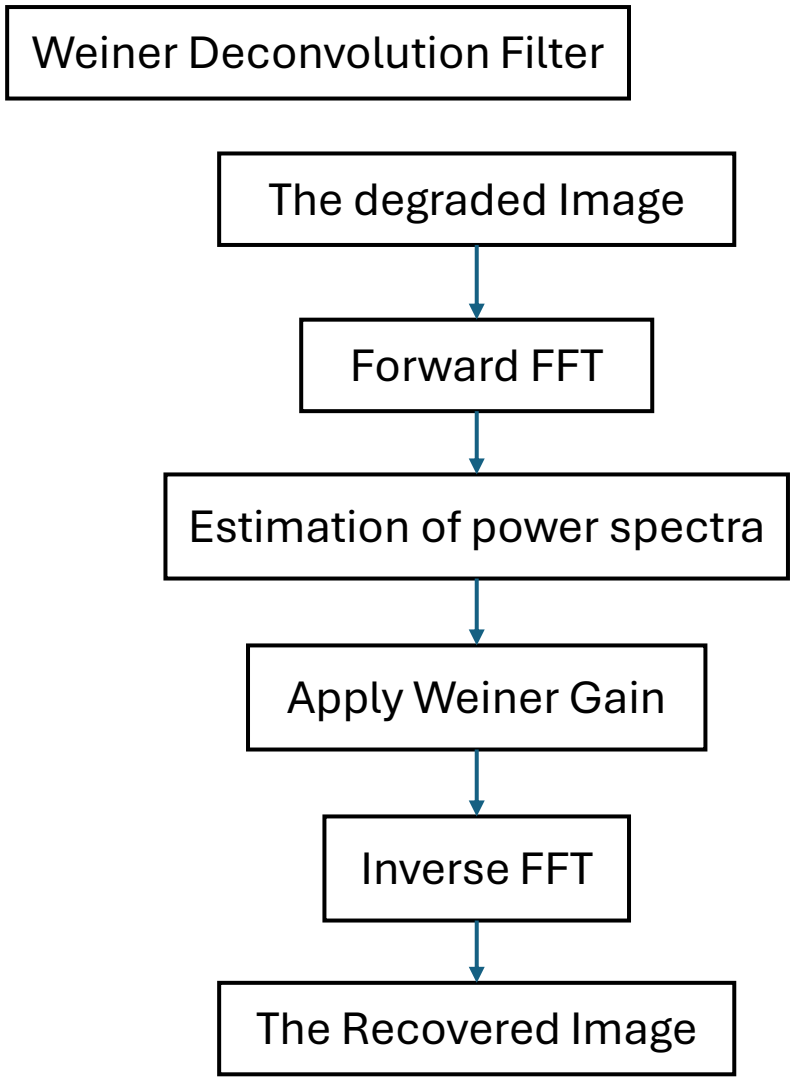
We need some kind of Noise Suppression.

Here, H(u,v) = Fourier transform of point spread function.
Basically H(u,v) is a low pass filter.

Let's say noise is present in the image. If it is high noise, then you will get more high frequencies in the image.

As we mentioned, Motion blur filter H(u.v) is a low pass filter so that output of it is approx. zero. Result in, F'(u,v) = infinite i.e. not recoverable.

Solution : During deconvolution process we need certain kind of noise suppression.



Weiner filter used to minimize the square distance between the Estimated and true image.