UNIT 2

Image Enhancement in the Frequency Domain

IMAGE ENHANCEMENT IN THE SPATIAL DOMAIN

- image enhancement is a process that improves the quality of an image for a specific application

- so depending upon the application the quality of the image will be improved in image enhancement
- highlight the interesting details or important details in an image
- It remove the noise from an image and also makes look more appealing

IMAGE ENHANCEMENT IN THE SPATIAL DOMAIN

3 methods can be used for image enhancement

spatial domain

Example: If we brighten an image, we just increase the pixel values.

frequency domain

instead of working with pixels, we **convert the image into frequencies** using the **Fourier Transform.** This helps us analyze the details of an image in terms of **low and high frequencies**.

- Some operations (like blurring, sharpening, or noise removal) work **better** in the frequency domain.Example:**Low frequencies** represent smooth areas.
- **High frequencies** represent sharp edges and details.
- If we remove high frequencies, we get a blurred image (used in noise reduction).

combination method

- the spatial domain deals with direct manipulation of pixel values
- the frequency domain deals with modifying the Fourier transform of the image
- In this frequency domain the image will be converted from spatial domain to frequency domain then it is processed using fouriertransform after processing the image is converted back to spatial domain

IMAGE ENHANCEMENT IN THE SPATIAL DOMAIN

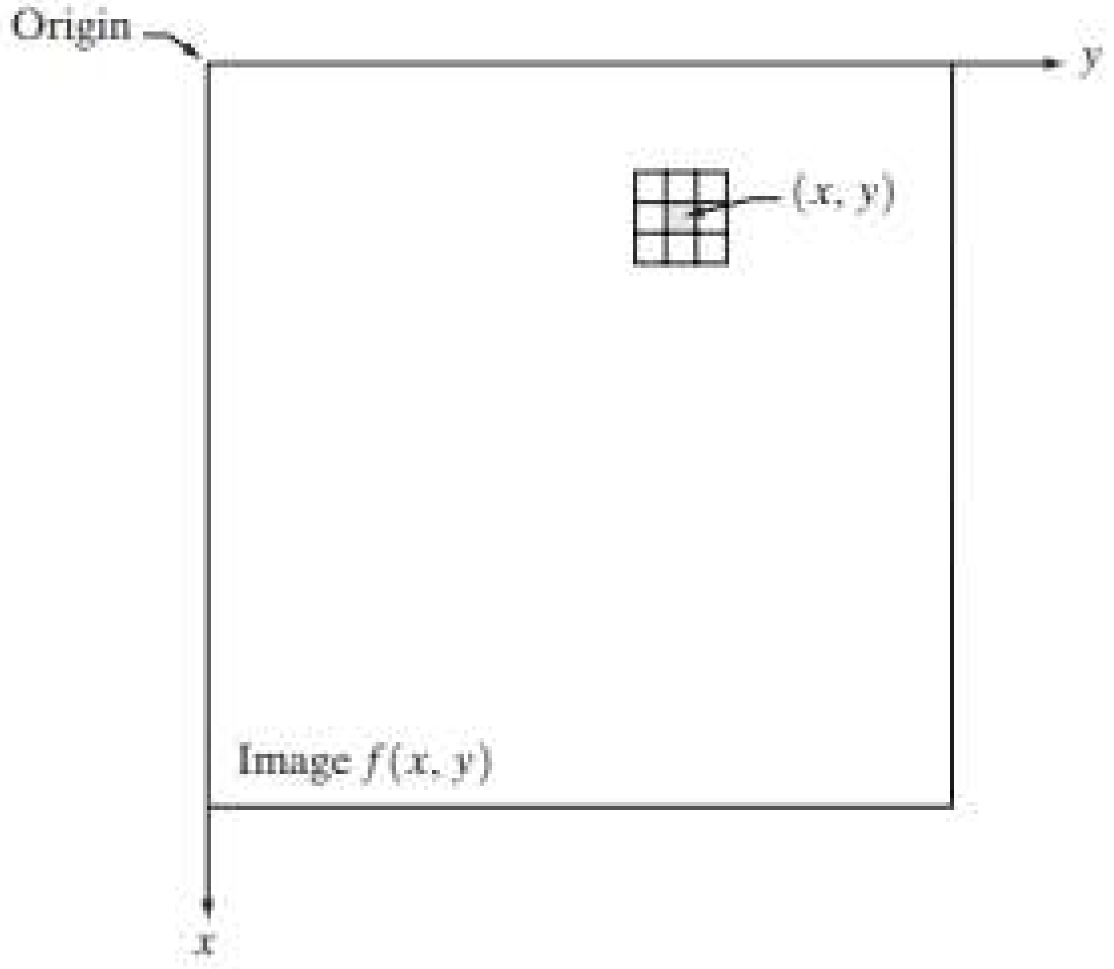
- next method is combination method which is combination of first and second method

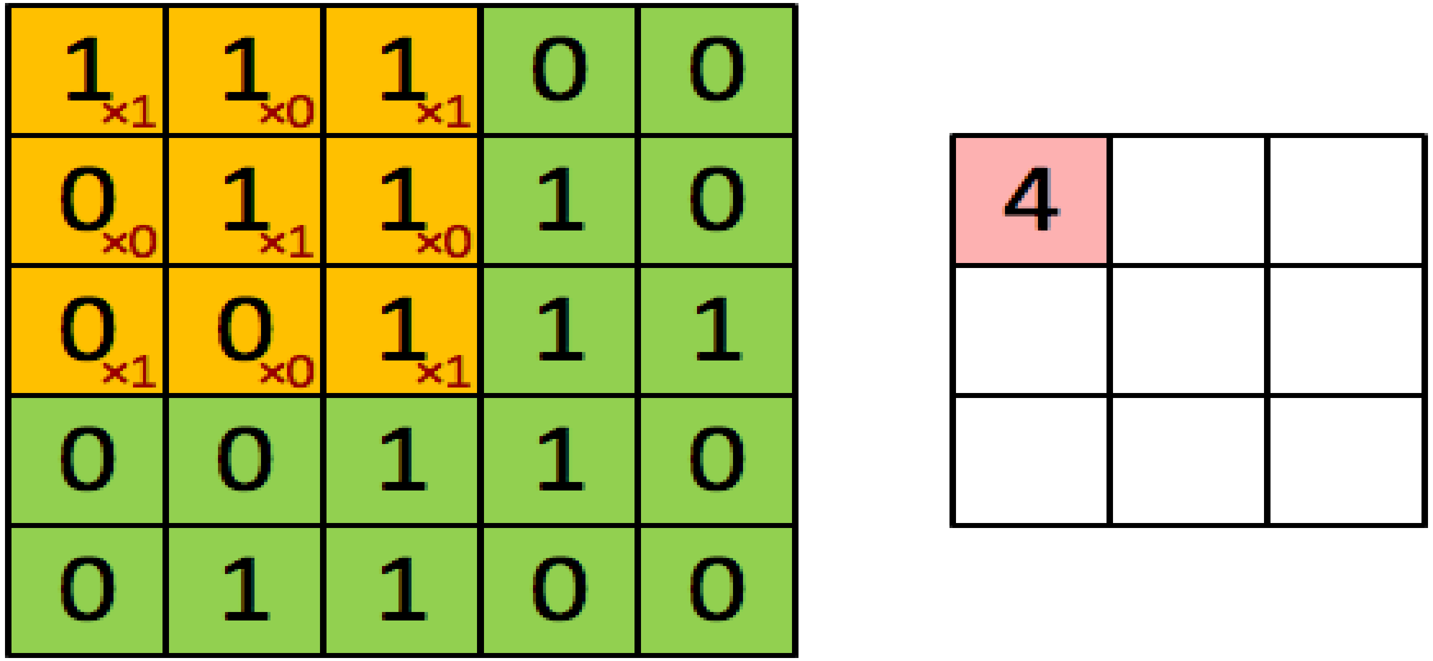
spatial domain

- The term spatial domain refers to the image plane itself
- the image with pixel values are nothing but spatial domain
- the image processing in this domain is direct manipulation of pixel values of an image
- there are 2 categories of spatial domain 1) intensity transformation in which the modification of intensity values of pixel will take place 2) spatial filtering here there will be direct manipulation of pixels using a mask the spatial domain technique operates directly on the pixels of an image

IMAGE ENHANCEMENT IN THE SPATIAL DOMAIN

- **Spatial Domain: Input -> Image Processing -> Output**
- **Frequency Domain: Frequency + Distribution -> Image Processing -> Inverse Transformation -> Output**
- g(x, y) = T(f(x, y))
- g (x, y) is the output image
- T is an operator
- f (x, y) is the input image





Image

Convolved Feature

Introduction to Image Enhancement using Spatial domain.

Image Enhancement - Process → Improves the Quality of an Image.

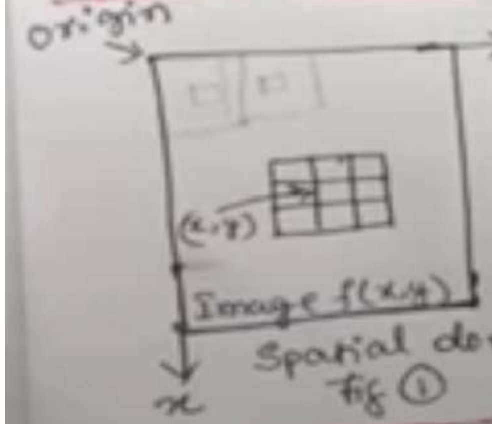
→ To highlight the Important details

→ To remove noise → Image → more appealing.

Methods:

1. Spatial domain :- Manipulation of pixel Values.
2. Frequency domain :- Modifying the F.T. of Image.
3. Combination method :- Combination of first two method.

Spatial domain: → Image plane itself



Spatial domain
fig 1

→ Direct manipulation of pixels

(a) Intensity Transformation.

(b) Spatial Filtering

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

T → operator

Simplest form → neighbourhood - size 1×1

T → grey level transformation [intensity (or) mapping].

$$\boxed{s = T(r)}$$

s → o/p Image pixel Value

r → i/p Image pixel Value

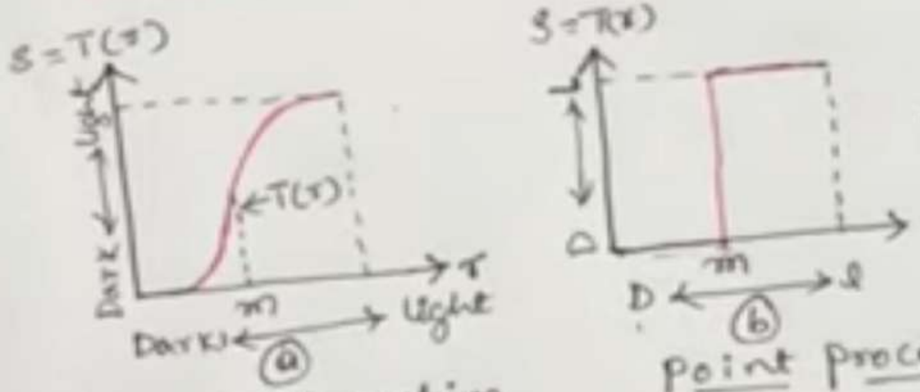


Fig 2

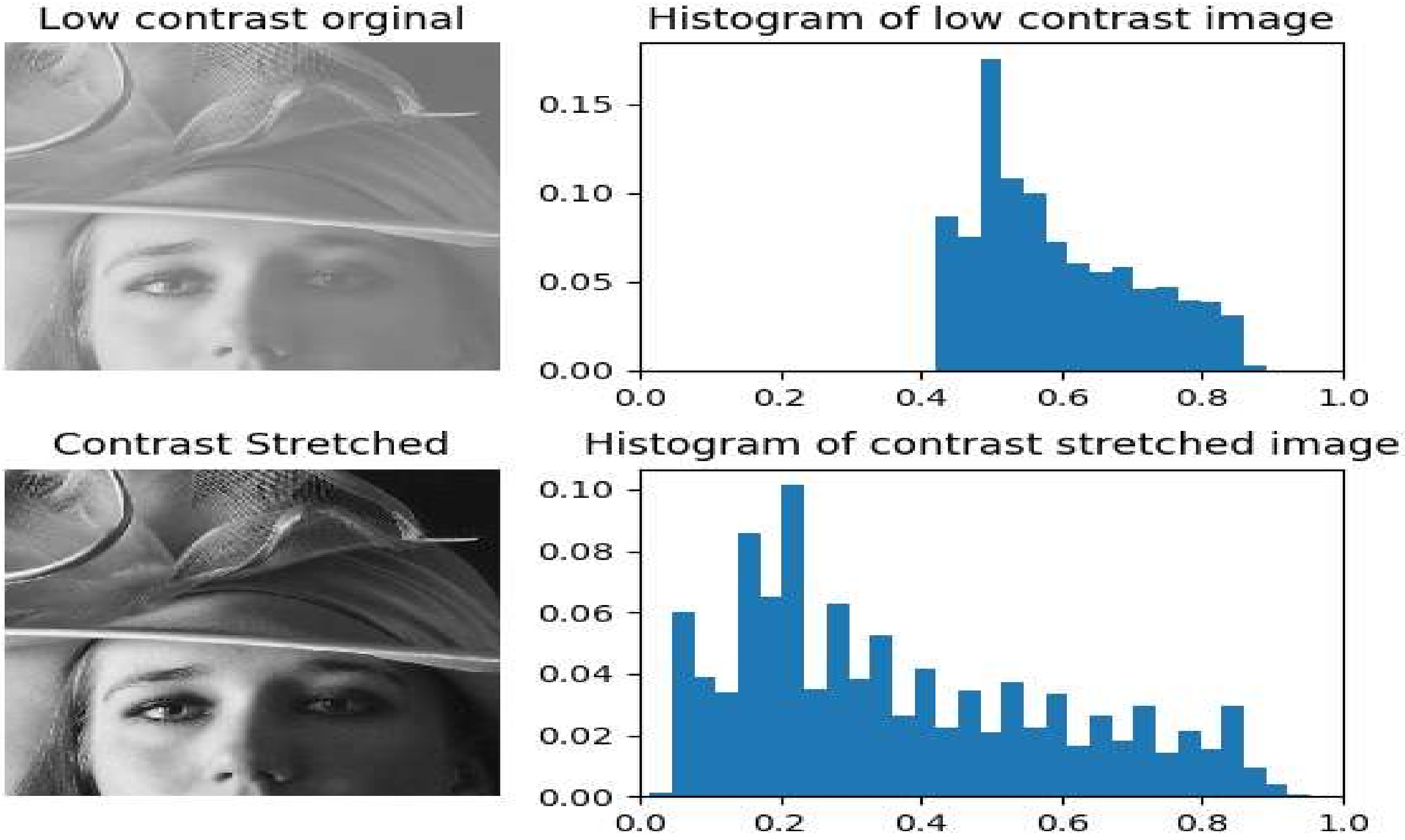
Contrast Streaching

Point processing

mask → Small [3×3] 2D array (or) window]

↳ [Filters, Kernels, Tamplets

→ Mask processing (or) Filtering.



Range of values

Gray level transformation.

$\boxed{S = T(r)}$

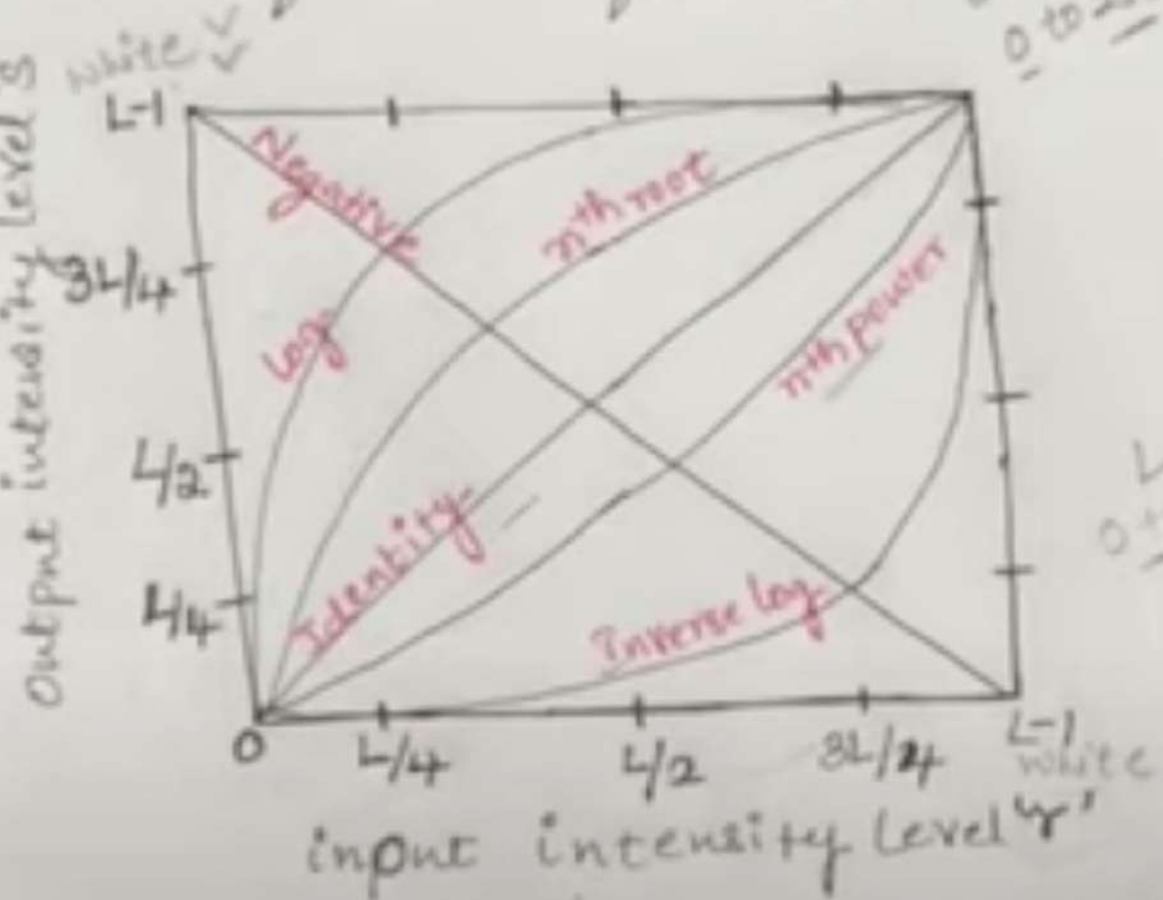
$r \rightarrow$ Value of pixel before processing

$s \rightarrow$ value of pixel after processing

$T \rightarrow$ Transformation function.

$2^8 = 256$

0 to 255



Some Basic transformation funs

L

0 to L-1

Intensity level $\rightarrow [0, L-1]$

$\boxed{S = L-1-r}$

Ex:- $r=0$ ✓ $\boxed{S = L-1}$

$r = L-1$ ✓ $\boxed{S=0}$ ✓

(b) Log Transformation

$\boxed{S = C\log(1+r)}$

$c \rightarrow$ Constant

$r \geq 0$

(c) Power law [gamma correction]

$\boxed{S = Cr^{\gamma}}$

$\gamma > 1 \rightarrow n^{th}$ power

$\gamma < 1 \rightarrow n^{th}$ root

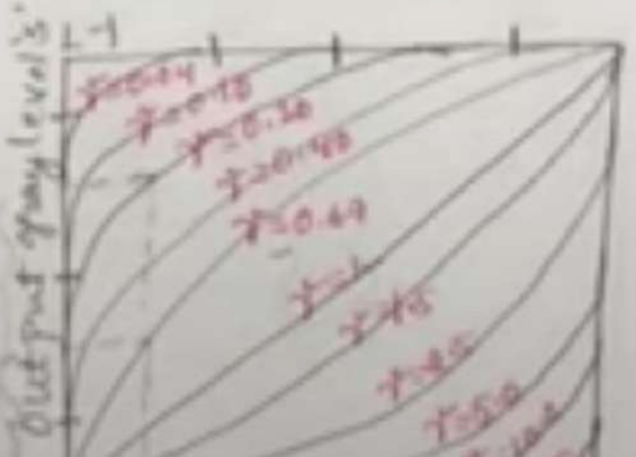


IMAGE ENHANCEMENT

- Enhancing an image provides better contrast and a more detailed image as compare to non enhanced image.

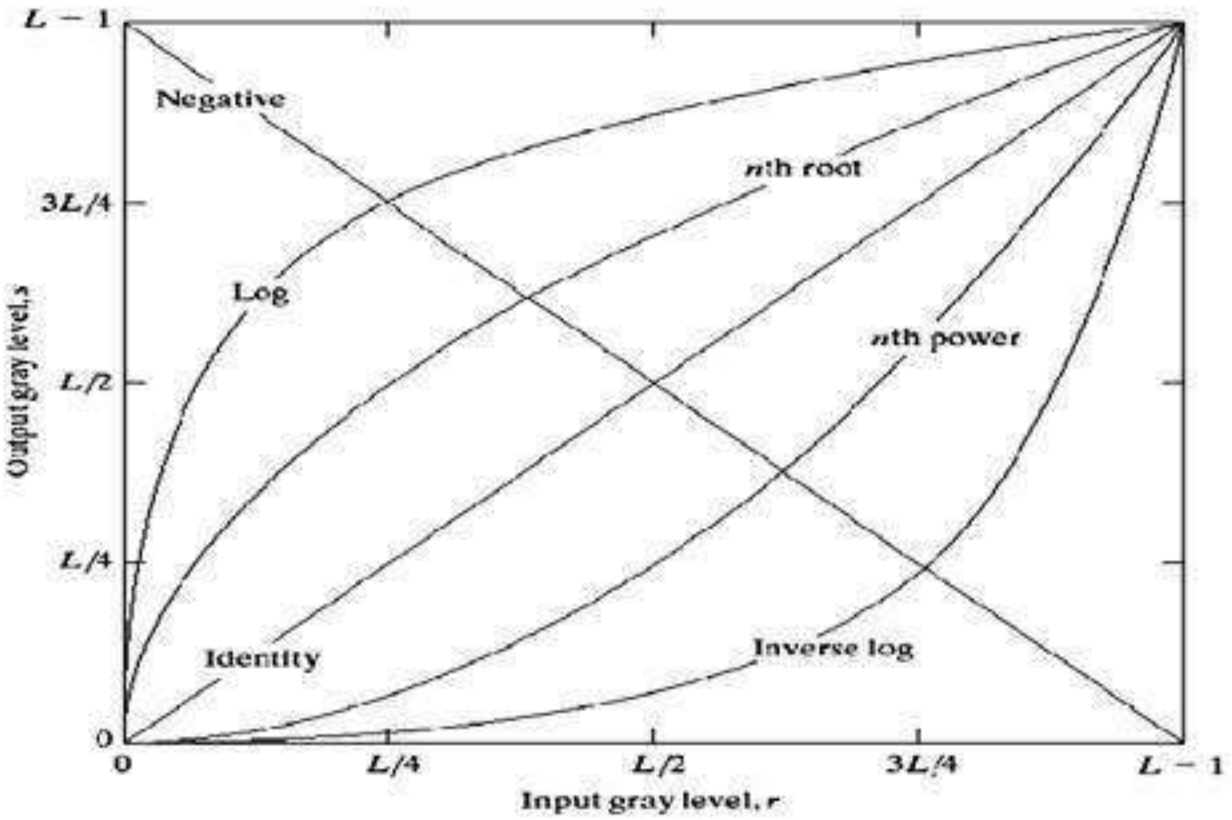
The transformation function has been given below

s = T (r)

- Gray level transformation

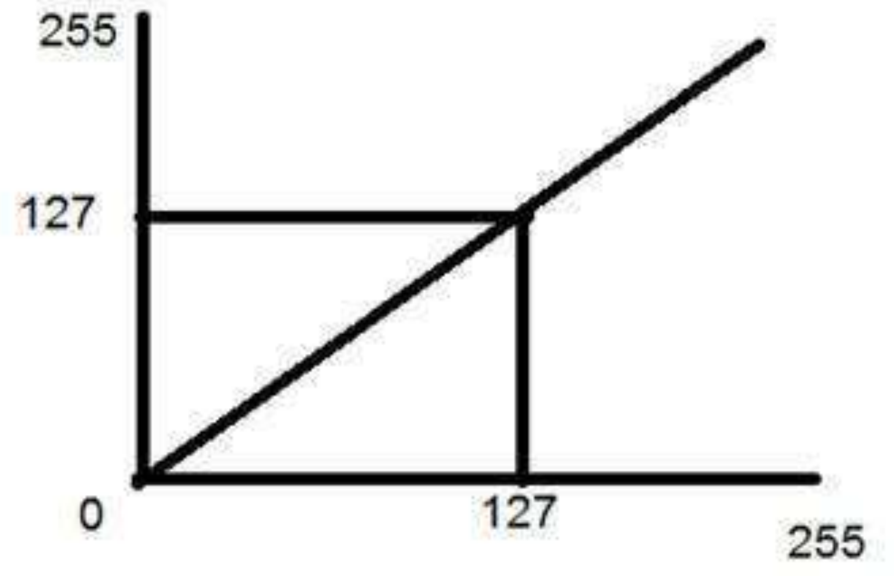
There are three basic gray level transformation.

- Linear
- Logarithmic
- Power – law


L − 1
3L/4
L/2
L/4
0
Output gray level, s
Negative
Log
nth root
Identity
nth power
Inverse log
0
L/4
L/2
3L/4
L − 1
Input gray level, r

LINEAR TRANSFORMATION

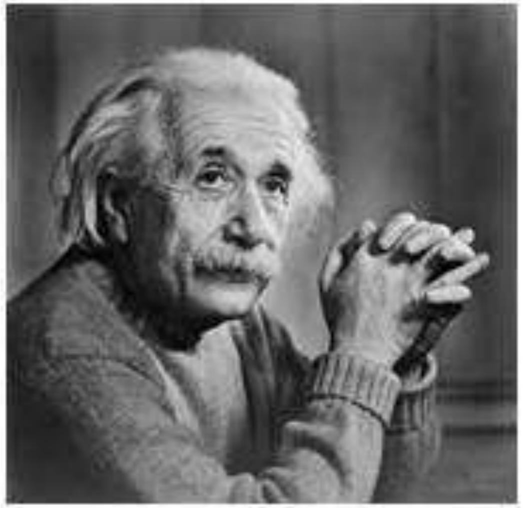
- Linear transformation includes simple identity and negative transformation.
- Identity transition is shown by a straight line. In this transition, each value of the input image is directly mapped to each other value of output image. That results in the same input image and output image. And hence is called identity transformation. It has been shown below:

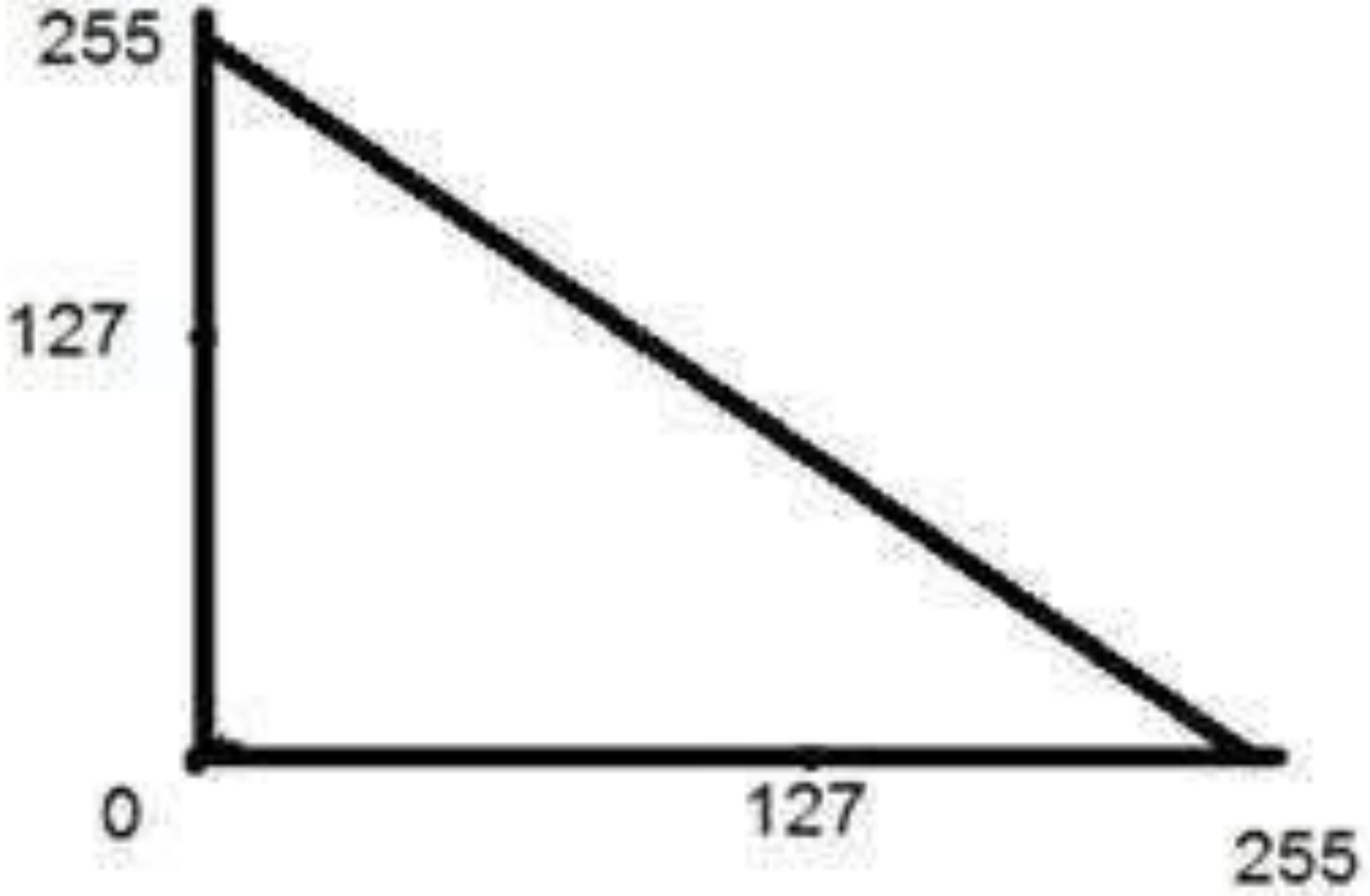


NEGATIVE TRANSFORMATION

- The second linear transformation is negative transformation, which is invert of identity transformation. In negative transformation, each value of the input image is subtracted from the L-1 and mapped onto the output image.

1. In this case the following transition has been done.
2. s = (L – 1) – r

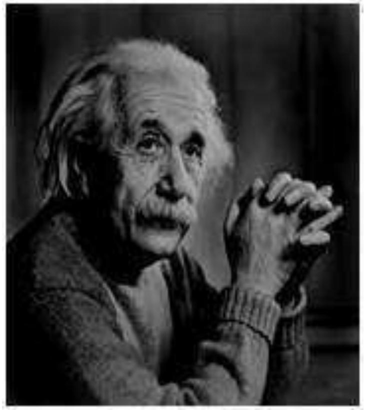



255
127
0
127
255

LOG TRANSFORMATION

- This formula can define the log transformations
- It **brightens dark areas** more than bright areas. It’s useful when the image is **very dark** (like photos taken at night or in low light).
- $s = c \log(r + 1)$.

1. **Log transformation** stretches the pixel values in the dark region (low intensities) to make them more visible.Bright areas won’t change much, but dark areas become clearer.



INVERSE LOG TRANSFORMATION

1. It does the **opposite** — it **darkens bright areas** more than dark areas. It's useful when the image is **too bright** (like overexposed photos).
2. Imagine you took a photo on a sunny day, and everything looks too bright (overexposed).

- **Inverse log transformation** compresses the bright areas to make them darker and more natural.

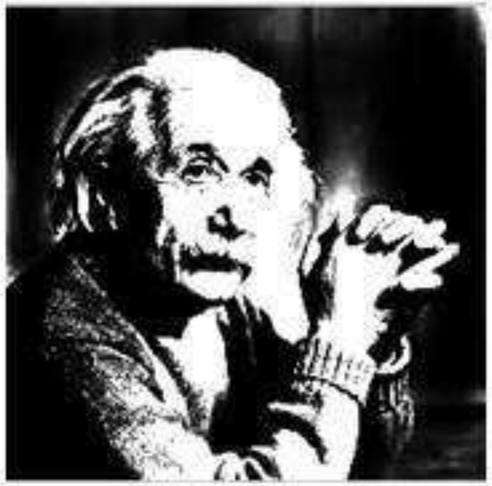
POWER – LAW TRANSFORMATIONS

- There are further two transformation is power law transformations, that include nth power and nth root transformation. The expression can give these transformations :
- It is used to **control brightness**.
- It works using a power (exponent) called **gamma (γ)**.
- **If gamma < 1** → **Brightens the image.**
- **If gamma > 1** → **Darkens the image.**

s=cr^γ

- This symbol γ is called gamma, due to which this transformation is also known as gamma transformation.
- Variation in the value of γ varies the enhancement of the images.

1. Gamma=8



gamma=6

SUMMARY:

Linear Intensity Transform

a) Identity Transformation

Definition: Output intensity = Input intensity.

- It does **nothing** — image stays the same.
- $s=r$
- **Real-time Example:** Just viewing the image without any enhancement.
- **Advantage:** No loss of information.
- **Disadvantage:** No enhancement is applied — so low contrast or poor images remain poor.

CONTINUED

Negative Transformation

Definition: Reverses intensity — bright areas become dark, dark becomes bright.

$s = L - 1 - r$ (where L is max intensity like 255 for 8-bit images)

- **Real-time Example:** Used in **X-rays** and **medical images** to highlight tissues or bones.
- **Advantage:** Useful for **medical imaging** where bright and dark regions need to be swapped for better visibility.
- **Disadvantage:** Not suitable for natural images (photos) — looks unnatural.

CONTINUED

Logarithmic Intensity Transformation

This method **brightens the dark parts of the image** and **reduces brightness in very bright parts**.
It **compresses bright pixels** (so they don't appear too bright) and **expands dark pixels** (so they become more visible).

The formula is:

- $s = c\log(1+r)$
- r = input pixel intensity (original image)
- s = output pixel intensity (transformed image)
- c= constant that controls how strong the effect is

CONTINUED

Real-time Example

1. Think about **satellite images**. Some parts of the image (like areas in sunlight) are **very bright**, while others (like shadowed areas) are **very dark**.
   Log transformation helps make both areas visible — it **brings out details from shadows** without making the bright areas too bright.

Advantage

- Makes **dark regions clearer** without losing too much detail in bright areas.
- Very useful for images with **big brightness differences,** like satellite images or medical scans.

Disadvantage

- You have to choose the **value of c** carefully.
  If you choose it wrong, the image may become too **dull** or too **overexposed**.
- It's not "one setting fits all" — you need to **adjust for each image**.

- For **dark images** → use larger c to brighten.
- For **bright images** → use smaller c to avoid over-brightening.

INVERSE LOG

It does the **opposite of Log Transformation**.

- **Log Transformation:** Compresses bright pixels and expands dark pixels (helps with dark areas).
- **Inverse Log Transformation:** Expands bright pixels and compresses dark pixels (helps with bright areas).

Formula:

- $s = e^r - 1$ r = input pixel intensity (normalized to 0-1)
- s = output pixel intensity (normalized to 0-1)

If working with pixel values in range 0-255, you adjust formula:

$s = c \cdot (e^r - 1)$where c is a scaling constant.

nth Root ($\gamma < 1$)
This **expands brightness** (makes dark areas **brighter**).

- It **spreads out low intensities** (dark pixels) and compresses high intensities (bright pixels slightly).
- Works like **a gentle brightening filter**.
- **In short:**
  Root expands brightness in dark areas (good for dark images).

nth Power ($\gamma > 1$)
This **compresses brightness** (makes the whole image darker).

- It **spreads out bright pixels** and compresses dark pixels.
- Works like **a gentle darkening filter**.

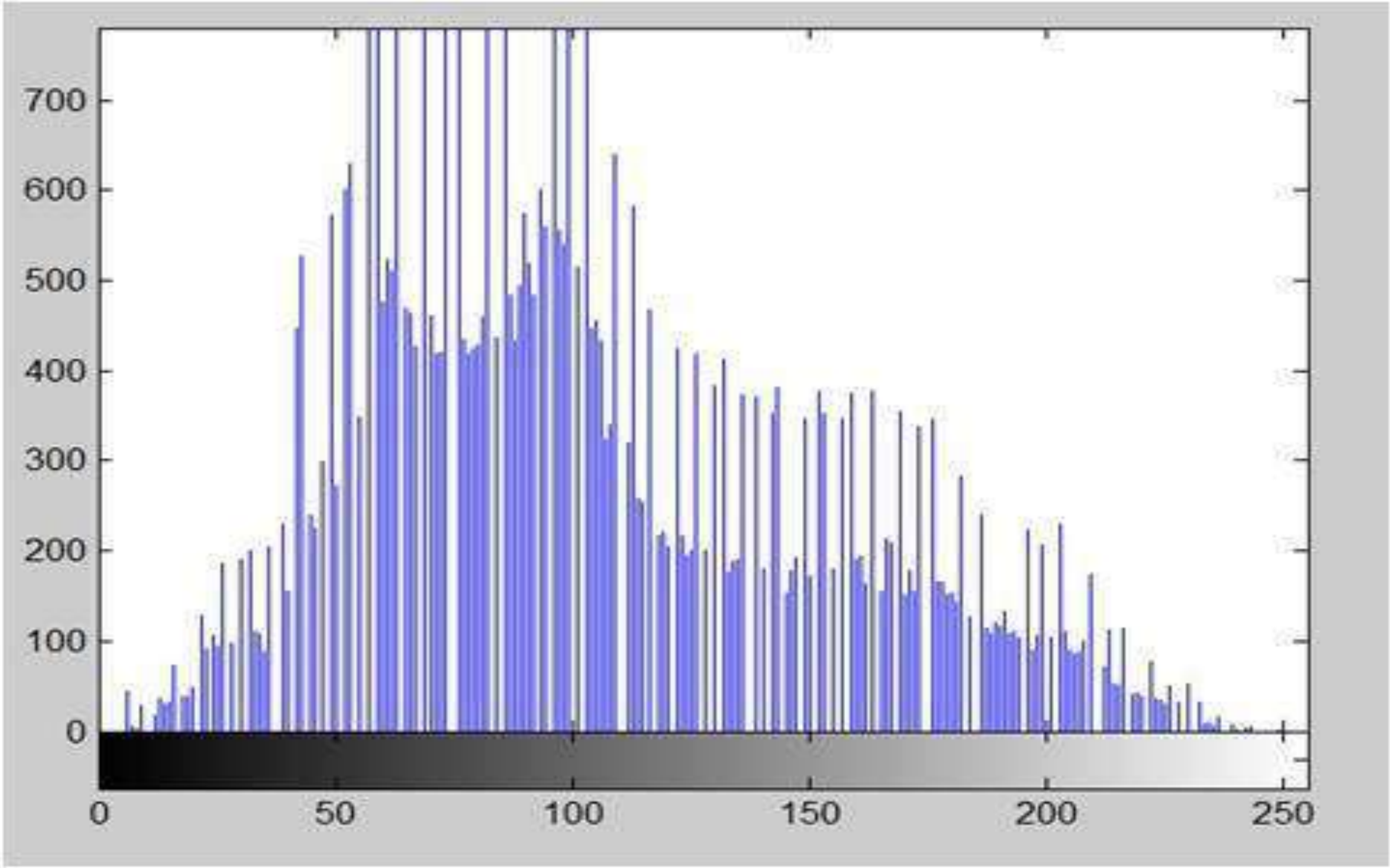
In short:
Power compresses brightness (good for bright images).

HISTOGRAMS

- A histogram is a graph.
- A Histogram has two axis the x axis and the y axis.
- The x axis contains event whose frequency you have to count.
- The y axis contains frequency.

Histogram of an image

- Histogram of an image shows frequency of pixels intensity values
- In an image histogram, the x axis shows the gray level intensities and the y axis shows the frequency of these intensities.



700
600
500
400
300
200
100
0
0
50
100
150
200
250

HISTOGRAMS

1. A **histogram** of an image shows **how many pixels** have each brightness (gray level) value — from 0 (black) to 255 (white).
2. **Histogram Processing** means using the **histogram to improve the image** — making it clearer, brighter, or better balanced.
3. **Types of Histogram Processing**
4. **Histogram Equalization**

- Spreads out the brightness levels **evenly**.
- Useful when the image is **too dark** or **too bright**.
- It improves **contrast** (makes hidden details visible).

Real Example:

- Old black-and-white photo looks faded.
- After **Histogram Equalization**, the photo has better contrast, and details are easier to see.

2. Histogram Matching (Histogram Specification)

- Change one image’s brightness distribution to match another image’s histogram.
- Useful to make **two images look similar** (e.g., medical images from different machines).

HISTOGRAM EQUALIZATION → Image Enhancement.

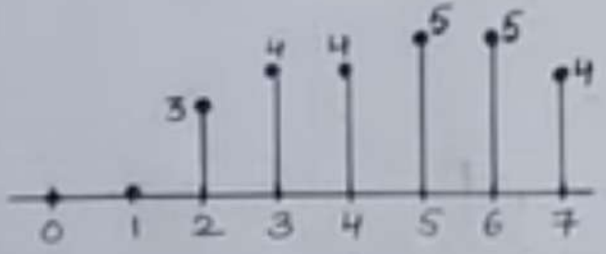
→ Graphical Representation → Data.

→ Image Processing → Data related to the Digital Image.

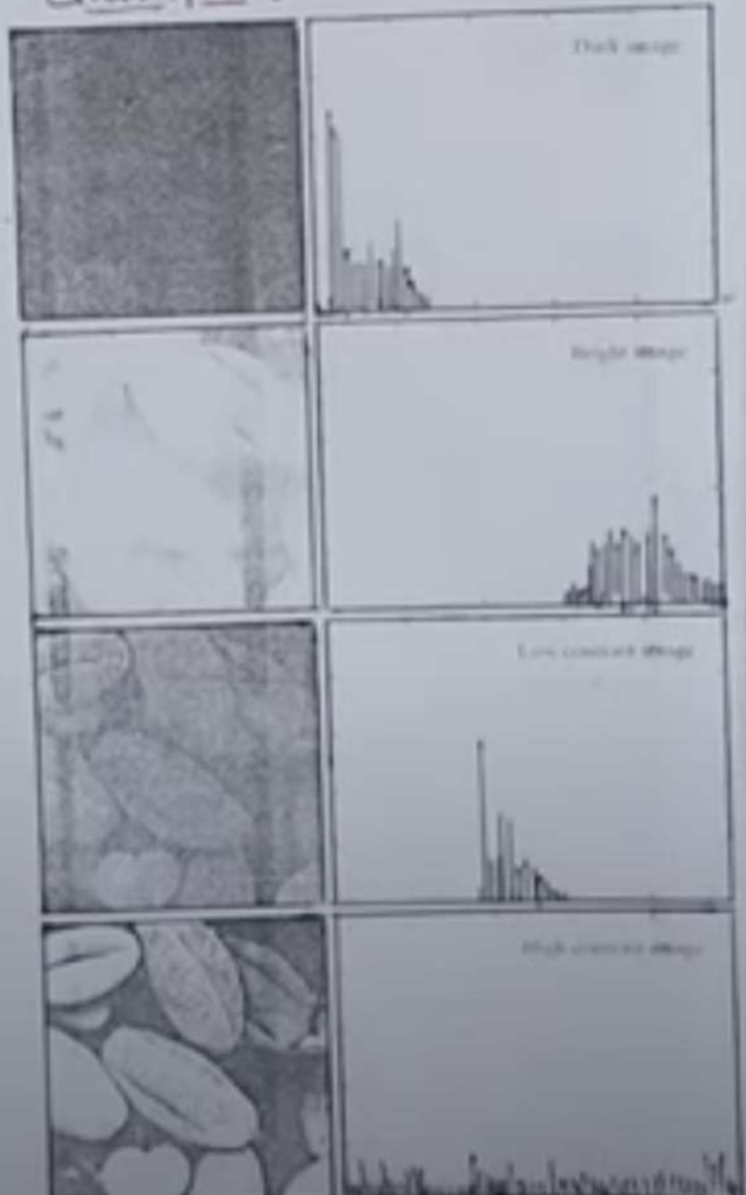
→ Representation → Frequency of occurance of Various gray levels.

6	6	7	7	6
5	2	2	3	4
3	3	4	4	5
5	7	3	6	2
7	6	5	5	4

0 → 0
1 → 0
2 → 3
3 → 4
4 → 4
5 → 5
6 → 5
7 → 4



Example:

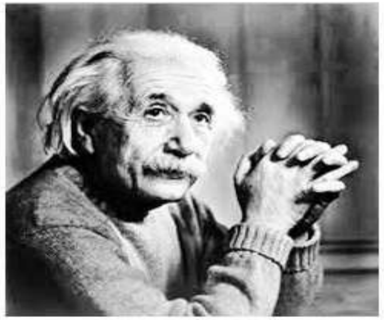


→ Used → Manipulating Contrast & Brightness.

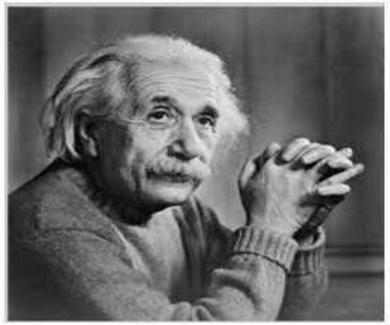
→ Quality → Normalizing → Histogram → Flat profile

HISTOGRAM EQUALIZATION

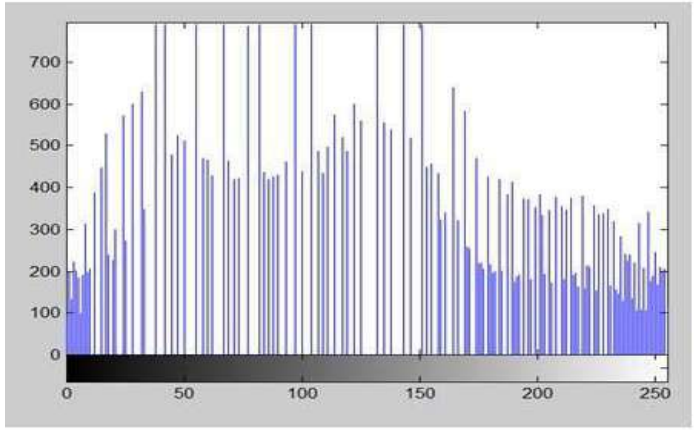
New Image



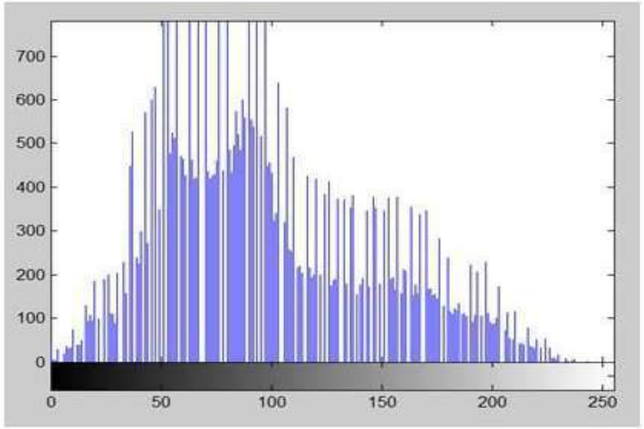
Old image



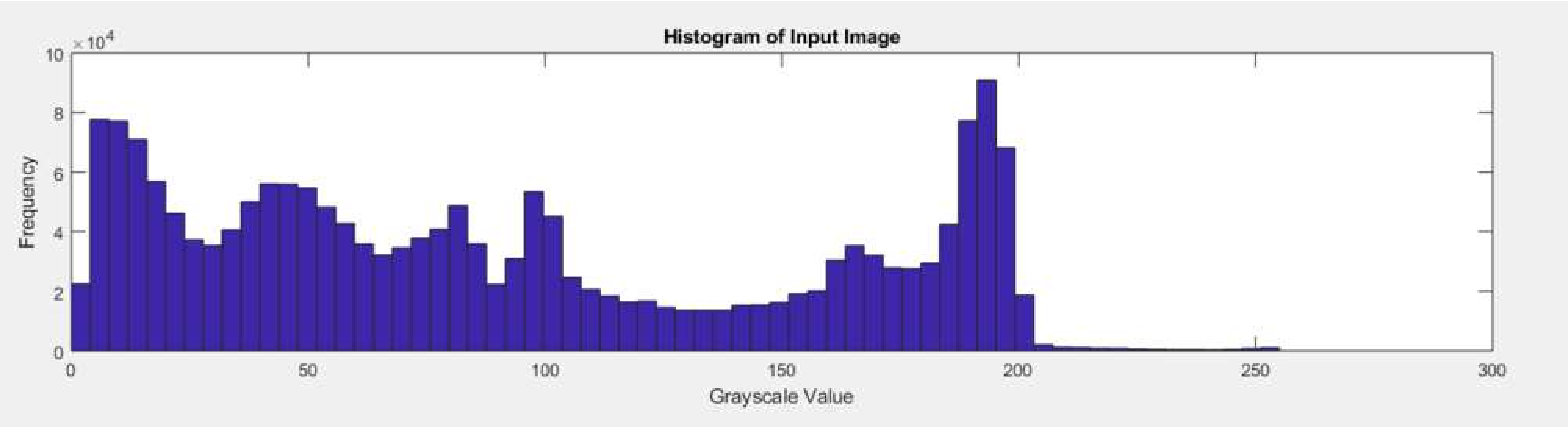
New Histogram



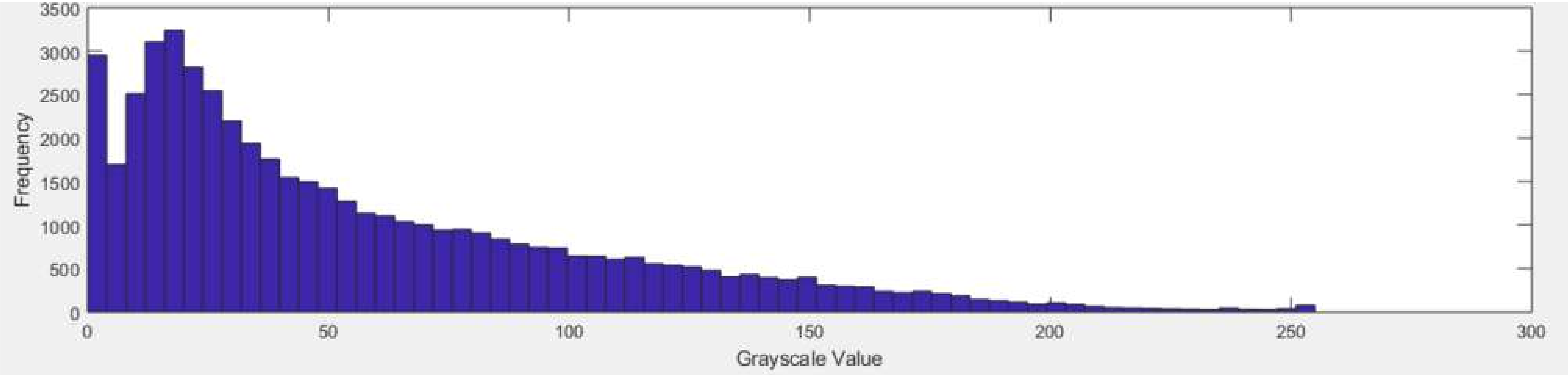
Old Histogram



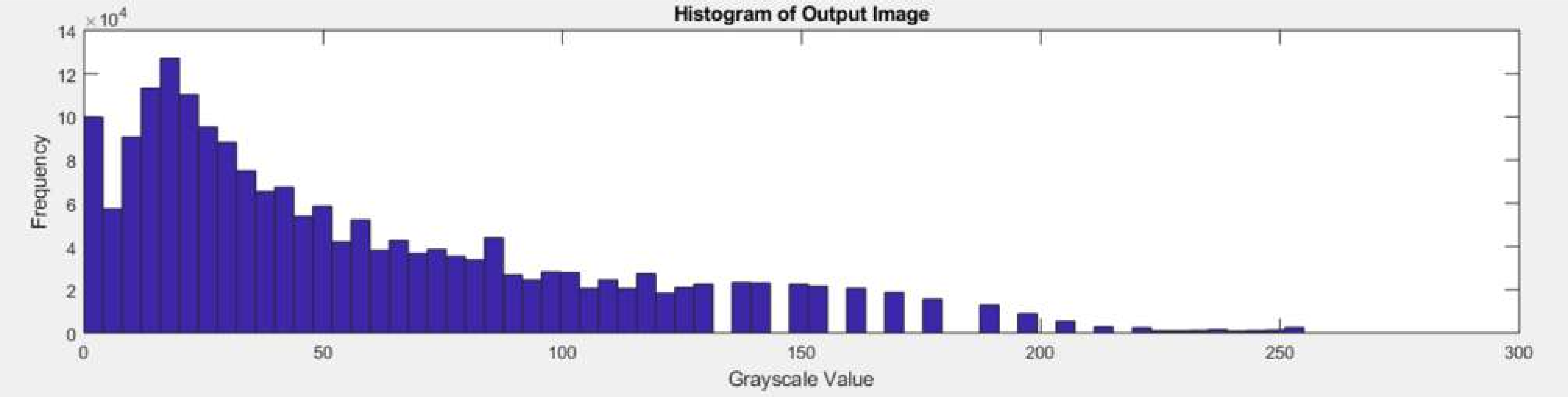

ISSI
DE1-SoC
terasic
Download DE1-SoC CD from
http://cd-de1-soc.terasic.com



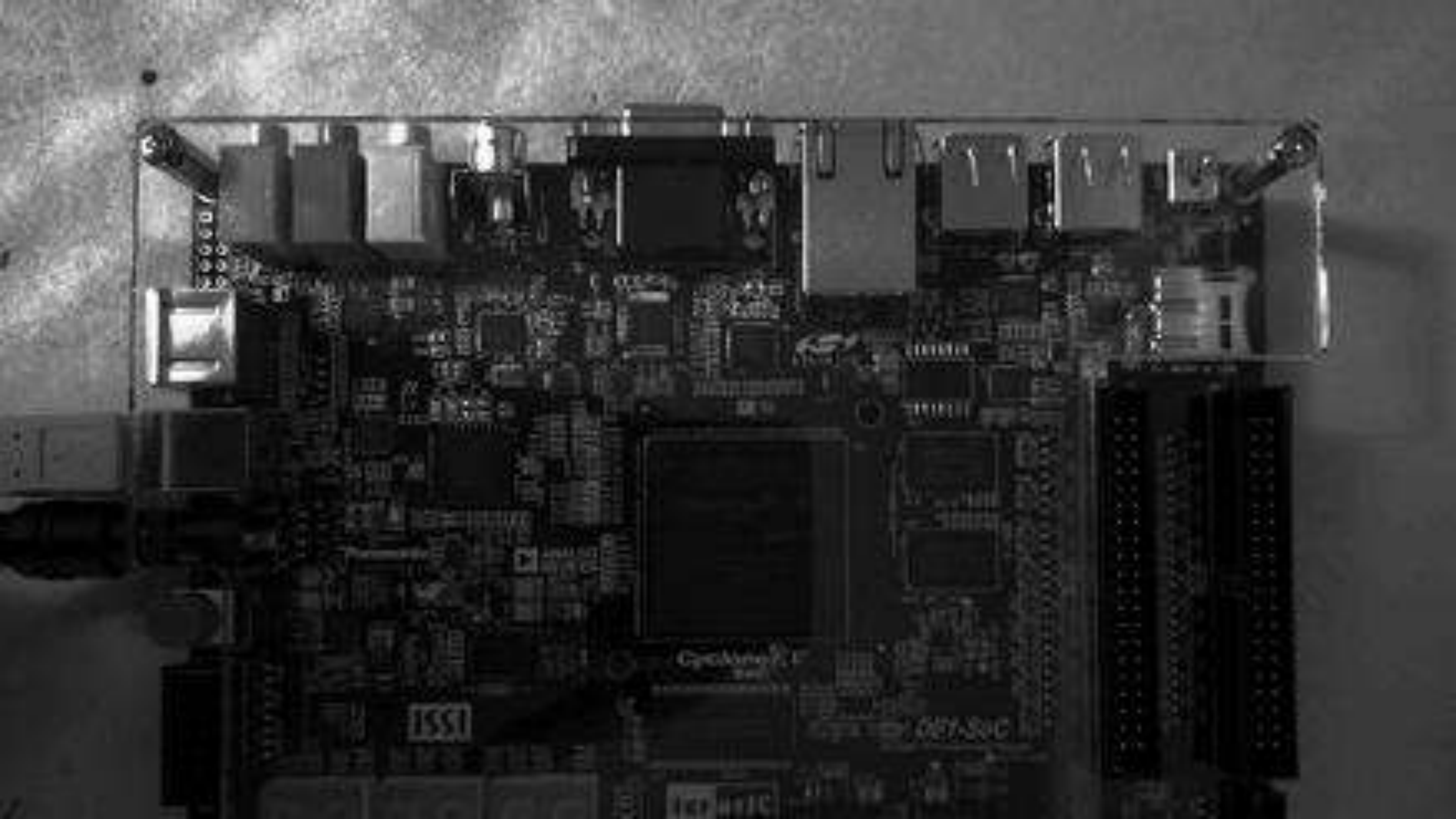
Histogram of Input Image
×10^4
Frequency
Grayscale Value
0
50
100
150
200
250
300
0
2
4
6
8
10



3500
3000
2500
2000
1500
1000
500
0
Frequency
0
50
100
150
200
250
300
Grayscale Value


Histogram of Output Image
$\times 10^4$
14
12
10
8
6
4
2
0
Frequency
0
50
100
150
200
250
300
Grayscale Value

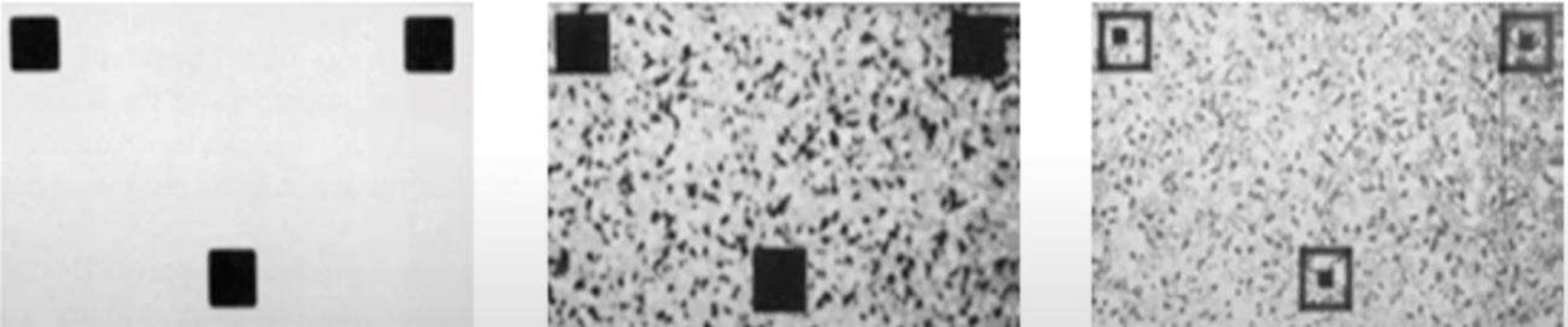
1. The histogram processing methods discussed in the previous two sections are *global*,
2. . Although this global approach is suitable for overall enhancement, there are cases in which it is necessary to en- hance details over small areas in an image


ISSI

Local Enhancement

- The histogram processing techniques are easily adaptable to local enhancement.
- The procedure is to define a square or rectangular neighborhood and move the center of this area from pixel to pixel.
- At each location, the histogram of the points in the neighborhood is computed and either a histogram equalization or histogram specification transformation function is obtained.
- This function is finally used to map the gray level of the pixel centered in the neighborhood.
- The center of the neighborhood region is then moved to an adjacent pixel location and the procedure is repeated.
- Since only one new row or column of the neighborhood changes during a pixel-to-pixel translation of the region, updating the histogram obtained in the previous location with the new data introduced at each motion step is possible.
- This approach has obvious advantages over repeatedly computing the histogram over all pixels in the neighborhood region each time the region is moved one pixel location.

- Another approach used some times to reduce computation is to utilize non-overlapping regions, but this method usually produces an undesirable checkerboard effect.
- Fig. 6(a) shows an image that has been slightly blurred to reduce its noise content.
- Fig. 6(b) shows the result of global histogram equalization.
- As is often the case when this technique is applied to smooth, noisy areas, Fig. 6(c) shows considerable enhancement of the noise, with a slight increase in contrast.



Basics of Spatial Filtering

- What is a mask?

- Concept of masking is also known as spatial filtering. Masking is also known as filtering. In this concept we just deal with the filtering operation that is performed directly on the image.

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

- What is filtering

The process of filtering is also known as convolving a mask with an image.

- How it is done

The general process of filtering and applying masks is consists of moving the filter mask from point to point in an image. At each point (x,y) of the original image, the response of a filter is calculated by a pre defined relationship. All the filters values are pre defined and are a standard.

- Types of filters

Generally there are two types of filters. One is called as linear filters or smoothing filters and others are called as frequency domain filters.

INTRODUCTION TO FREQUENCY DOMAIN

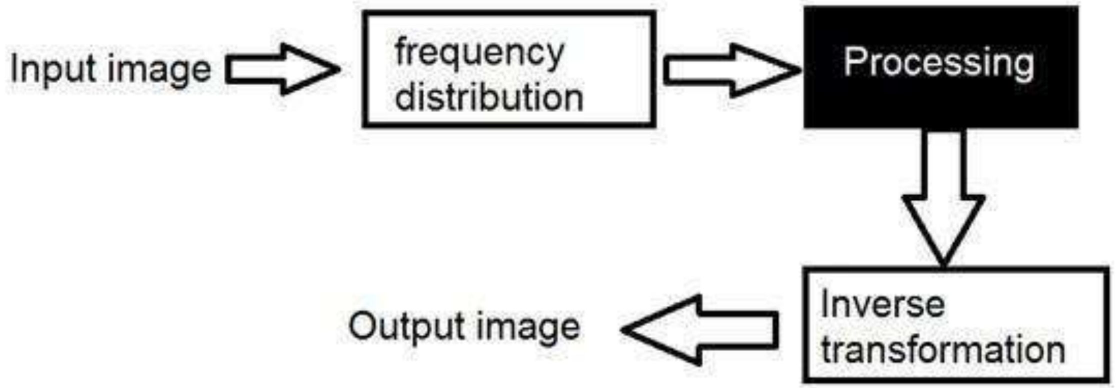
we are processing signals (images) in frequency domain.

the domains in which we have analysed a signal , we analyze it with respect to time. But in frequency domain we don’t analyze signal with respect to time, but with respect of frequency.

1. Spatial domain

FREQUENCY DOMAIN

- We first transform the image to its frequency distribution. Then our black box system perform what ever processing it has to performed, and the output of the black box in this case is not an image, but a transformation. After performing inverse transformation, it is converted into an image which is then viewed in spatial domain.
- It can be pictorially viewed as



TRANSFORMATION

1. A signal can be converted from time domain into frequency domain using mathematical operators called transforms. There are many kind of transformation that does this. Some of them are given below.

- Fourier Series
- Fourier transformation
- Laplace transform
- Z transform

FREQUENCY COMPONENTS

Any image in spatial domain can be represented in a frequency domain. But what do this frequencies actually mean.

We will divide frequency components into two major components.

High frequency components

1. High frequency components correspond to edges in an image.

Low frequency components

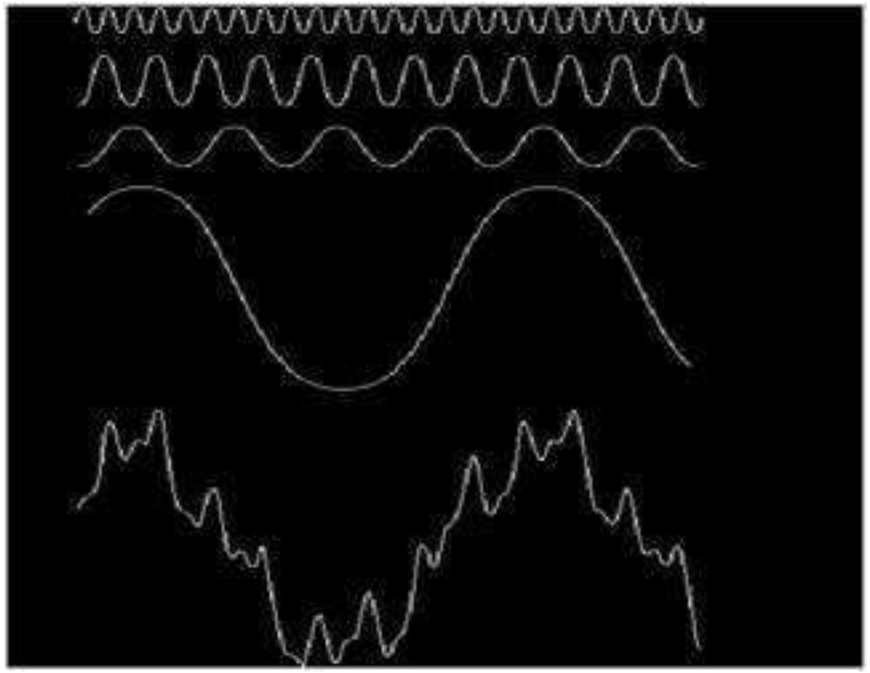
1. Low frequency components in an image correspond to smooth regions.

FOURIER

Fourier was a mathematician in 1822. He give Fourier series and Fourier transform to convert a signal into frequency domain.

Fourier series simply states that, periodic signals can be represented into sum of sines and cosines when multiplied with a certain weight . It further states that periodic signals can be broken down into further signals with the following properties.

- The signals are sines and cosines
- The signals are harmonics of each other



HOW IT IS CALCULATED

- Fourier series
- The Fourier series can be denoted by this formula.

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)}\, dx\, d$$

- The inverse can be calculated by this formula.

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j2\pi(ux+vy)}\, du\, dv.$$

DIFFERENCE BETWEEN FOURIER SERIES AND TRANSFORM

Although both Fourier series and Fourier transform are given by Fourier , but the difference between them is Fourier series is applied on periodic signals and Fourier transform is applied for non periodic signals.

Discrete Fourier transform

Since we are dealing with images, and in fact digital images, so for digital images we will be working on discrete Fourier transform

1. Consider the above Fourier term of a sinusoid. It include three things.

- Spatial Frequency
- Magnitude
- Phase

The spatial frequency directly relates with the brightness of the image. The magnitude of the sinusoid directly relates with the contrast. Contrast is the difference between maximum and minimum pixel intensity. Phase contains the color information.

$$F(u,v)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)e^{-j2\pi(ux/M+vy/N)}$$

The formula for 2 dimensional discrete Fourier transform

1. The discrete Fourier transform is actually the sampled Fourier transform, so it contains some samples that denotes an image. In the above formula f(x,y) denotes the image, and F(u,v) denotes the discrete Fourier transform. The formula for 2 dimensional inverse discrete Fourier transform is given below.

$$f(x,y)=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)e^{j2\pi(ux/M+vy/N)}$$

CONVOLUTION THEOREM

It

The relationship between the spatial domain and the frequency domain can be established by convolution theorem.

The convolution theorem can be represented as.

It can be stated as the convolution in spatial domain is equal to filtering in frequency domain and vice versa.

$$f(x,y)*h(x,y) \Leftrightarrow F(u,v)H(u,v)$$

$$f(x,y)h(x,y) \Leftrightarrow F(u,v)*H(u,v)$$

$$h(x,y) \Leftrightarrow H(u,v)$$

Convolution and Fourier Transform

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

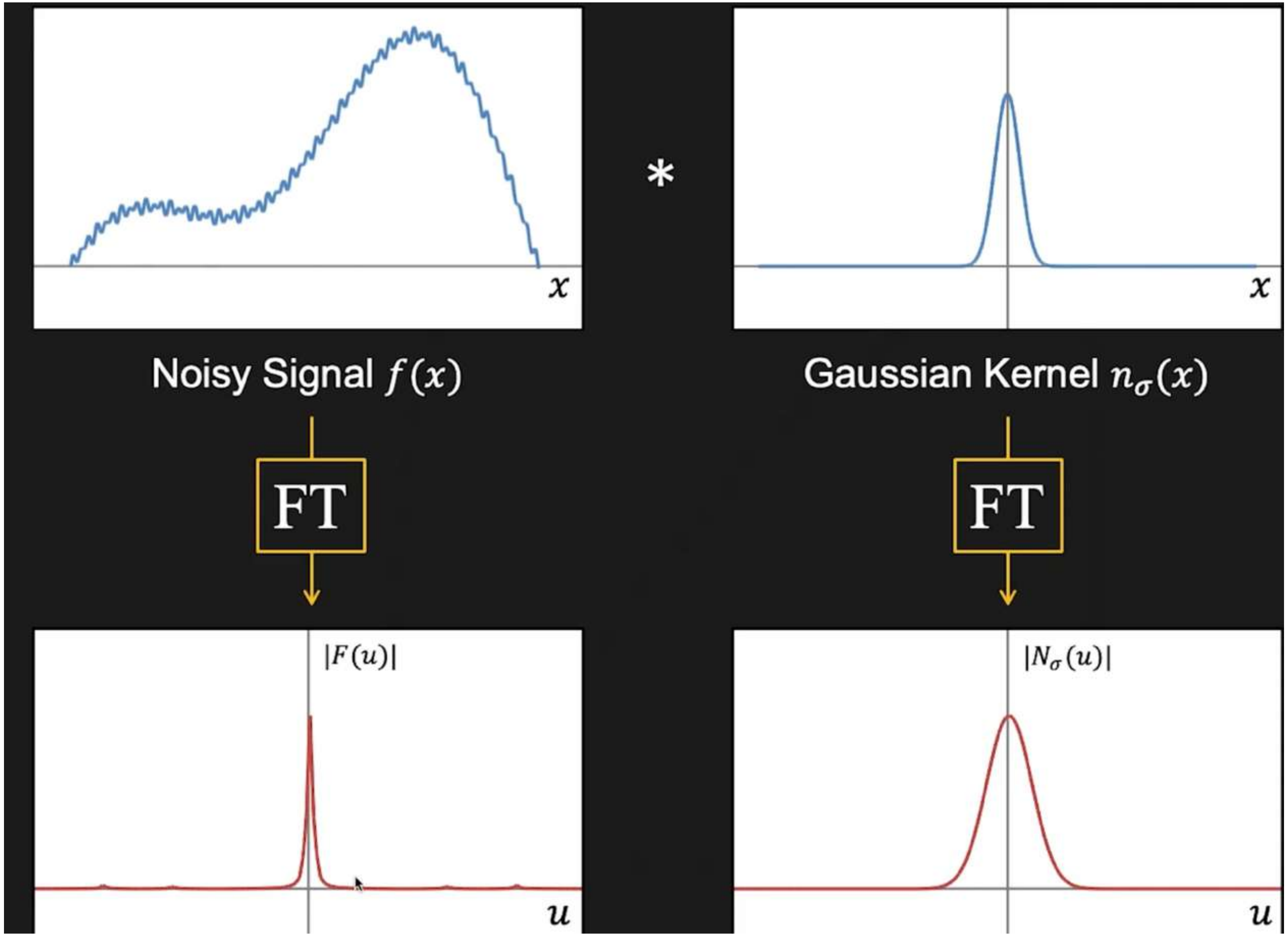
Fourier Transform of $g(x)$:

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

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

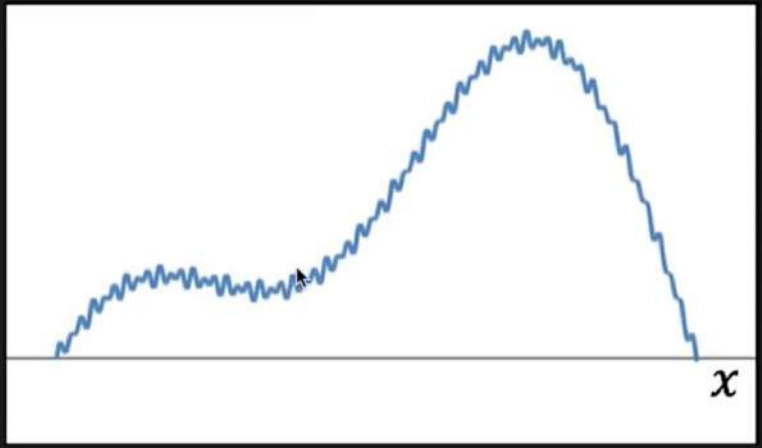
$$G(u) = \underbrace{\int_{-\infty}^{\infty} f(\tau)e^{-i2\pi u\tau}d\tau}_{F(u)} \; \underline{\int_{-\infty}^{\infty} h(x-\tau)e^{-i2\pi u(x-\tau)}\,dx}$$

with h in frequency domain.



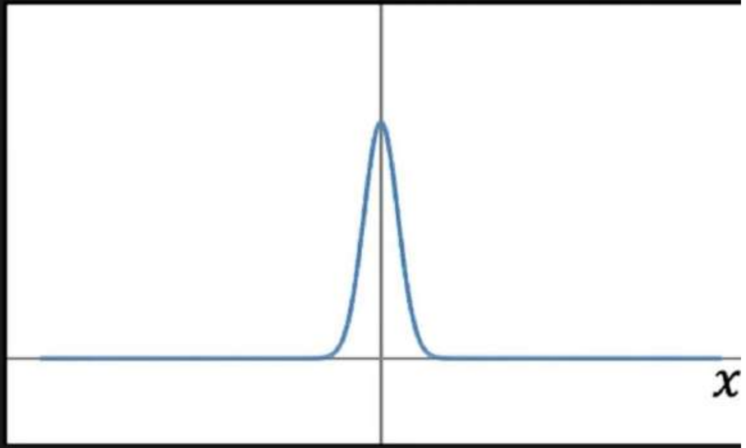
*
x
x
Noisy Signal $f(x)$
Gaussian Kernel $n_\sigma(x)$
FT
FT
$|F(u)|$
$|N_\sigma(u)|$
u
u

Gaussian Smoothing in Fourier Domain



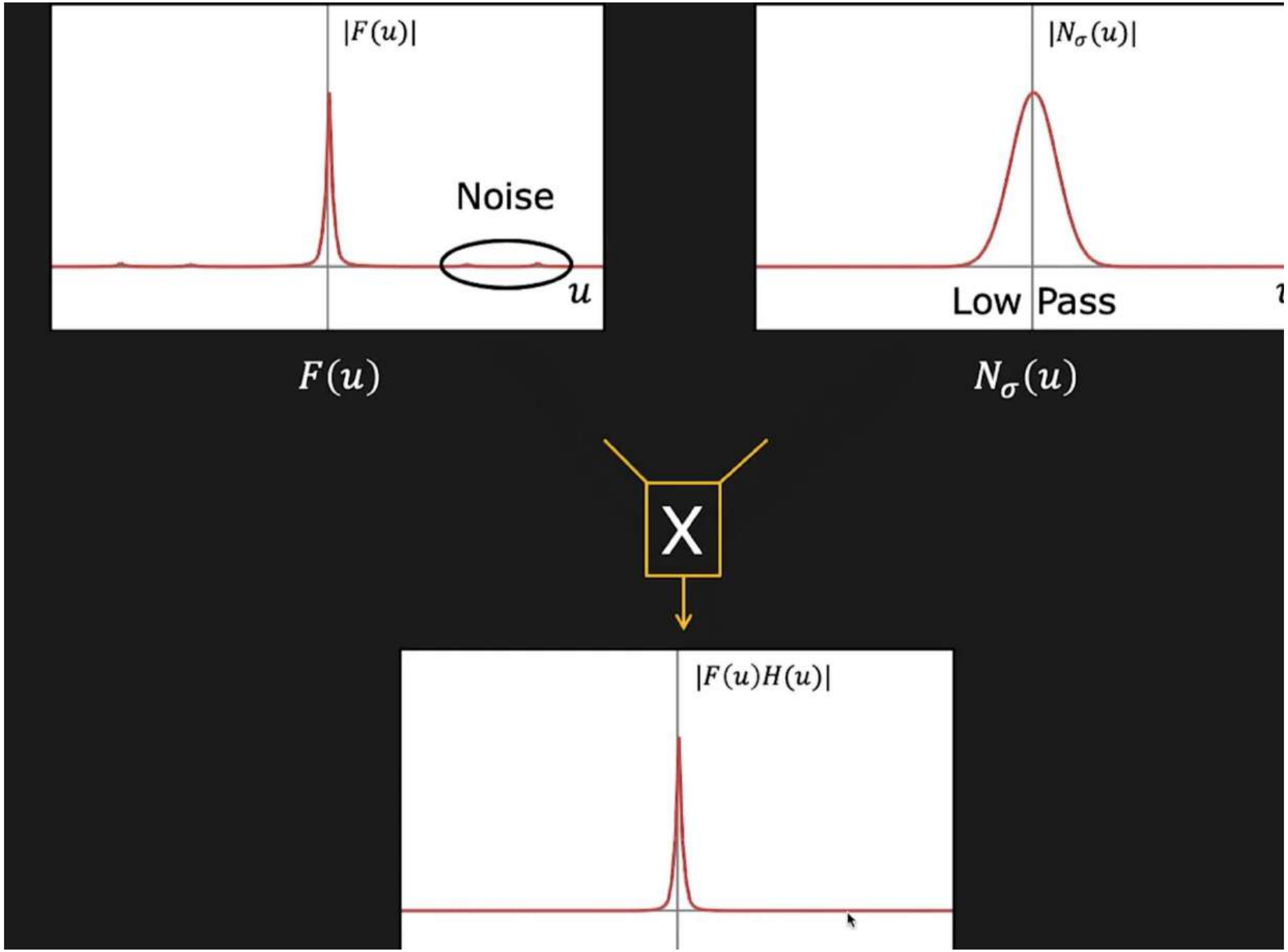
Noisy Signal $f(x)$

$*$



Gaussian Kernel $n_\sigma(x)$

Convolve the Noisy Signal with a Gaussian Kernel



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

THE FILTERING IN FREQUENCY DOMAIN CAN BE REPRESENTED AS FOLLOWING:

FILTERS

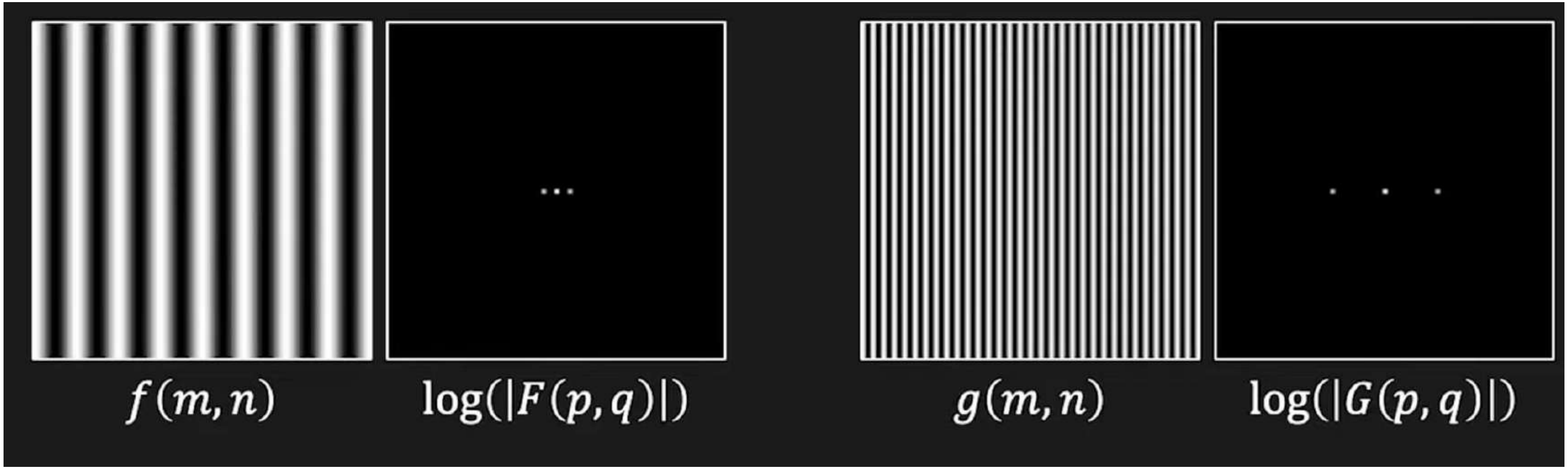
The concept of filter in frequency domain is same as the concept of a mask in convolution.

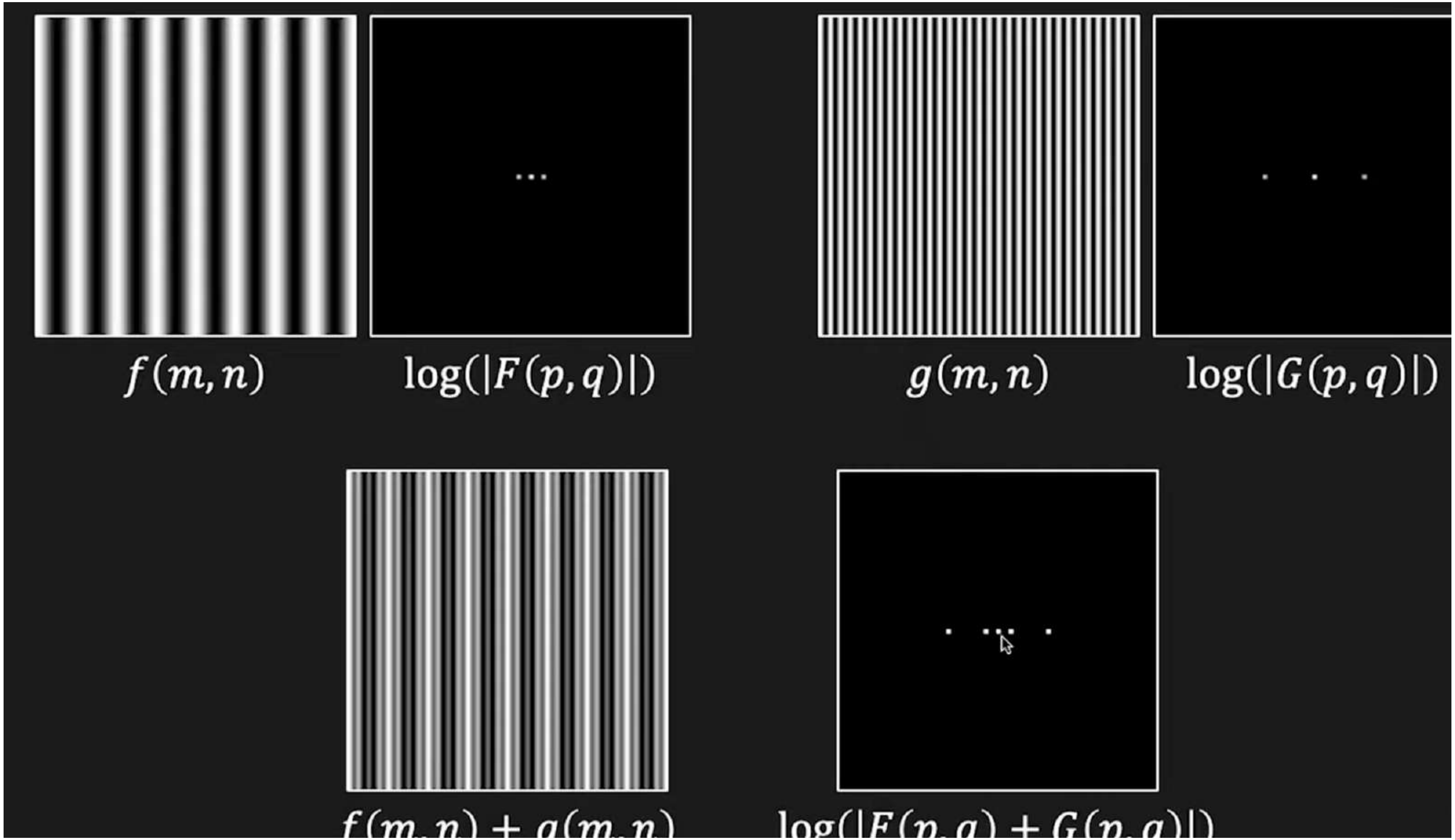
After converting an image to frequency domain, some filters are applied in filtering process to perform different kind of processing on an image. The processing include blurring an image, sharpening an image e.t.c.

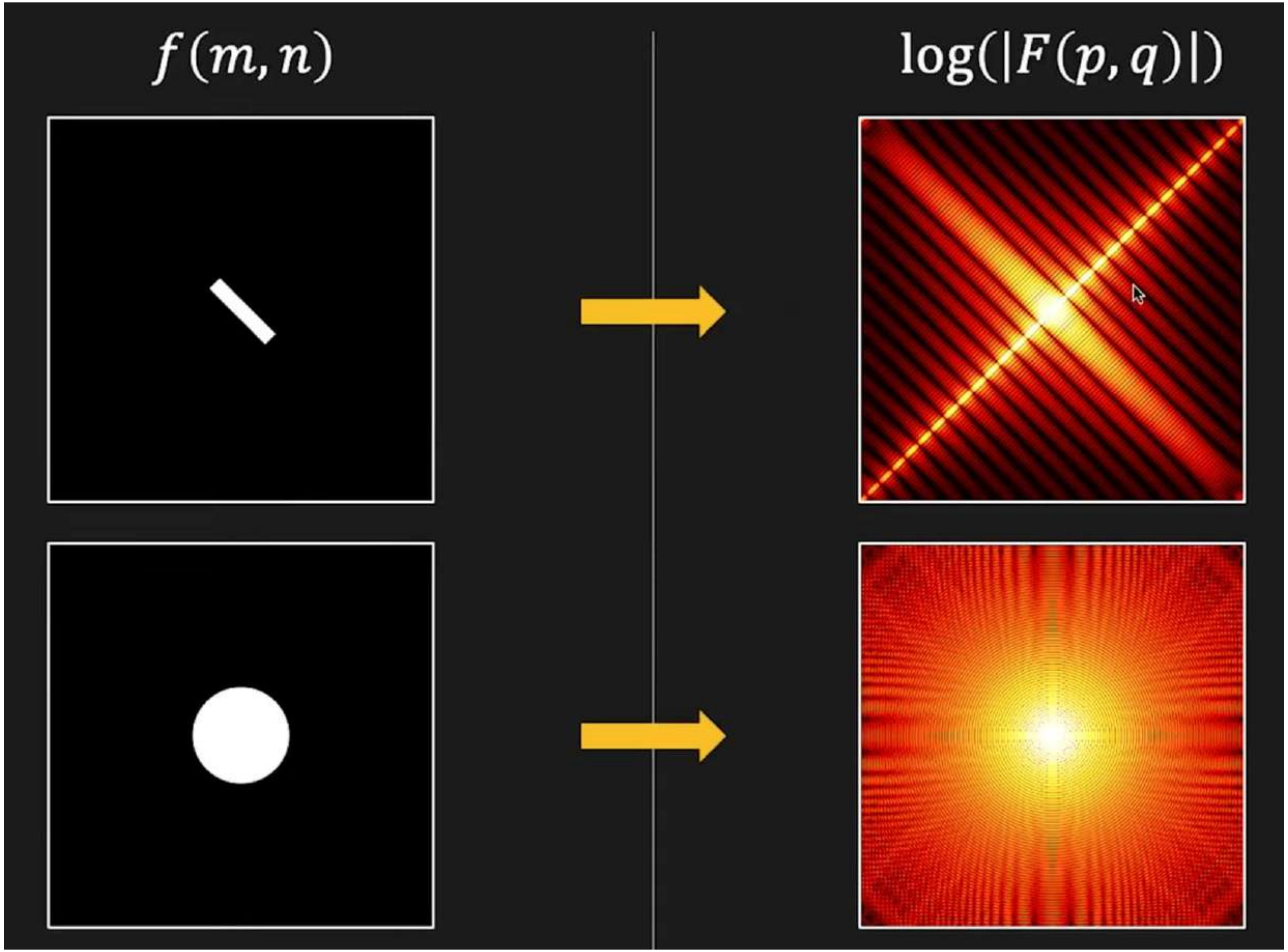
The common type of filters for these purposes are:

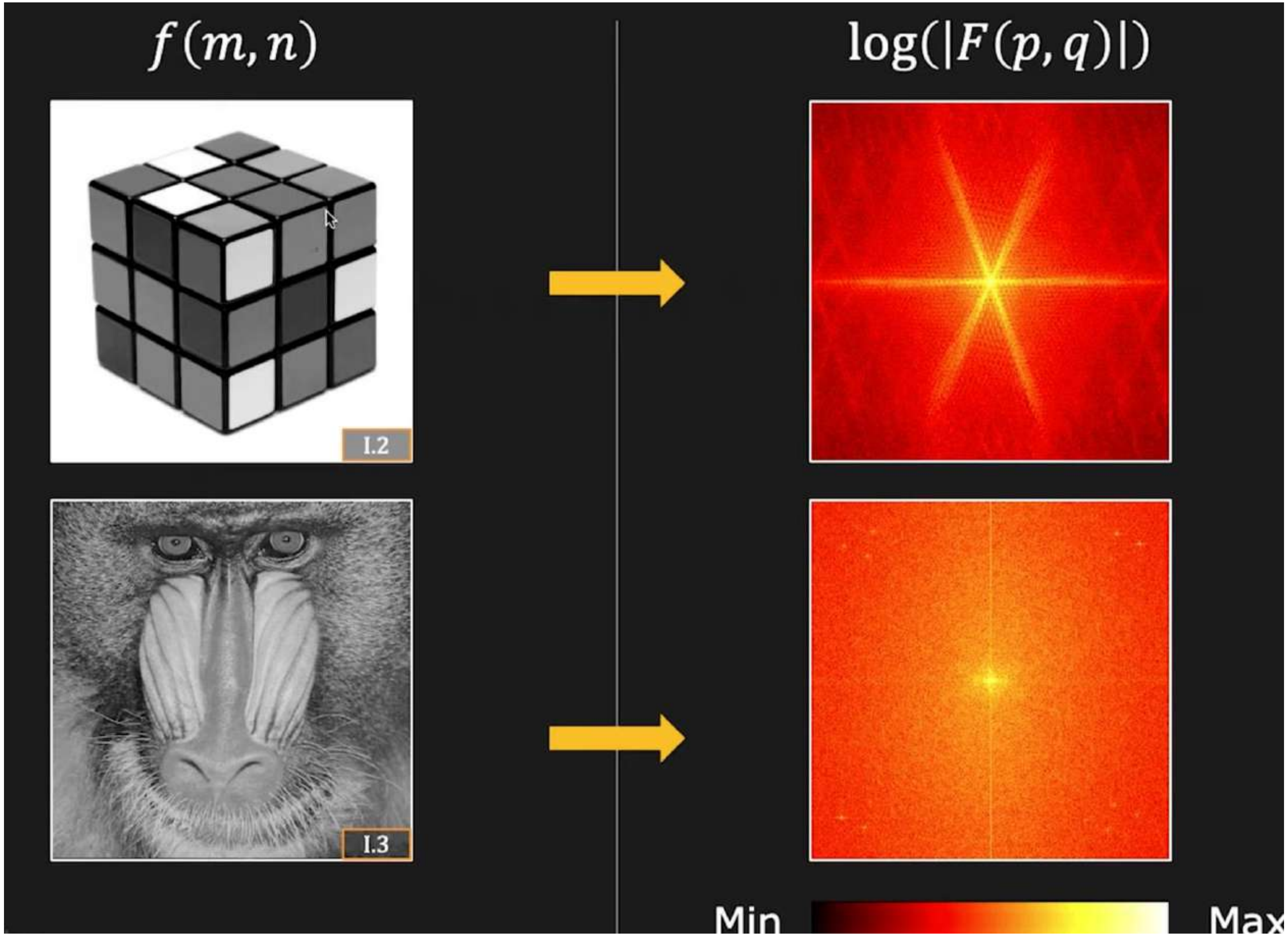
- Ideal high pass filter
- Ideal low pass filter
- Gaussian high pass filter
- Gaussian low pass filter

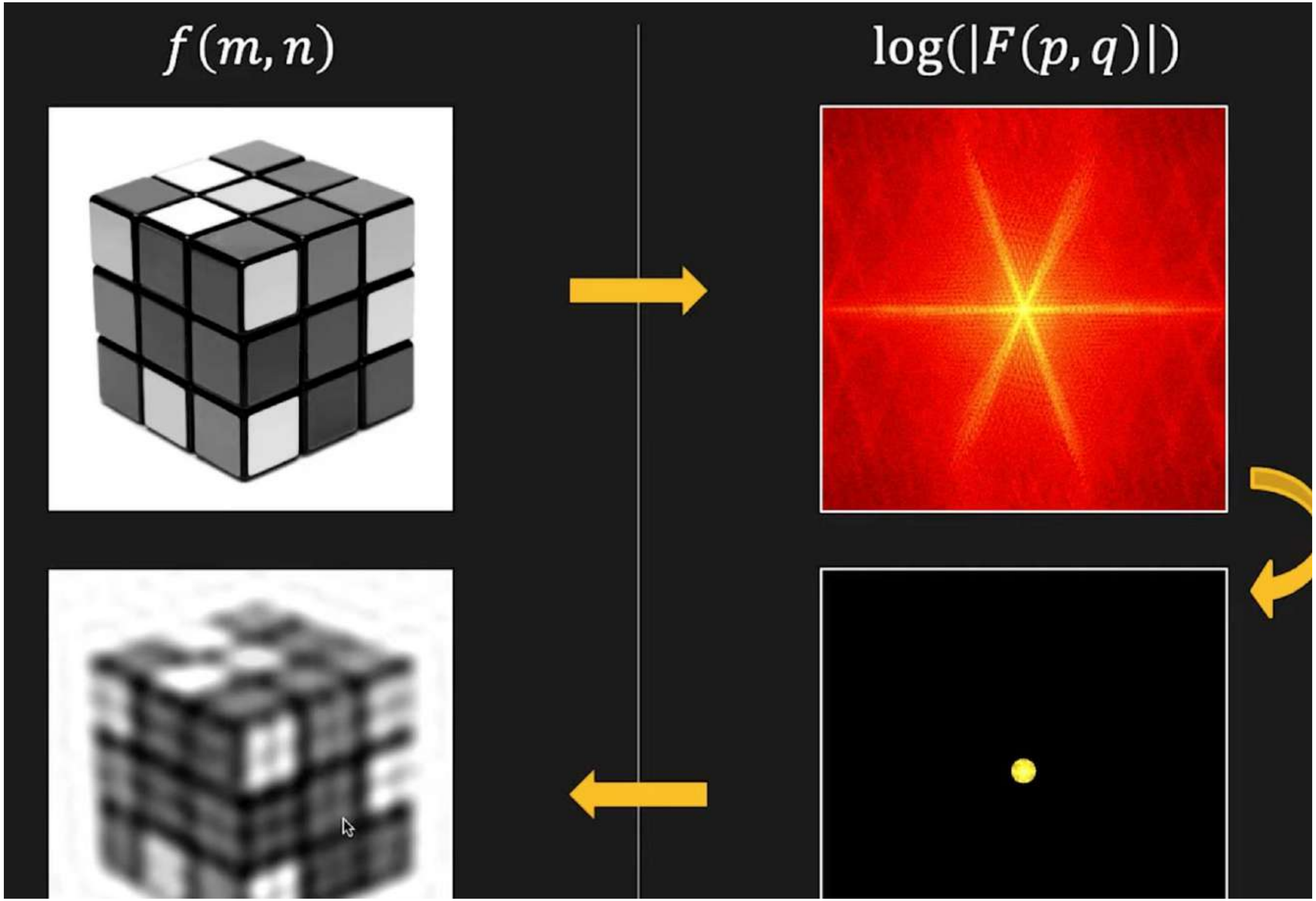
FILTERS



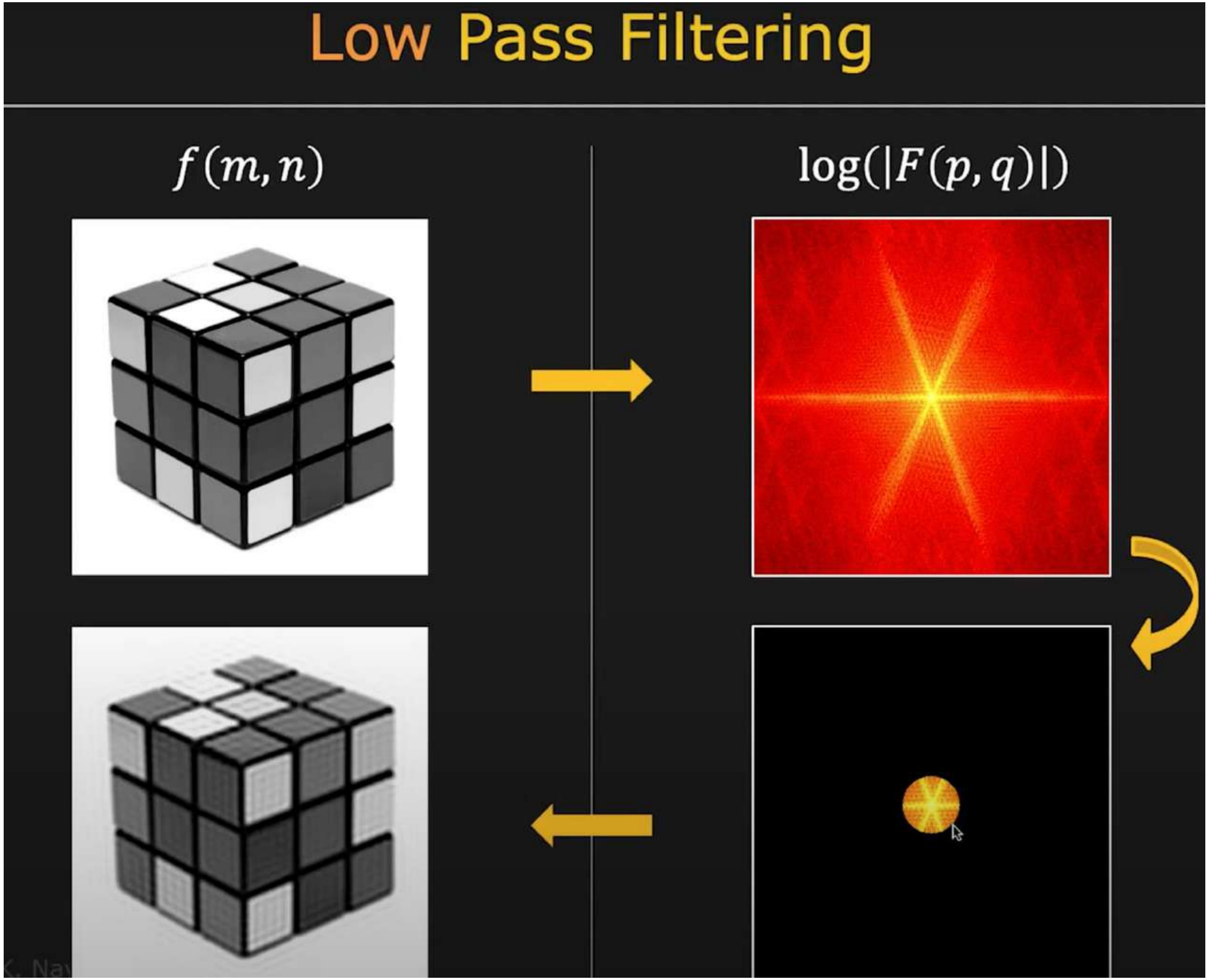

$f(m,n)$
$\log(|F(p,q)|)$
$g(m,n)$
$\log(|G(p,q)|)$
$f(m,n) + g(m,n)$
$\log(|F(p,q) + G(p,q)|)$

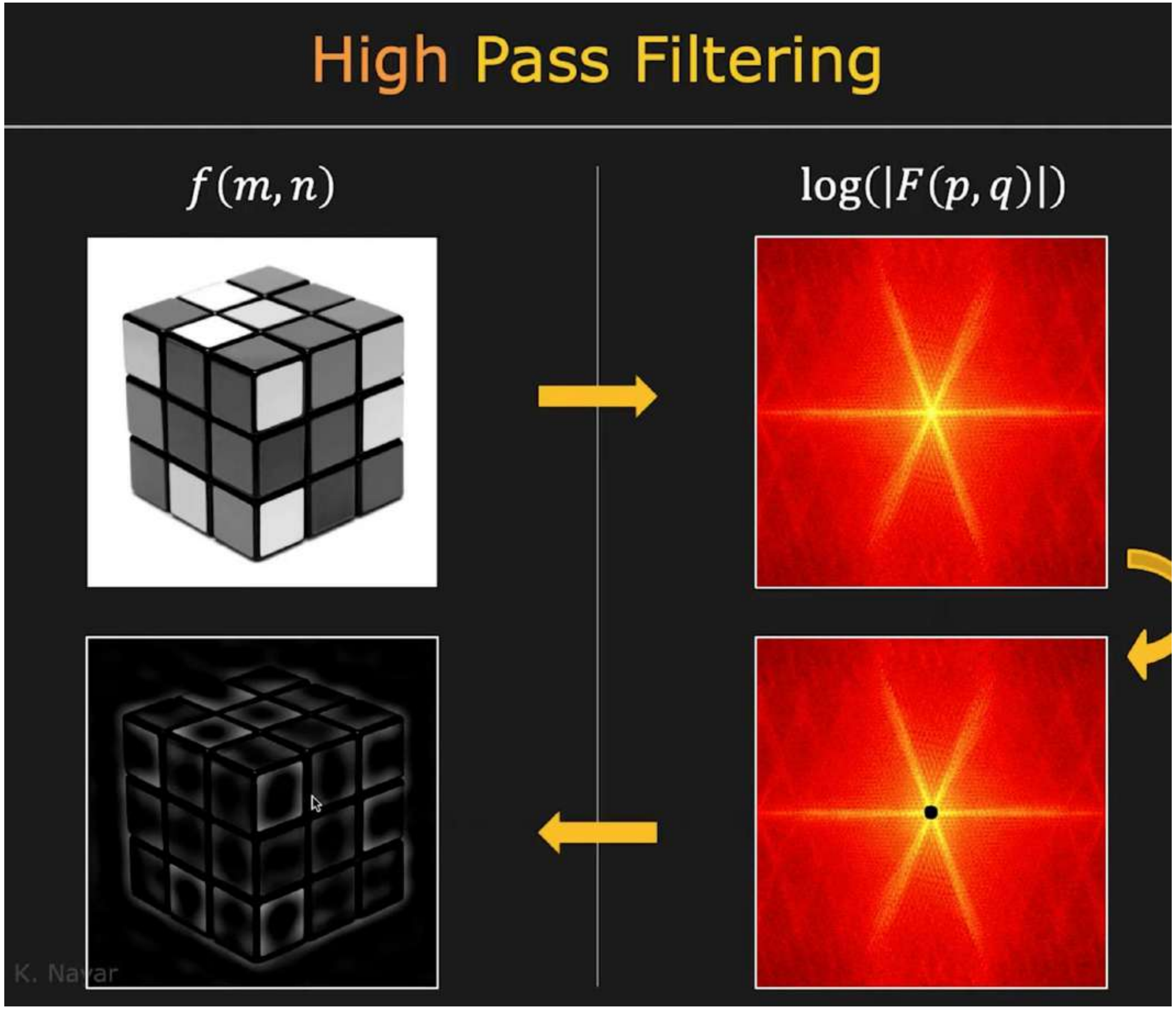

$f(m,n)$
$\log(|F(p,q)|)$

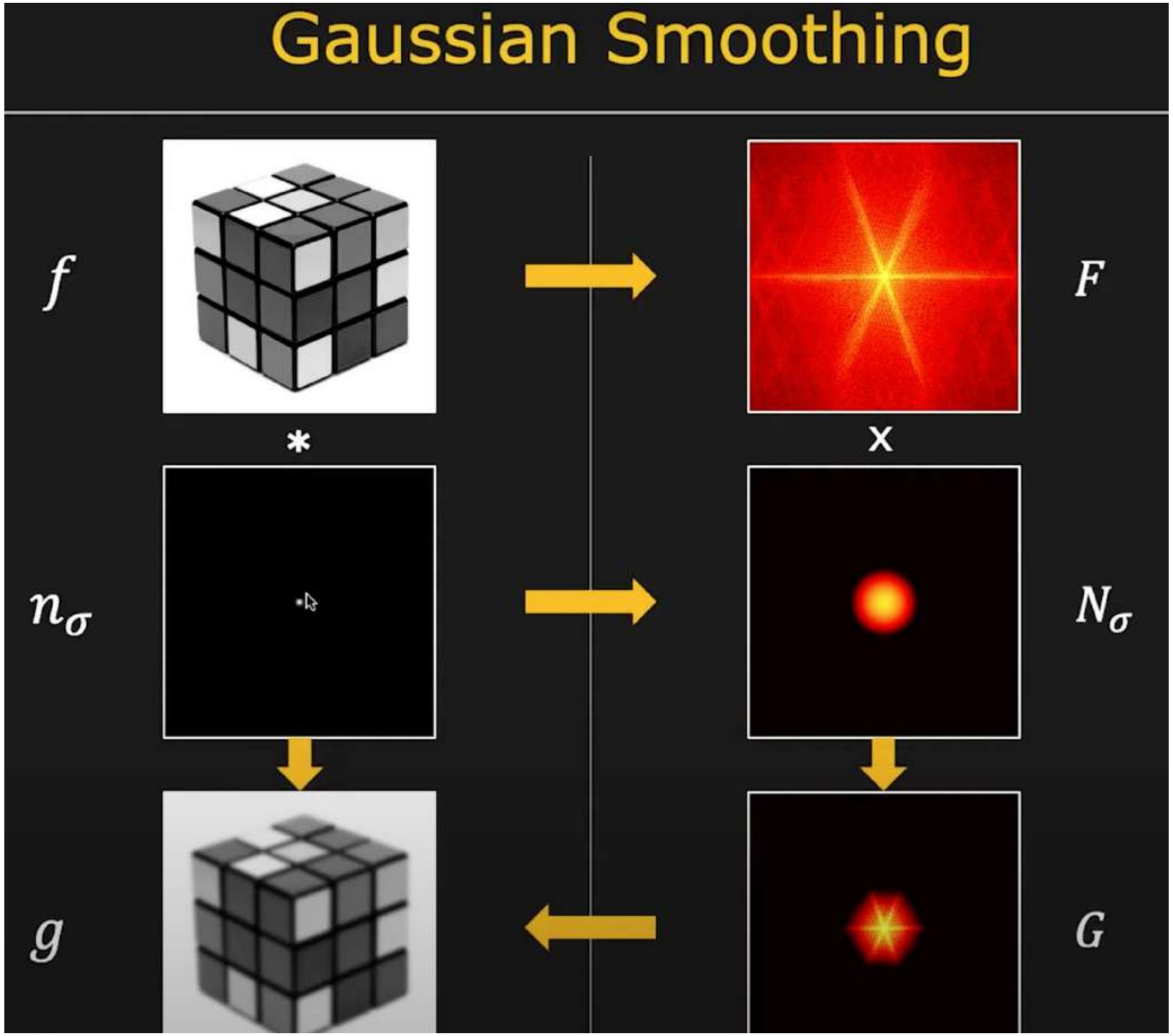

$f(m,n)$
$\log(|F(p,q)|)$
I.2
I.3
Min
Max

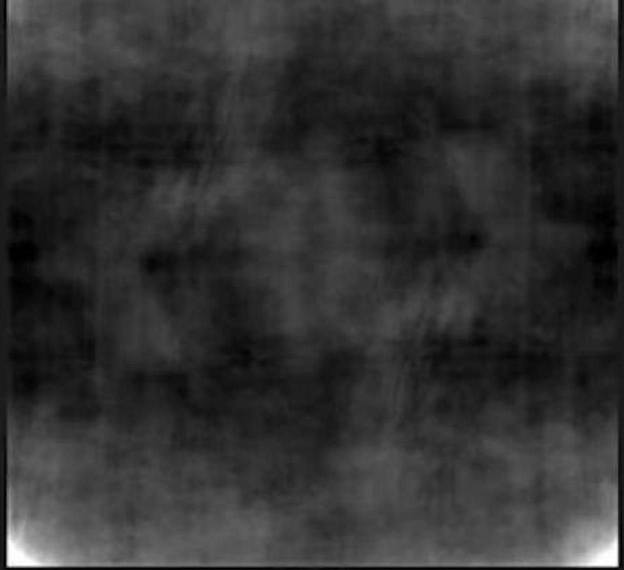


$f(m,n)$
$\log(|F(p,q)|)$


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


High Pass Filtering
$f(m,n)$
$\log(|F(p,q)|)$
K. Nayar

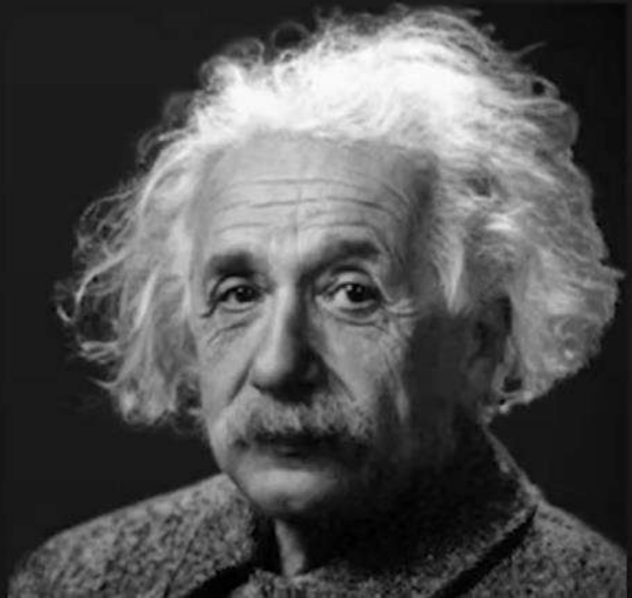

Gaussian Smoothing
f
F
*
x
n_σ
N_σ
g
G

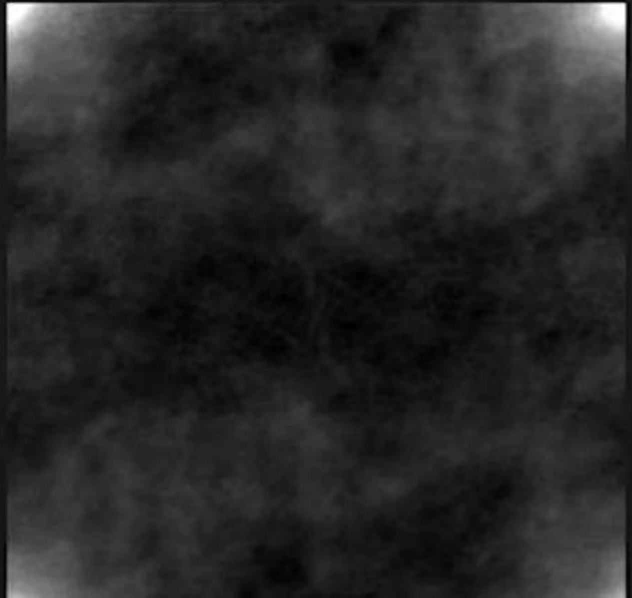


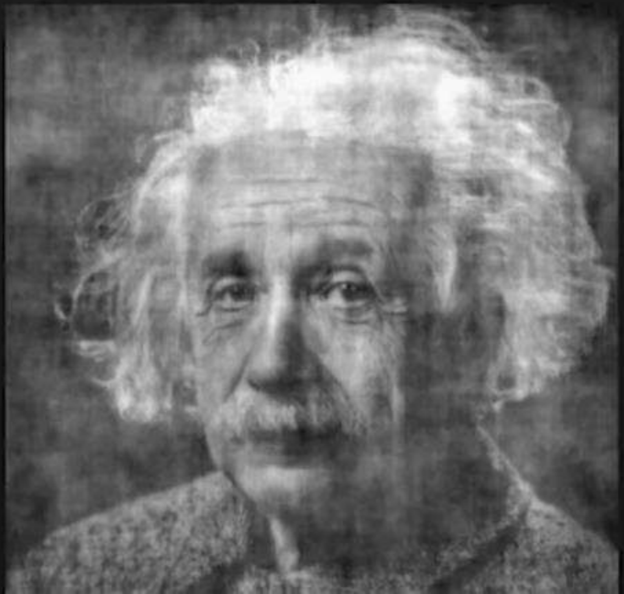
Original Image

Magnitude Preserved,
Phase Set to Zero

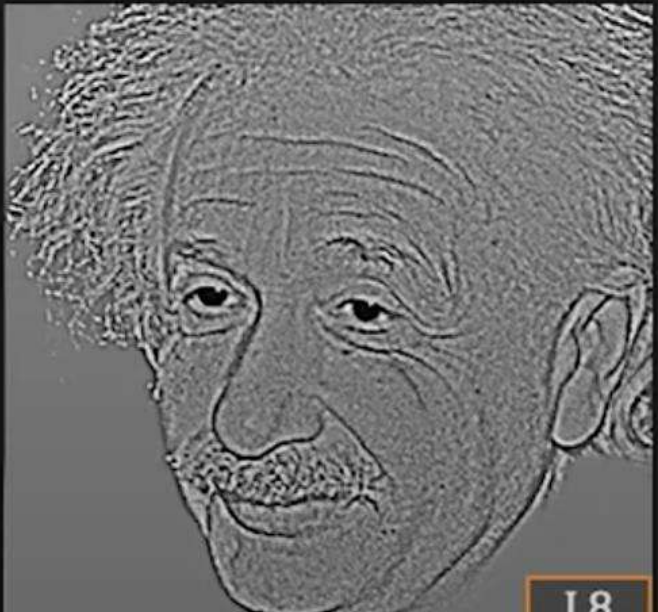
Phase Preserved,
Magnitude Set to Averag
of Natural Images







Low Freq Only



HOMOMORPHIC FILTERING

1. **Homomorphic filtering** is a generalized technique for signal and image processing.
2. Homomorphic filtering is sometimes used for image enhancement.
3. It simultaneously normalizes the brightness across an image and increases contrast. Here homomorphic filtering is used to remove multiplicative noise.
4. To make the illumination of an image more even, the high-frequency components are increased and low-frequency components are decreased, because the high-frequency components are assumed to represent mostly the reflectance in the scene
5. whereas the low-frequency components are assumed to represent mostly the illumination in the scene. That is, high-pass filtering is used to suppress low frequencies and amplify high frequencies, in the log-intensity domain

Homomorphic filtering can be used for improving the appearance of a grayscale image by simultaneous intensity range compression (illumination) and contrast enhancement (reflection).

Where,

m = image,

i = illumination,

r = reflectance

We have to transform the equation into frequency domain in order to apply high pass filter. However, it's very difficult to do calculation after applying Fourier transformation to this equation because it's not a product equation anymore. Therefore, we use 'log' to help solving this problem.

$$\ln(m(x,y)) = \ln(i(x,y)) + \ln(r(x,y))$$

Then, applying Fourier transformation

$$F(ln(m(x,y))) = F(ln(i(x,y))) + F(ln(r(x,y)))$$

$$M(u,v) = I(u,v) + R(u,v)$$

Next, applying high-pass filter to the image. To make the illumination of an image more even, the high-frequency components are increased and low-frequency components are decrease.

$$N(u,v) = H(u,v) \bullet M(u,v)$$

Where

H = any high-pass filter

N = filtered image in frequency domain

Afterward, returning frequency domain back to the spatial domain by using inverse Fourier transform.

$$n(x, y) = invF(N(u, v))$$

Finally, using exponential function to eliminate the log we used at the beginning to get the enhanced image

$$n(x, y) = invF(N(u, v))$$

Smoothing Spatial Filters:

→ Used for Blurring & Noise reduction

→ Blurring: Removal of small details from an Image. Prior to object extraction.

→ Noise reduction: blurring with a linear (or) Non linear filters.

SMOOTHING LINEAR FILTERS:

→ o/p → average of pixels contained in the neighborhood → Filter mask

→ Averaging Filters (or) low pass Filters

Ex:-

(a)

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

(b)

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

3x3 Smoothing Filter mask

→ fig (a) → Standard average of pixel value

→ m×n mask → (1/m×n)

→ Box Filters

→ fig (b) → weighted avg

→ pixel at the center of mask → more importance

→ This is to reduce blurring during smoothing process

→ general Implementation for Image → M×N & mask → m×n

$$g(x,y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)}$$

$x = 0, 1, 2, \ldots M-1$ & $y = 0, 1, 2 \ldots N-1$

ORDER-STATISTICS Filters:

→ non linear Spatial filters
→ response → ordering [Ranking] the pixels → Image
→ Replacing the Center pixel value with value determined by ranking result.

1. Median Filter:

→ Replaces the value of a pixel by the median of the gray level.
→ Most popular → excellent noise reduction
→ less blurring
→ Effective for Impulse noise
↓

Ex:
⇒

10	20	20
20	$\boxed{15}$	20
20	25	100

⇒

10	20	20
20	20	20
20	25	100

10, 15, 20, 20, 20, 20, 20, 25, 100
↑

2. Max filter:

→ Finding the brightest poin

$$R = \max \{Z_k \mid \underbrace{k = 1, 2, 3 \ldots 9}_{3\times3}\}$$

3. Min filter:

→ Finding the darkest poin

$$R = \min \{Z_k \mid k = 1, 2, 3 \ldots 9$$

Ex:
max. Value : 100 – brightest p

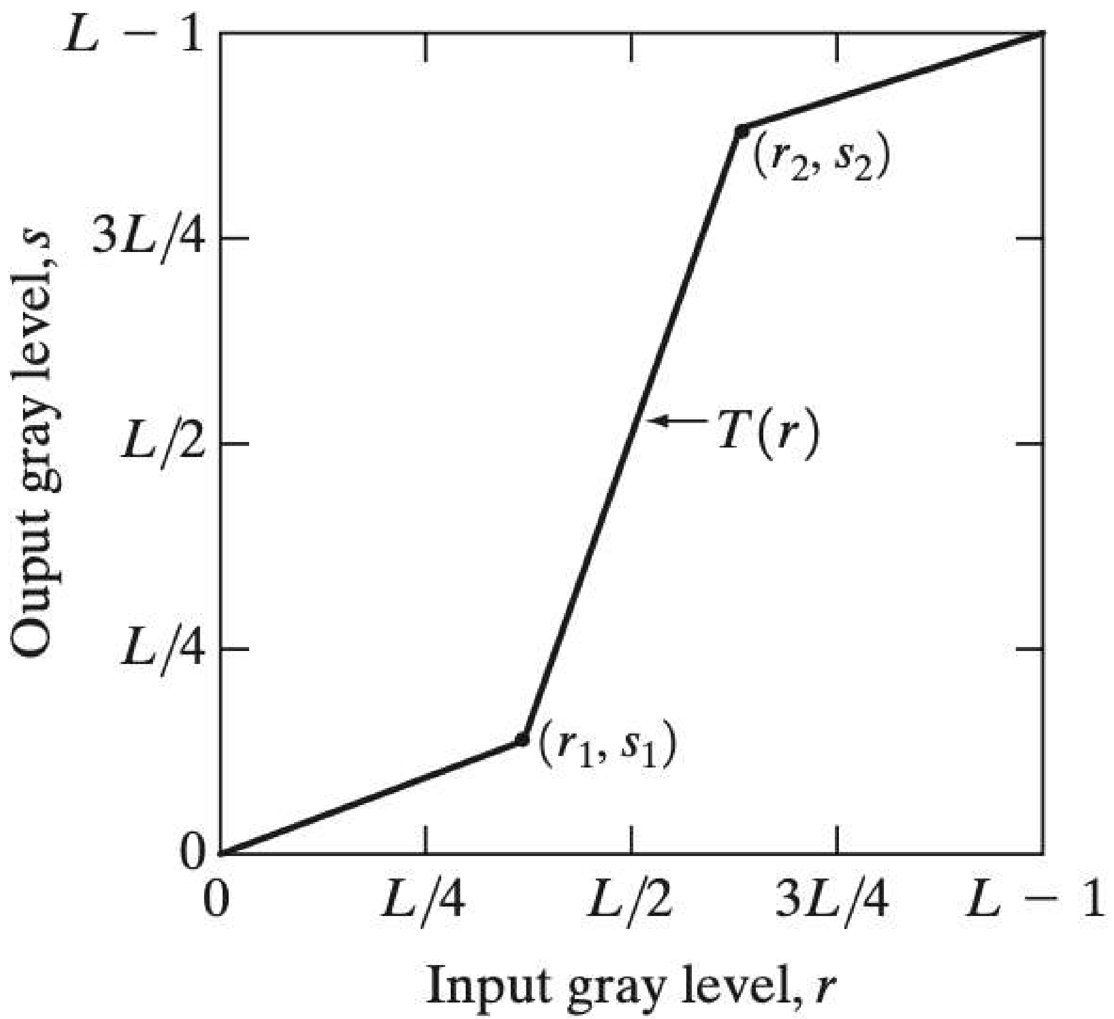
PIECEWISE-LINEAR TRANSFORMATION FUNCTIONS

- complementary approach to the methods discussed in the previous three sections
- piecewise functions can be arbitrarily complex (more options to design)
- **Contrast stretching**
- **Gray-level slicing**
- **Bit-plane slicing**

CONTRAST STRETCHING

- Low-contrast images can result from poor illumination, lack of dynamic range in the imaging sensor,

- contrast stretching is to increase the dynamic range of the graylevels in the image being processed.



1. **Dynamic range** tells us how much variation a particular **imaging** system can capture, and this in turn helps us to understand how well the system ... It is difference between highest intensity and lowest intensity pixel

1. shows a typical transformation used for contrast stretching.
2. The locations of points (r1 , s1) and (r2 , s2) control the shape of the transformation
3. If r1=s1 and r2=s2 , the transformation is a linear function that produces no changes in gray levels.
4. If r1=r2 , s1=0 and s2=L-1, the transformation becomes a *thresholding function*
5. Intermediate values of (r1 , s1) and (r2 , s2) produce various degrees of spread in the graylevels of the output image,

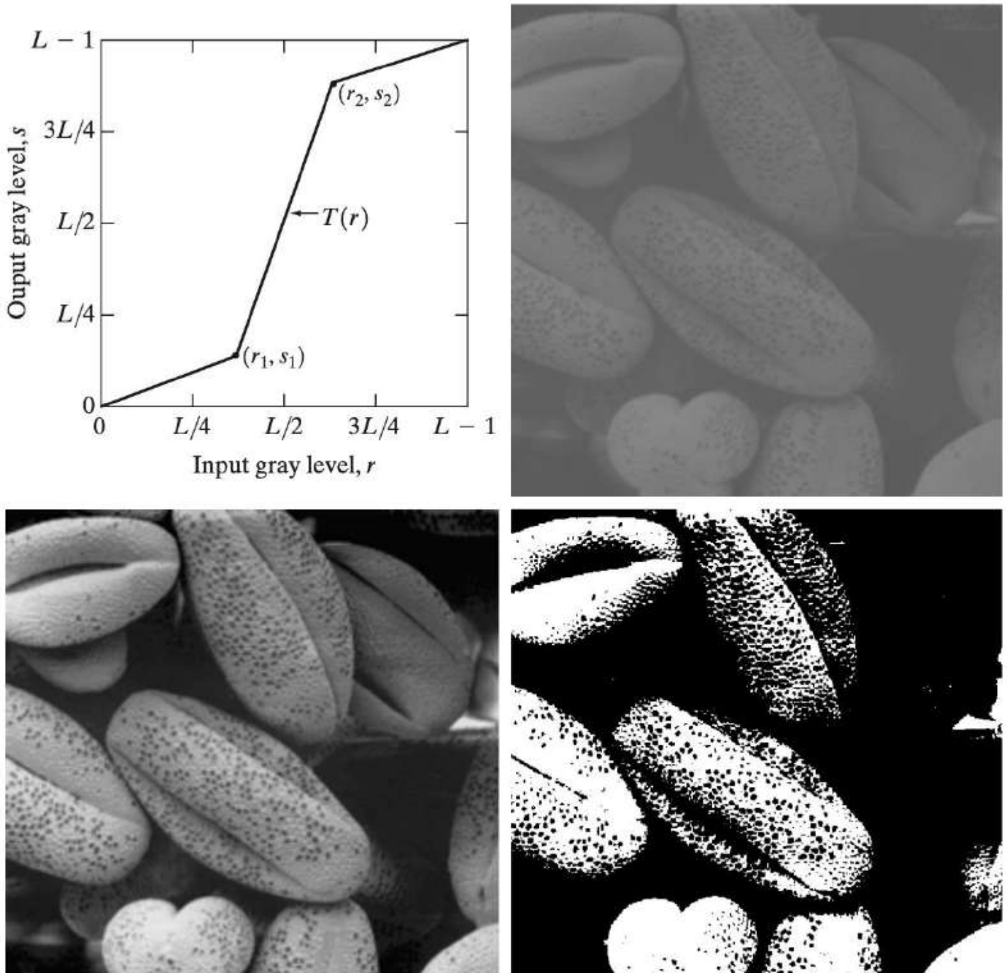
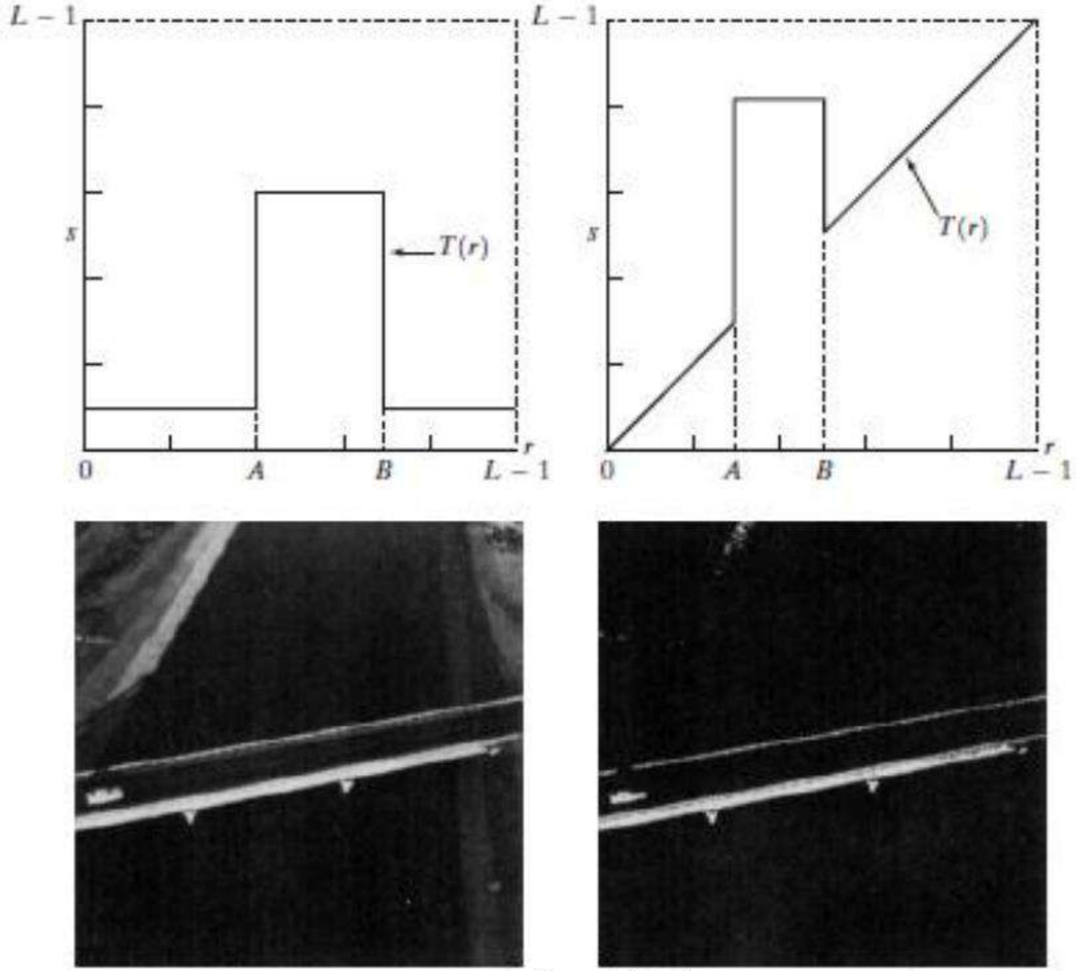


FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

GRAY-LEVEL SLICING

1. ***It manipulates group of intensity levels in an image up to specific range by diminishing rest or by leaving them alone. This transformation is applicable in medical images and satellite images such as X-ray flaws, CT scan.***
2. One approach is to display a high value for all graylevels in the range of interest and a low value for all other graylevels.
3. The second approach, based on the transformation brightens the desired range of graylevels but preserve graylevels unchanged.

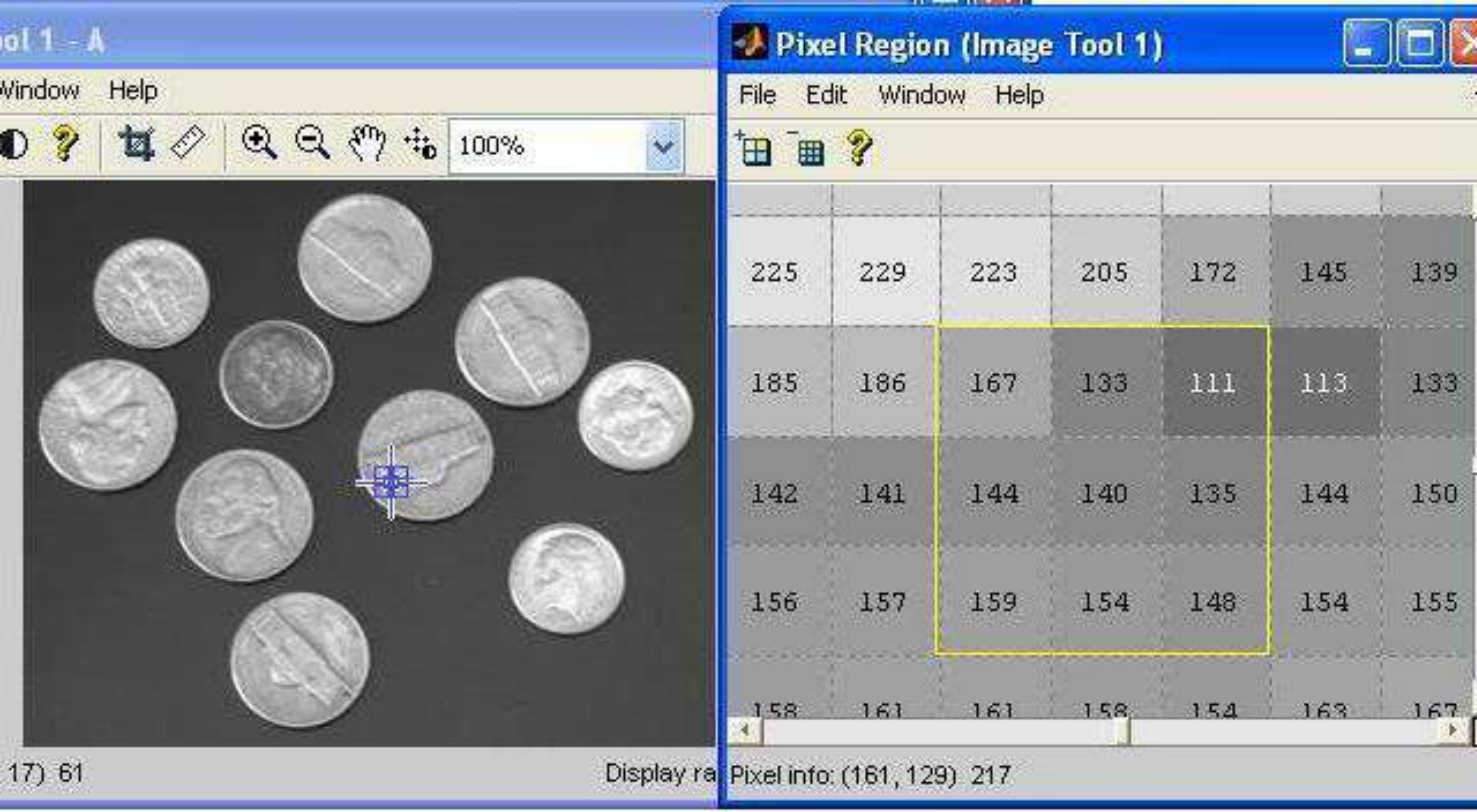


a b
c d

FIGURE 3.11 (a) This transformation highlights range [A, B] of gray levels and reduces all others to a constant level. (b) This transformation highlights range [A, B] but preserves all other levels. (c) An image. (d) Result of using the transformation in (a).

BIT-PLANE SLICING

1. Digitally, an image is represented in terms of pixels.
2. These pixels can be expressed further in terms of bits.


ool 1 - A
Window Help
100%
17) 61
Display ra
Pixel Region (Image Tool 1)
File Edit Window Help
225 229 223 205 172 145 139
185 186 167 133 111 113 133
142 141 144 140 135 144 150
156 157 159 154 148 154 155
Pixel info: (161, 129) 217

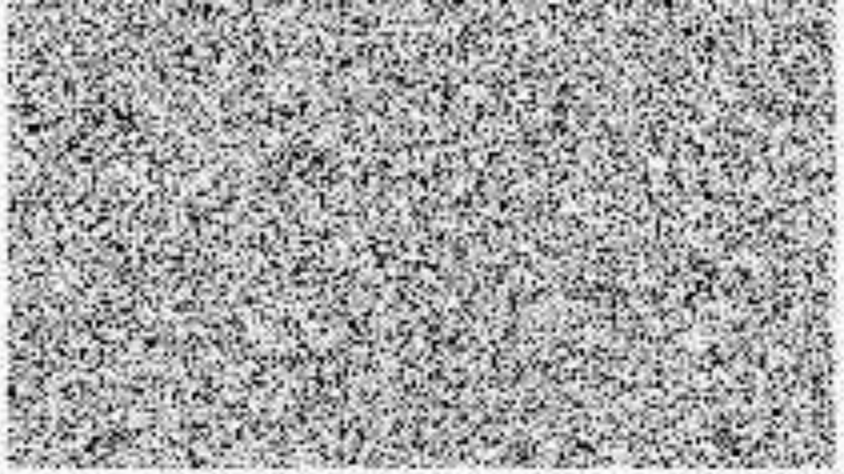
10100111	10000101	01101111
10010000	10001100	10000111
10011111	10011010	10010100

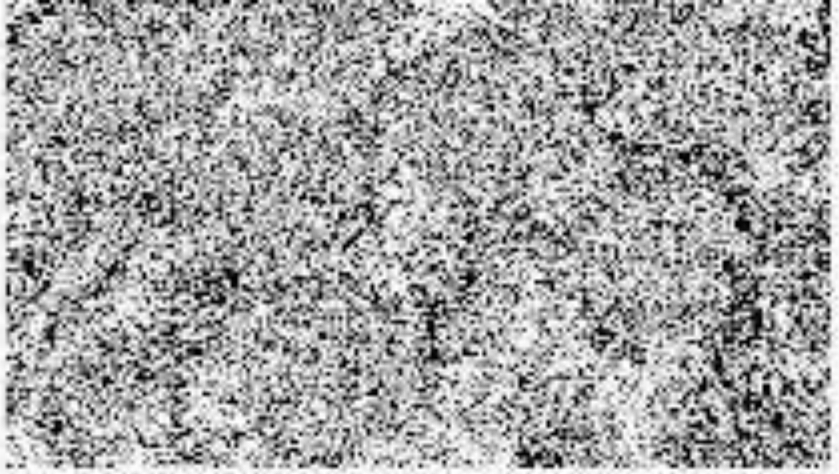
- Consider the pixels that are bounded within the yellow line. The binary formats for those values are

1. The binary format for the pixel value 167 is 10100111
2. Similarly, for 144 it is 10010000
3. This 8-bit image is composed of eight 1-bit planes.
4. Plane 1 contains the lowest order bit of all the pixels in the image.

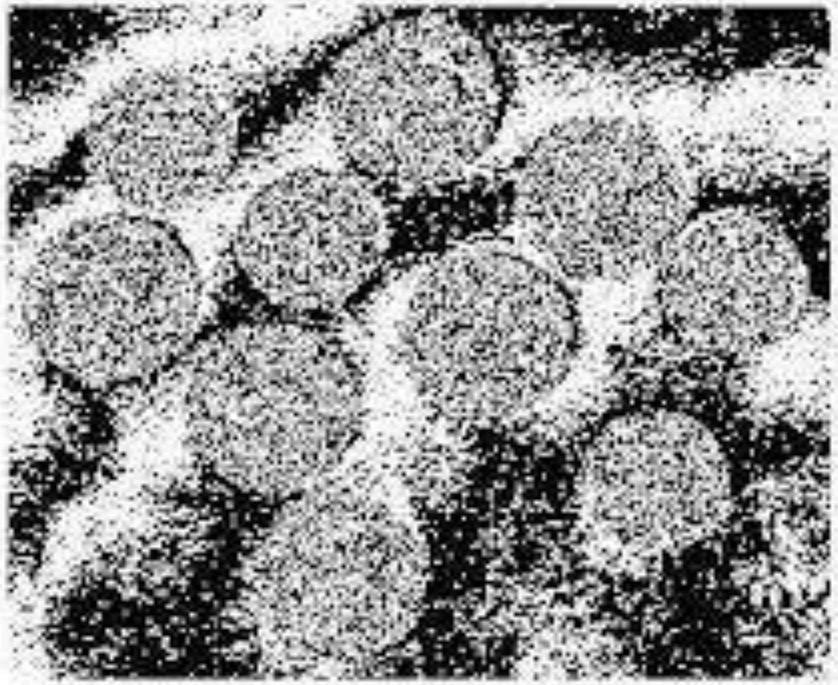
10100111	10000101	01101111
10010000	10001100	10000111
10011111	10011010	10010100

101001111	100001011	011011111
100100000	100011000	100001111
100111111	100110100	100101000

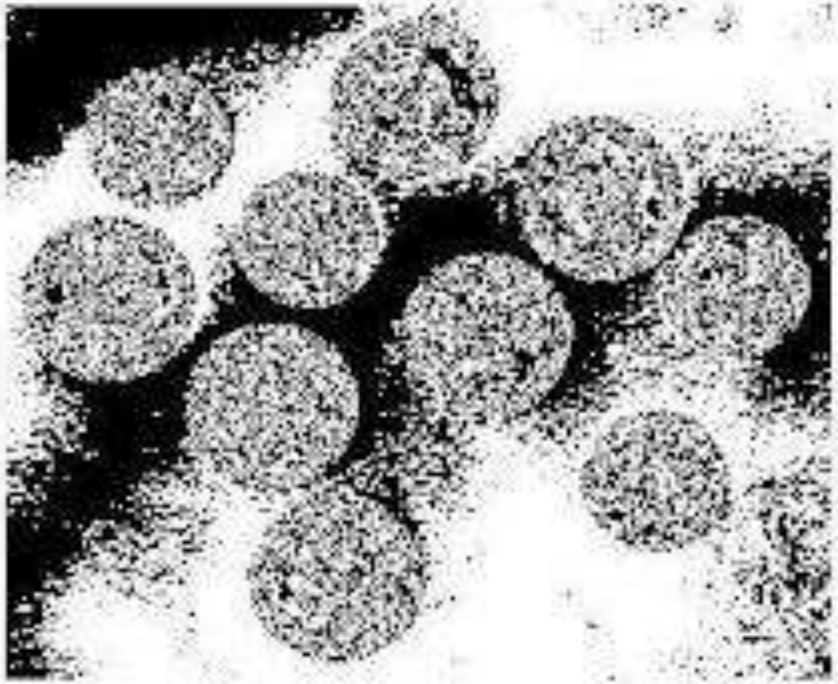


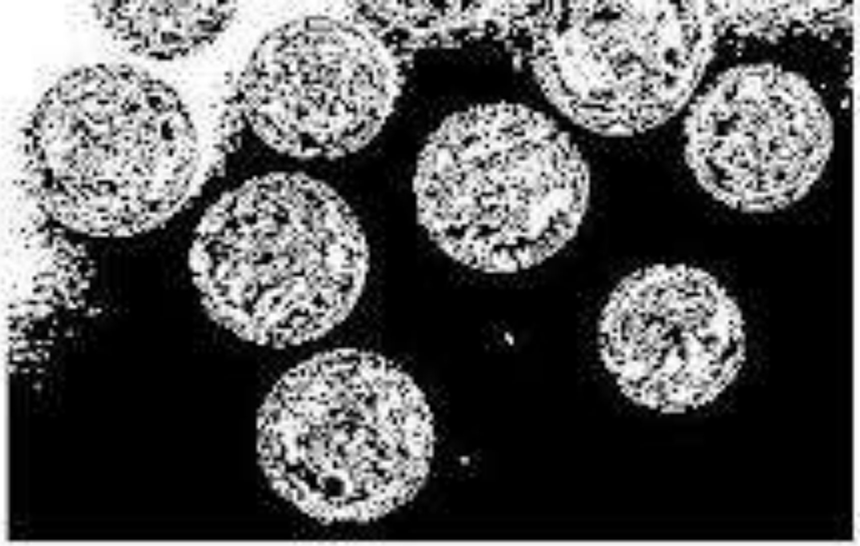


Bit plane 3



Bit plane 4





Bit plane 7

Bit plane 8