UNIT 2
IMAGE ENHANCEMENT

II AIML

21CSE251T - DIP

Syllabus

- **Spatial Domain:**
- Basic relationship between pixels
- Basic Gray level Transformations
- Histogram Processing
- Smoothing spatial filters
- Sharpening spatial filters.
- **Frequency Domain:**
  - Smoothing frequency domain filters
  - Sharpening frequency domain filters
- Homomorphic filtering.

Recap -- TO THE FOURIER TRANSFORM AND THE FREQUENCY DOMAIN

- The one-dimensional Fourier transform and its inverse
  - Fourier transform (continuous case)(single dimension)

$$\boxed{F(u)=\int_{-\infty}^{\infty} f(x)e^{-j2\pi ux}dx \quad \text{where } j=\sqrt{-1}}$$

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

  - Inverse Fourier transform:

$$\boxed{f(x)=\int_{-\infty}^{\infty} F(u)e^{j2\pi ux}du}$$

- The two-dimensional Fourier transform and its inverse
  - Fourier transform (2D-continuous case)

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

  - Inverse Fourier transform:

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

1D and 2D Discrete Fourier Transform
1D discrete Fourier transform is

$$F(u) = \sum_{x=0}^{M-1} f(x)\ e^{-2\pi j\left(\frac{xu}{M}\right)}$$

The inverse FT is,

$$f(x) = 1/M \sum_{u=0}^{M-1} \mathrm{F}(u)\ e^{2\pi j\left(\frac{xu}{M}\right)}$$

The *2D dimensions* Discrete Fourier Transform is,

$$F(u,v) = \sum_{N=0}^{N-1} \sum_{M=0}^{M-1} f(x,y)\ e^{-2\pi j\left(\frac{xu}{M}+\frac{yv}{N}\right)}$$

And, the *inverse* of 2D DFT would be,

$$f(x,y) = 1/MN \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} F(u,v)\ e^{2\pi j\left(\frac{xu}{M}+\frac{yv}{N}\right)}$$

Relationship between pixels

- 4 Neighbors
- Diagonal Neighbors
- Adjacency Pixels
- Digital Path
- Connected Set
- Region, Bounday, Contour, Edge

Gray level Transformations

- Gray-level transformations refers to the techniques used to adjust the pixel values (or intensity levels) of an image.
- These transformations are applied to improve the visual quality of an image or to extract useful information.
- The transformations operate on the intensity levels of pixels and alter the image's appearance, enhancing contrast, brightness, or other features.
- **Types of Gray Level Transformations**
- a) Linear (Negative and Identity)Transformations
- b) Logarithmic(log and inverse-log) Transformations
- c) Power – law(nth power and nth root) Transformations

Quiz - 1

- Gray-level transformations refers to the techniques used to adjust the ______________ of an image.
- A. depth
- B. breadth
- C. width
- D. pixel

Quiz - 1

- Gray-level transformations refers to the techniques used to adjust the ______________ of an image.
- A. depth
- B. breadth
- C. width
- D. pixel

1. Linear Transformations

- Linear gray-level transformations are operations where the output pixel value is a linear function of the input pixel value.
- The general form is:
- $s=T(r)=a \cdot r+b$,
- Where:
  - r is the original pixel intensity.
  - s is the transformed pixel intensity.
  - a and b are constants that control the contrast and brightness.

Linear:- Identity and negative transformation

- In identity transformation, each value of the image is directly mapped to each
- other values of the output image. Example: **Contrast stretching**
- S=L-1-r
- Negative transformation is the opposite of identity transformation. Here, each
- value of the input image is subtracted from L-1 and then it is mapped onto the output image
- (e.g) photographic image, medical image

Quiz - 2

- In ____________ , each value of the image is directly mapped to each other values of the output image.
- A). Eigen transformation
- B). Euler transformation
- C). Identity transformation
- D). Negative transformation

Quiz - 2

- In ____________ , each value of the image is directly mapped to each other values of the output image.
- A). Eigen transformation
- B). Euler transformation
- C). Identity transformation
- D). Negative transformation

2. Logarithmic Transformations

- The logarithmic transformation enhances low-intensity values while compressing high-intensity values. This is useful for images with a high dynamic range.
- s

where:

- r is the input pixel intensity (0 to $L-1$).
- s is the output pixel intensity.
- c is a scaling constant, typically chosen as $c = \frac{(L-1)}{\log(1+L-1)}$, where L is the maximum gray level (e.g., 255 for an 8-bit image).
- The "+1" inside the logarithm ensures that even when $r = 0$, the function remains defined.

Effect:

- Enhances details in **darker** regions by expanding their intensity values.
- Compresses **brighter** intensities, reducing contrast in bright areas.
- Useful in applications such as medical imaging, satellite imaging, and low-light photography.
- **Example**:
- If an image has a high range of intensity values (such as an X-ray image where bones appear much brighter), applying a log transformation can make the finer details in the darker areas more visible.

Quiz - 3

- The ____________ transformation enhances low-intensity values while compressing high-intensity values.
- A). Linear
- B). Non-Linear
- C). Algorithmic
- D). Logarithmic

Quiz - 3

- The ___________ transformation enhances low-intensity values while compressing high-intensity values.
- A). Linear
- B). Non-Linear
- C). Algorithmic
- D). Logarithmic

Inverse Log (Exponential) Transformation

- The inverse log transformation performs the opposite of the log transformation. It is used to **enhance brighter** regions while compressing the darker ones.

Formula:

$$s = c \cdot (e^r - 1)$$

where:

- r is the input intensity value.
- s is the transformed intensity.
- c is a scaling constant to normalize the output values.

Effect:

- Enhances **bright** regions by expanding their intensity values.
- Compresses **dark** intensity values, reducing contrast in dark areas.
- Useful for applications where bright details need to be emphasized.
- **Example :**
- If an image has very bright areas with details that need to be enhanced, inverse log transformation can help make those details more prominent.

Power – law(nth power and nth root) Transformations

1. Power-law transformations are widely used to enhance images by adjusting brightness and contrast.
2. These transformations follow a power function that modifies pixel intensity values.
3. Two common forms are **nth power transformation** and **nth root transformation**, which help in enhancing different intensity regions.

a. Power-Law (Nth Power) Transformation

- This transformation is used to **enhance bright regions** and suppress darker ones.

$$s = c \cdot r^{\gamma}$$

where:

- r = input pixel intensity (normalized to [0,1]).
- s = output pixel intensity.
- c = scaling constant (typically chosen as 1).
- γ = power value (greater than 1 for nth power transformation).

a. Power-Law (Nth Power) Transformation

- **Effect:**
- Expands **bright** pixel values (enhancing bright areas).
- Compresses **dark** pixel values (reducing details in darker regions).
- Higher values of γ lead to more extreme brightening.
- **Example Use Case:**
- Used for applications where **bright regions** need more emphasis, such as medical imaging of bright tissues.

Quiz -4

- This transformation is used to **enhance _________ regions** and suppress darker ones.
- A). Dull
- B). Black
- C). Brighter
- D). White

Quiz -4

- This transformation is used to **enhance _________ regions** and suppress darker ones.
- A). Dull
- B). Black
- C). Brighter
- D). White

b. Power-Law (Nth Root) Transformation

- This transformation is used to **enhance dark regions** while compressing bright regions.

$$s = c \cdot r^{\gamma}$$

where $0 < \gamma < 1$, effectively applying an **nth root transformation.**

Effect:

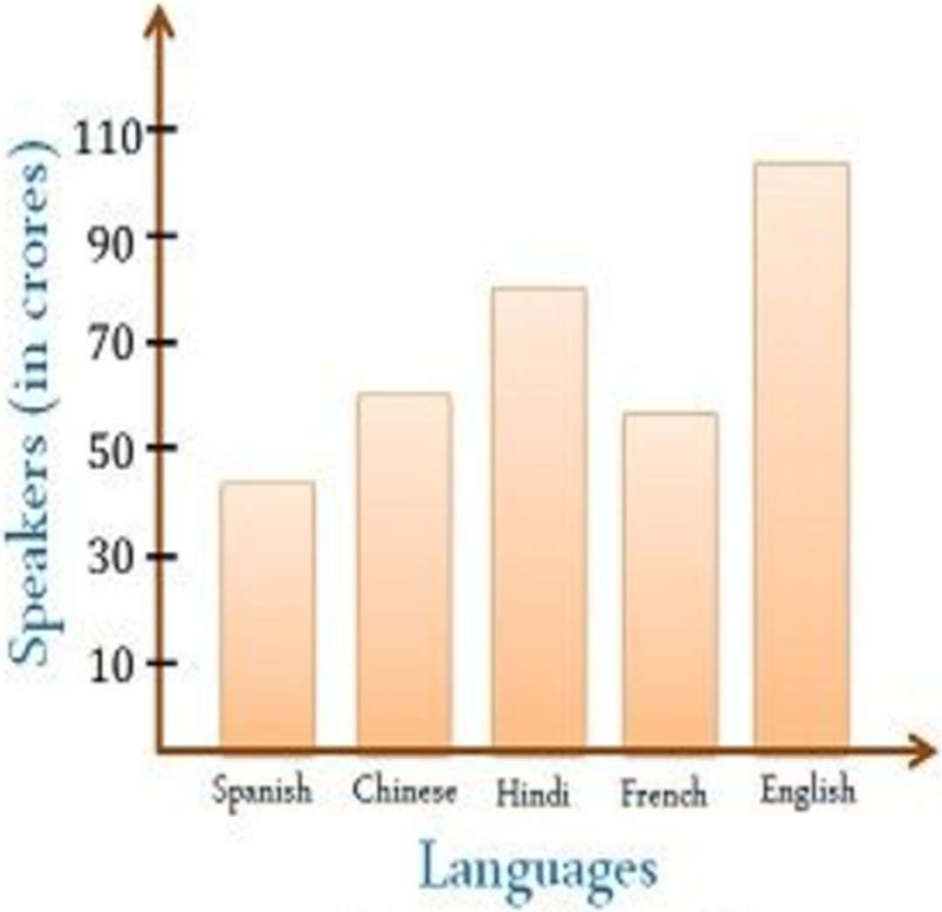
- Expands **dark** pixel values (revealing details in dark areas).
- Compresses **bright** pixel values.
- Useful for low-light images where darker regions need enhancement.

b. Power-Law (Nth Root) Transformation

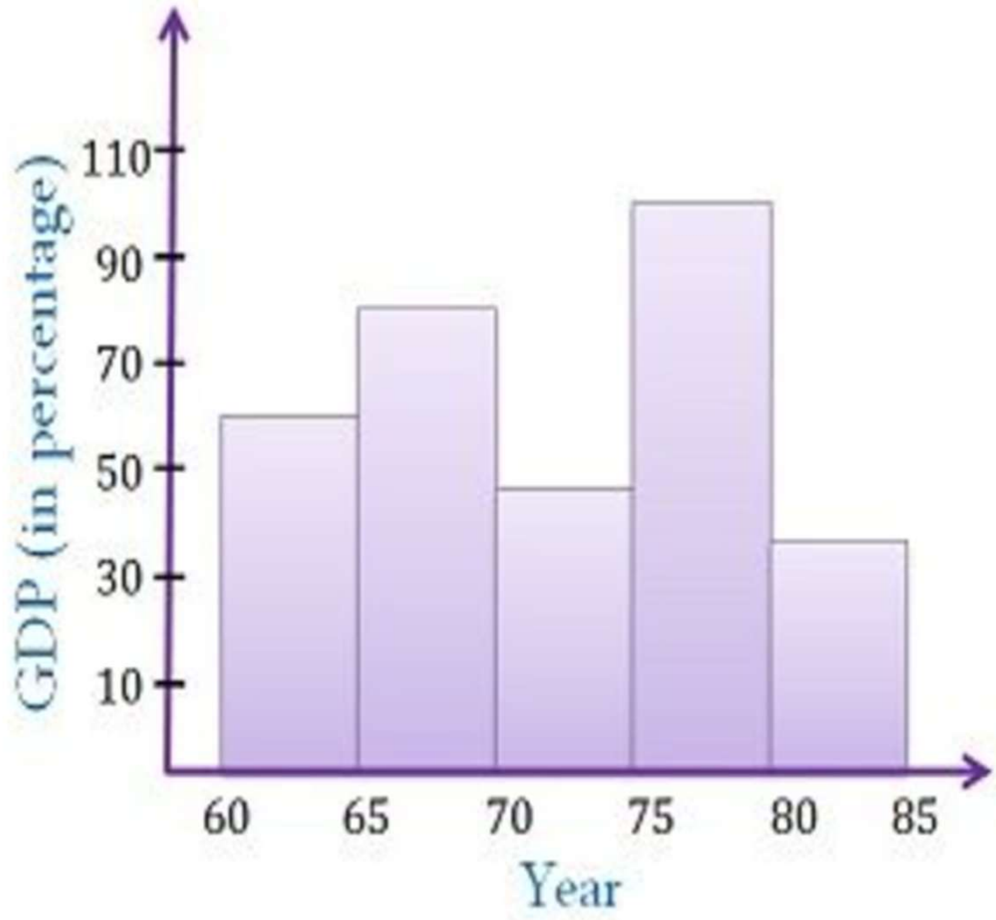
- **Example:**
- Used for **low-light images**, such as night photography or satellite images, to improve visibility of dark areas.

Recap of (5.2.2025) **Gray Level Transformations**

- a) Linear (Negative and Identity)Transformations
- b) Logarithmic(log and inverse-log) Transformations
  - **Log transform**: enhances low-intensity values while compressing high-intensity values
  - **Inverse-Log Transform:** enhance brighter regions while compressing the darker ones
- c) Power – law(nth power and nth root) Transformations
  - **Nth power transform:** enhance bright regions and suppress darker ones.
  - **Nth root transform:** enhance dark regions while compressing bright regions.

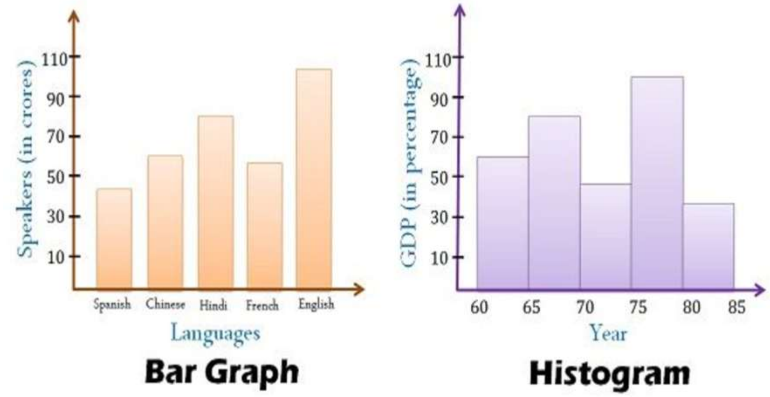


Bar Graph



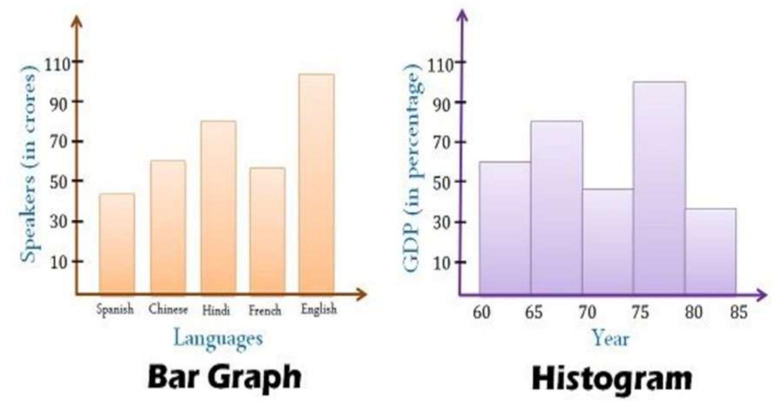
Histogram

Quiz - 1



- What can be interpreted from the diagram? Choose all the correct answer.
  - There are gaps between bars in a bar graph but in the histogram, the bars are adjacent to each other.
  - AAP lost to NDA.
  - Histogram presents numerical data whereas bar graph shows categorical data.
  - DeepSeek is a Chinese artificial intelligence company.

Quiz - 1



- What can be interpreted from the diagram? Choose all the correct answer.
  - There are gaps between bars in a bar graph but in the histogram, the bars are adjacent to each other.
  - AAP lost to NDA.
  - Histogram presents numerical data whereas bar graph shows categorical data.
  - DeepSeek is a Chinese artificial intelligence company.

Histogram Processing

- **What is Histogram Processing?**
  - A technique used to analyze and enhance images by modifying pixel intensity distributions.
  - Helps in contrast enhancement, thresholding and equalization.
  - Essential in applications like medical imaging, satellite imaging and industrial quality inspection.
- **Image Histogram:** A graphical representation of pixel intensity distribution.
  - **X-axis:** Intensity levels (0-255 for an 8-bit image).
  - **Y-axis:** Number of pixels at each intensity level.

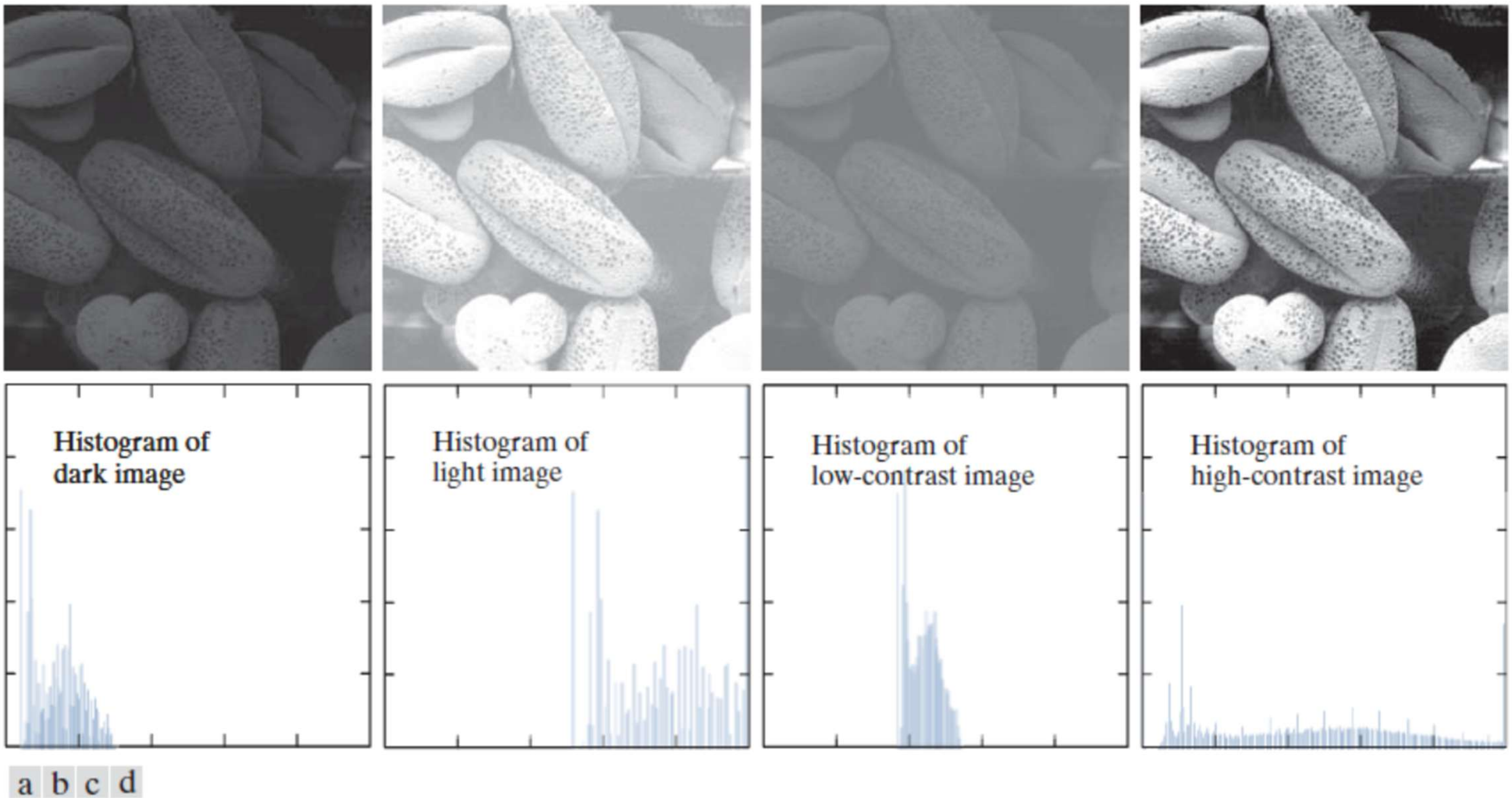


FIGURE 3.16 Four image types and their corresponding histograms. (a) dark; (b) light; (c) low contrast; (d) high contrast. The horizontal axis of the histograms are values of r_k and the vertical axis are values of $p(r_k)$.

Interpretation from the histogram

- Intuitively, it is reasonable to conclude that an image whose pixels tend to
  - occupy the entire range of possible intensity levels and,
  - distributed uniformly, will have an appearance of high contrast and will exhibit a large variety of gray tones.

Types of Histogram Processing (1/2)

1. Histogram Stretching (Normalization)

- Expands the intensity range to improve contrast.
- Formula:
  $s = \frac{(r - r_{\min})}{(r_{\max} - r_{\min})} \times (s_{\max} - s_{\min}) + s_{\min}$

2. Histogram Equalization

- Redistributes pixel intensities to achieve a uniform histogram.
- Uses cumulative distribution function (CDF).
- Improves visibility in low-contrast images

Types of Histogram Processing (2/2)

3. Histogram Specification (Matching)

- Adjusts the histogram to match a desired histogram.
- Useful in standardizing image appearances.

Quiz - 2

- Choose the correct Histogram processing methods.
  - Histogram Stretching
  - Histogram Equalization
  - Histogram Specification
  - Histogram Transformation

Quiz - 2

- Choose the correct Histogram processing methods.
  - Histogram Stretching
  - Histogram Equalization
  - Histogram Specification
  - Histogram Transformation

Applications of Histogram Processing

- **Medical Imaging:** Enhancing X-ray and MRI images.
- **Satellite Imaging**: Improving visibility in aerial and space images.
- **Machine Vision**: Detecting defects in manufacturing.
- **Face Recognition**: Enhancing facial details in low-light conditions.

Histogram Equalization

- Let the variable r denote the intensities of an image to be processed.
- As usual, we assume that r is in the range $[0, L-1]$, with $r = 0$ representing black and $r = L - 1$ representing white.

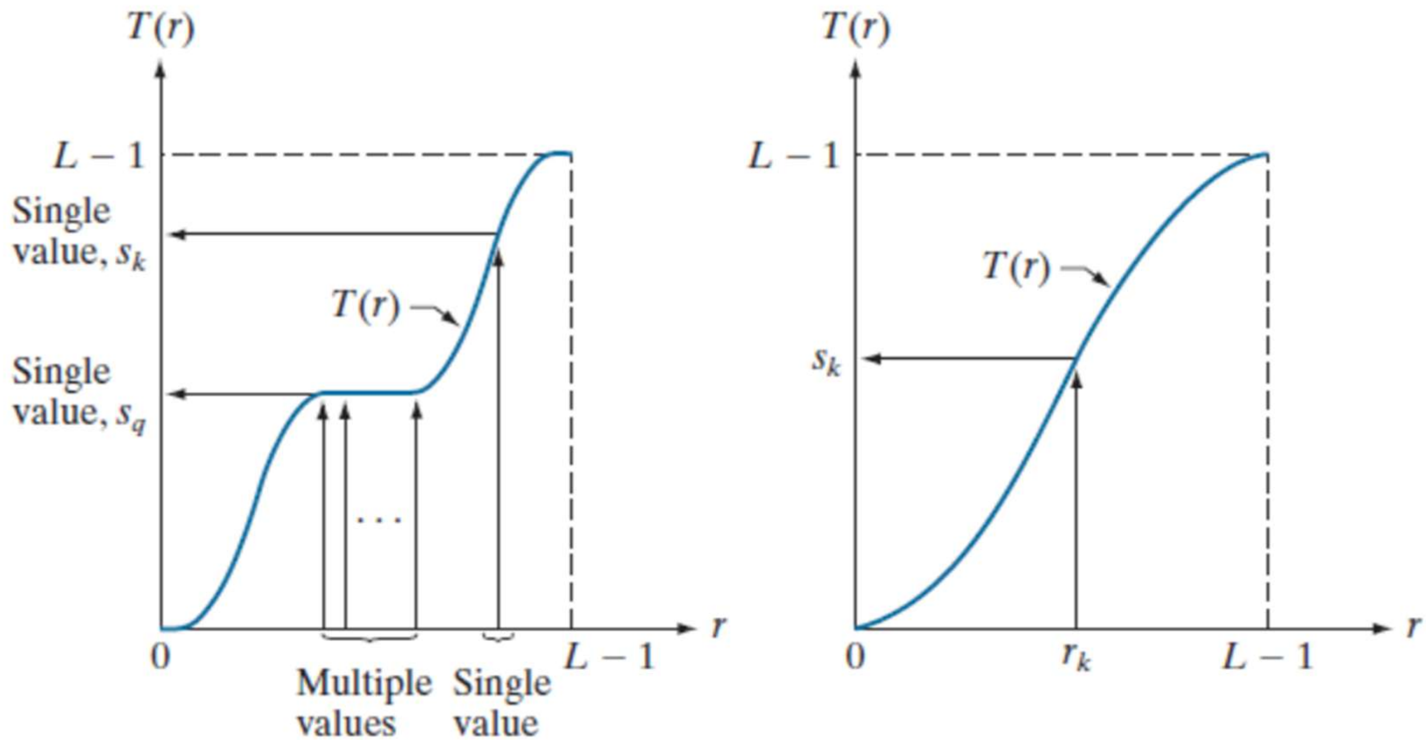


FIGURE 3.17 (a) Monotonic increasing function, showing how multiple values can map to a single value. (b) Strictly monotonic increasing function. This is a one-to-one mapping, both ways.

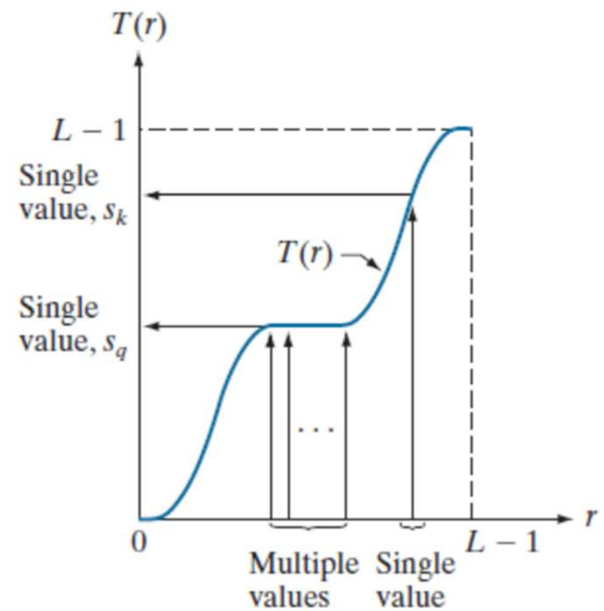
$$s = T(r) \qquad 0 \le r \le L-1$$

that produce an output intensity value, s, for a given intensity value r in the input image. We assume that

(a) $T(r)$ is a monotonic[†] increasing function in the interval $0 \le r \le L-1$; and
(b) $0 \le T(r) \le L-1$ for $0 \le r \le L-1$.

(a) Monotonic Increasing Function

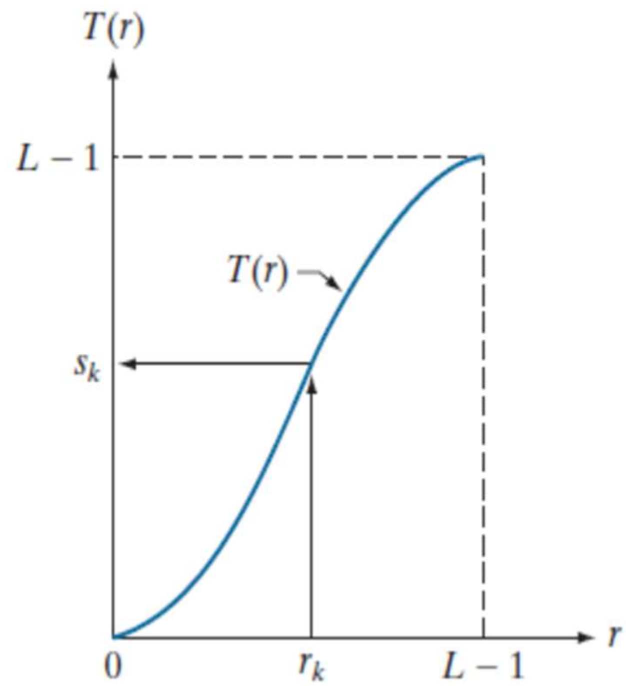
- The left graph shows that multiple values of r (input intensity) can be mapped to a single output intensity ‘s’.
- This means that different pixel intensities in the original image may merge into the same intensity in the processed image, which can lead to loss of details.
- This type of transformation is useful for certain applications but may reduce image contrast.



(b) Strictly Monotonic Increasing Function

- The right graph shows a one-to-one mapping between r and s, ensuring that each input intensity has a unique corresponding output intensity.
- This preserves all details and prevents information loss, making it ideal for contrast enhancement.
- If T(r) is strictly monotonic, the inverse function is **single-valued** (one-to-one).

- These transformations are commonly used in histogram equalization and contrast adjustments in digital image processing.



Quiz - 3

- Which of the following statements correctly differentiates a monotonic increasing function from a strictly monotonic increasing function in histogram processing?
  - **A)** A monotonic increasing function ensures intensity values do not decrease, while a strictly monotonic increasing function ensures distinct input values map to distinct output values.
    B) A strictly monotonic increasing function allows intensity values to remain constant, whereas a monotonic increasing function does not.
    C) A monotonic increasing function always increases, while a strictly monotonic increasing function can decrease in certain cases.
    D) Both functions behave the same way and have no distinction in histogram processing.

Quiz - 3

- Which of the following statements correctly differentiates a monotonic increasing function from a strictly monotonic increasing function in histogram processing?
  - **A)** A monotonic increasing function ensures intensity values do not decrease, while a strictly monotonic increasing function ensures distinct input values map to distinct output values.
    B) A strictly monotonic increasing function allows intensity values to remain constant, whereas a monotonic increasing function does not.
    C) A monotonic increasing function always increases, while a strictly monotonic increasing function can decrease in certain cases.
    D) Both functions behave the same way and have no distinction in histogram processing.

Practical Challenges in Discrete Image Processing

- Pixel intensity values are stored as integers, leading to rounding errors.
- This may cause non-strict monotonicity, making inverse transformations less precise.

Probability Density Function (PDF) Transformation

- The intensity of an image can be modeled as a **random variable** in $[0, L-1]$.
- Using probability theory, the PDF of transformed intensity values $p_s(s)$ is related to the original PDF $p_r(r)$ through the derivative of $T(r)$:

$$p_s(s) = p_r(r) \left| \frac{dr}{ds} \right|$$

- This equation is crucial for **histogram equalization** and other intensity transformations.

FUNDAMENTALS OF SPATIAL FILTERING

- The name *filter* is borrowed from frequency domain processing components of an image. For example, a filter that passes low frequencies is called a *lowpass filter*.
- The net effect produced by a lowpass filter is to smooth an image by blurring it. We can accomplish similar smoothing directly on the image itself by using *spatial filters*.
- Spatial filtering modifies an image by replacing the value of each pixel by a function of the values of the pixel and its neighbors. If the operation performed on the image pixels is linear, then the filter is called a *linear spatial filter*. Otherwise, the filter is a *nonlinear spatial filter*.

FUNDAMENTALS OF SPATIAL FILTERING

- Spatial filtering is a fundamental technique in image processing used to enhance or suppress specific features in an image. The two major categories of spatial filters are:

1. **Smoothing Spatial Filters** – Used to reduce noise and blur an image.
2. **Sharpening Spatial Filters** – Used to enhance edges and fine details in an image.

Quiz - 1

- **Smoothing Spatial Filters** is used to reduce noise and blur an image.
- A). Frequency filters
- B). Rough filters
- C). Smoothing Spatial filters
- D). Sharpening Spatial filters

Quiz - 1

- **Smoothing Spatial Filters** is used to reduce noise and blur an image.
- A). Frequency filters
- B). Rough filters
- C). Smoothing Spatial filters
- D). Sharpening Spatial filters

1. Smoothing Spatial Filters

- Smoothing filters are designed to reduce noise and smooth variations in an image by averaging pixel values within a neighborhood. These filters are useful for:
  - Reducing noise
  - Removing small details
  - Blurring an image
- **Types of Smoothing Filters:**
  - (i) Averaging (Mean) Filter
  - (ii) Gaussian Filter
  - (iii) Median Filter

Other terms used to refer to a spatial filter kernel are *mask*, *template*, and *window*. We use the term *filter kernel* or simply *kernel*.

Separable Filter Kernels

- A **separable filter kernel** is a filter that can be broken down into two **1D filters**—one applied along the **rows** and the other along the **columns**—instead of applying a single 2D filter. This significantly reduces computational complexity.

Example: 2D Gaussian Filter

A **Gaussian blur kernel** is a common example of a separable filter. Consider this 2D Gaussian filter:

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

This can be separated into two **1D filters:**

$$k_r = \frac{1}{4}[1, 2, 1]$$

$$k_c = \frac{1}{4}[1, 2, 1]^T$$

1. Smoothing Spatial Filters

Averaging (Mean) Filter

- This filter replaces each pixel with the average value of its neighboring pixels.
- It is implemented using a kernel (filter mask) where all values are equal.

  Example of a 3×3 Averaging Kernel:

$$K = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- This results in a blurring effect, reducing sharp transitions.

1. Smoothing Spatial Filters

(ii) Gaussian Filter

- A weighted averaging filter that gives more importance to the center pixel.
- The weights follow a Gaussian distribution, making the blurring effect smoother.

  Example of a 3×3 Gaussian Kernel:

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

- Gaussian smoothing is widely used in preprocessing steps for edge detection.

1. Smoothing Spatial Filters

(iii) Median Filter

- Instead of averaging, it replaces each pixel with the median value of its neighborhood.
- This is highly effective in removing "salt and pepper" noise.

Example:

Original Pixel Neighborhood: [12, 5, 8, 200, 7, 10, 6, 9, 15]

Median Value: 9 (Replaces center pixel)

2. Sharpening Spatial Filters

- Sharpening filters highlight transitions (edges) in an image by enhancing high-frequency components. These filters are useful for:
  - Enhancing edges
  - Highlighting fine details
  - Increasing image contrast
- **Types of Sharpening Filters:**
  - (i) Laplacian Filter
  - (ii) High Boost Filtering
  - (iii) Unsharp Masking

Quiz - 2

- Sharpening filters highlight transitions (edges) in an image by enhancing high-frequency components. These filters are useful for:
- A). Enhancing edges
- B). Highlighting fine details
- C). Increasing image contrast
- D). Blur the image

Quiz - 2

- Sharpening filters highlight transitions (edges) in an image by enhancing high-frequency components. These filters are useful for:
- A). Enhancing edges
- B). Highlighting fine details
- C). Increasing image contrast
- D). Blur the image

Frequency Domain

- Images can be represented as a combination of **sinusoidal waves** (low/high frequencies).
  - **Low frequencies**: Smooth areas (e.g., walls, skies).
  - **High frequencies**: Edges, noise, textures.
- **Key Tool**: **Fourier Transform** (DFT/FFT) converts spatial-domain images to frequency-domain spectra.

Quiz - 1

- Why Convert from Spatial to Frequency Domain?
- A. Separation of Image Components (Low vs. High Frequency)
- B. Efficient Filtering (Convolution Theorem)
- C. Image Compression (Energy Compaction)
- D. Edge Detection and Feature Extraction

Quiz - 1

- Why Convert from Spatial to Frequency Domain?
- A. Separation of Image Components (Low vs. High Frequency)
- B. Efficient Filtering (Convolution Theorem)
- C. Image Compression (Energy Compaction)
- D. Edge Detection and Feature Extraction

Smoothing (Lowpass Filters)

- Filtering in the frequency domain consists of modifying the Fourier transform of an image, then computing the inverse transform to obtain the spatial domain representation of the processed result.
- Thus, given (a padded) digital image, $f(x, y)$, of size $P * Q$ pixels, the basic filtering equation in which we are interested has the form:

$$g(x,y) = \text{Real}\left\{\mathfrak{I}^{-1}\left[H(u,v)F(u,v)\right]\right\}$$

  - where F^{-1} is the IDFT, F(u,v) is the DFT of the input image, $f(x, y)$,
  - H(u,v) is a *filter transfer function* (which we often call just a *filter* or *filter function*), and
  - $g(x, y)$ is the *filtered* (*output*) *image*.
- Functions F, H, and g are arrays of size $P \cdot Q$, the same as the padded input image.
- The product H(u,v)F(u,v) is formed using elementwise multiplication.
- The filter transfer function modifies the transform of the input image to yield the processed output, $g(x, y)$.

Smoothing (Lowpass Filters)

- A function *H*(u,v) that attenuates high frequencies while passing low frequencies (called a *lowpass filter*, as noted before) would blur an image,
- while a filter with the opposite property (called a *highpass filter*) would enhance sharp detail, but cause a reduction in contrast in the image.

$$g(x,y) = \text{Real}\left\{\Im^{-1}\left[H(u,v)F(u,v)\right]\right\}$$

Quiz - 2

- ___________ *filter*, would blur an image; _________ *filter* would enhance sharp detail of an image.
- A. lowpass, lopass
- B. highpass, lowpass
- C. lowpass, highpass
- D. highpass, higpass

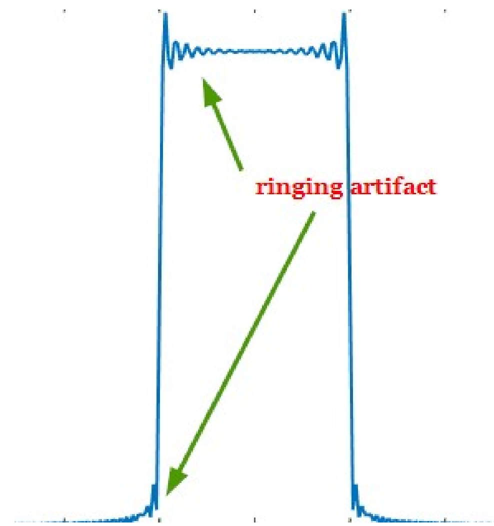
Quiz - 2

- ___________ *filter*, would blur an image; _________ *filter* would enhance sharp detail of an image.
- A. lowpass, lopass
- B. highpass, lowpass
- C. lowpass, highpass
- D. highpass, higpass

Types of Low-Pass Filters

- 1. Ideal Low-Pass Filter (ILPF)
- 2. Gaussian Low-Pass Filter (GLPF)
- 3. Butterworth Low-Pass Filter (BLPF)

Ringing artifacts, also known as the **Gibbs phenomenon**, refer to oscillations that appear near sharp edges or discontinuities when a signal or image is reconstructed using a truncated or limited-frequency representation



1. Ideal Low-Pass Filter (ILPF)

- **Concept:** Directly removes all frequencies beyond a cutoff D_0.
- **Mathematical Formula:**

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

where $D0$ is a positive constant, and D(u,v) is the distance between a point (u,v) in the frequency domain and the center of the $P \cdot Q$ frequency rectangle; that is,

$$D(u,v) = \left[(u - P/2)^2 + (v - Q/2)^2 \right]^{1/2}$$

- **Effect:** Sudden transitions lead to ringing artifacts (Gibbs phenomenon).

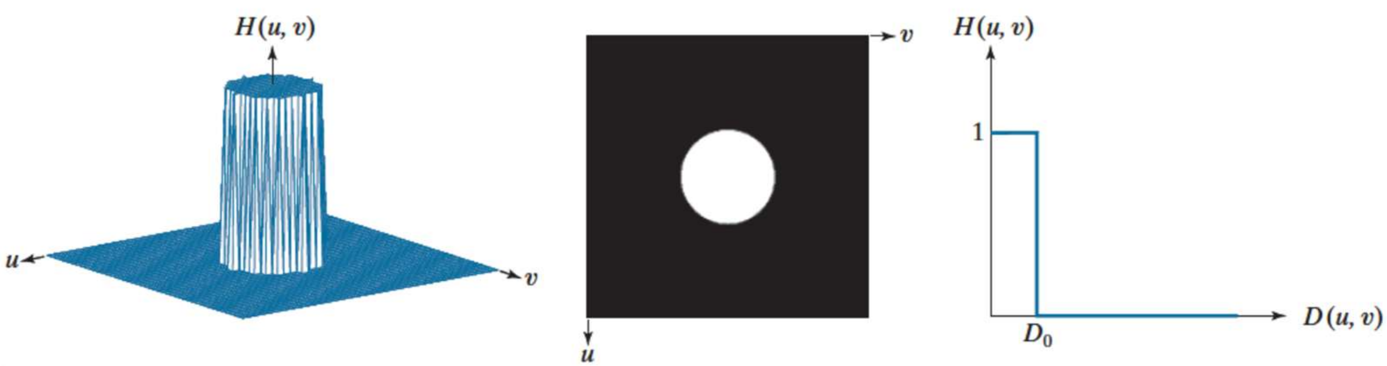


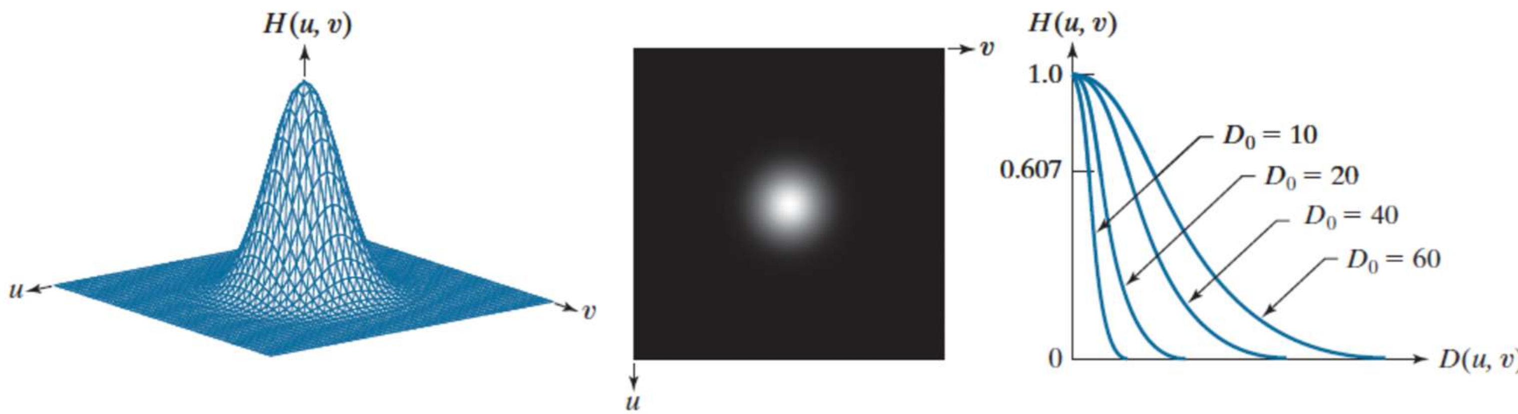
FIGURE 4.39 (a) Perspective plot of an ideal lowpass-filter transfer function. (b) Function displayed as an image. (c) Radial cross section.

2. Gaussian Low-Pass Filter (GLPF)

- **Concept:** Smoothly decreases frequencies beyond D_0 using a Gaussian function.
- **Formula:**

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

- **Effect:** Produces smooth blurring without ringing artifacts.



a b c

FIGURE 4.43 (a) Perspective plot of a GLPF transfer function. (b) Function displayed as an image. (c) Radial cross sections for various values of D_0.

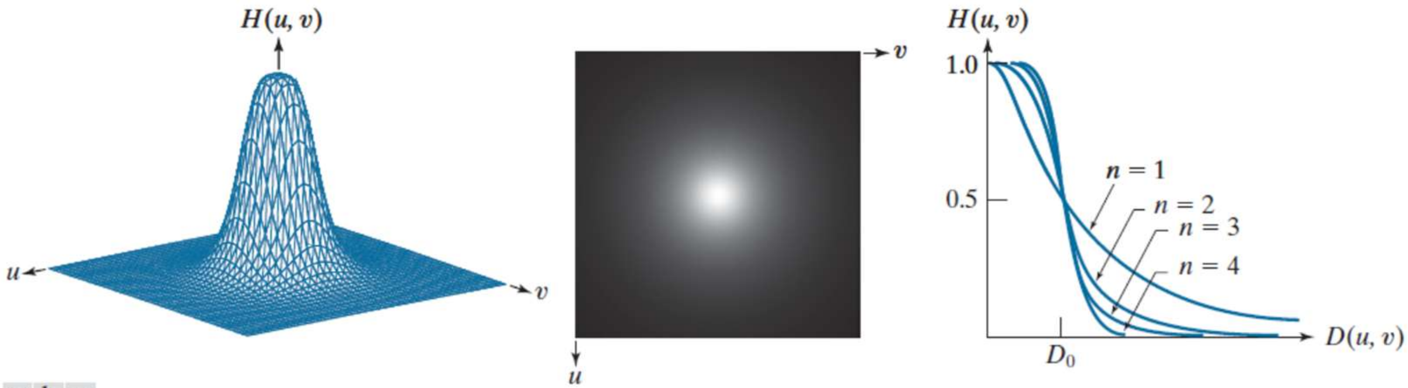
3. Butterworth Low-Pass Filter (BLPF)

- **Concept:** Provides a gradual transition between preserved and filtered frequencies.
- **Formula:**

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

- n controls the sharpness of the transition; higher n results in an effect closer to an ideal filter.

Comparing the cross section plots in Figs. 4.39, 4.43, and 4.45, we see that the BLPF function can be controlled to approach the characteristics of the ILPF using higher values of n, and the GLPF for lower values of n, while providing a smooth transition in from low to high frequencies.



a b c

FIGURE 4.45 (a) Perspective plot of a Butterworth lowpass-filter transfer function. (b) Function displayed as an image. (c) Radial cross sections of BLPFs of orders 1 through 4.

Frequency Domain - Sharpening frequency domain filters

- Sharpening in the **frequency domain** is achieved by emphasizing high-frequency components while suppressing low-frequency components.
- This enhances edges, fine details, and textures in an image.
  - **Low frequencies** represent smooth variations (backgrounds, gradual intensity changes).
  - **High frequencies** represent rapid changes (edges, fine textures, noise).
- Sharpening in the frequency domain focuses on boosting high-frequency components, making edges and details more pronounced.

Frequency Domain - Sharpening frequency domain filters

- Subtracting a lowpass filter transfer function from 1 yields the corresponding highpass filter transfer function in the frequency domain:

$$H_{\text{HP}}(u,v) = 1 - H_{\text{LP}}(u,v) \tag{4-118}$$

where $H_{\text{LP}}(u,v)$ is the transfer function of a lowpass filter. Thus, it follows from Eq. (4-111) that an ideal highpass filter (IHPF) transfer function is given by

$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases} \tag{4-119}$$

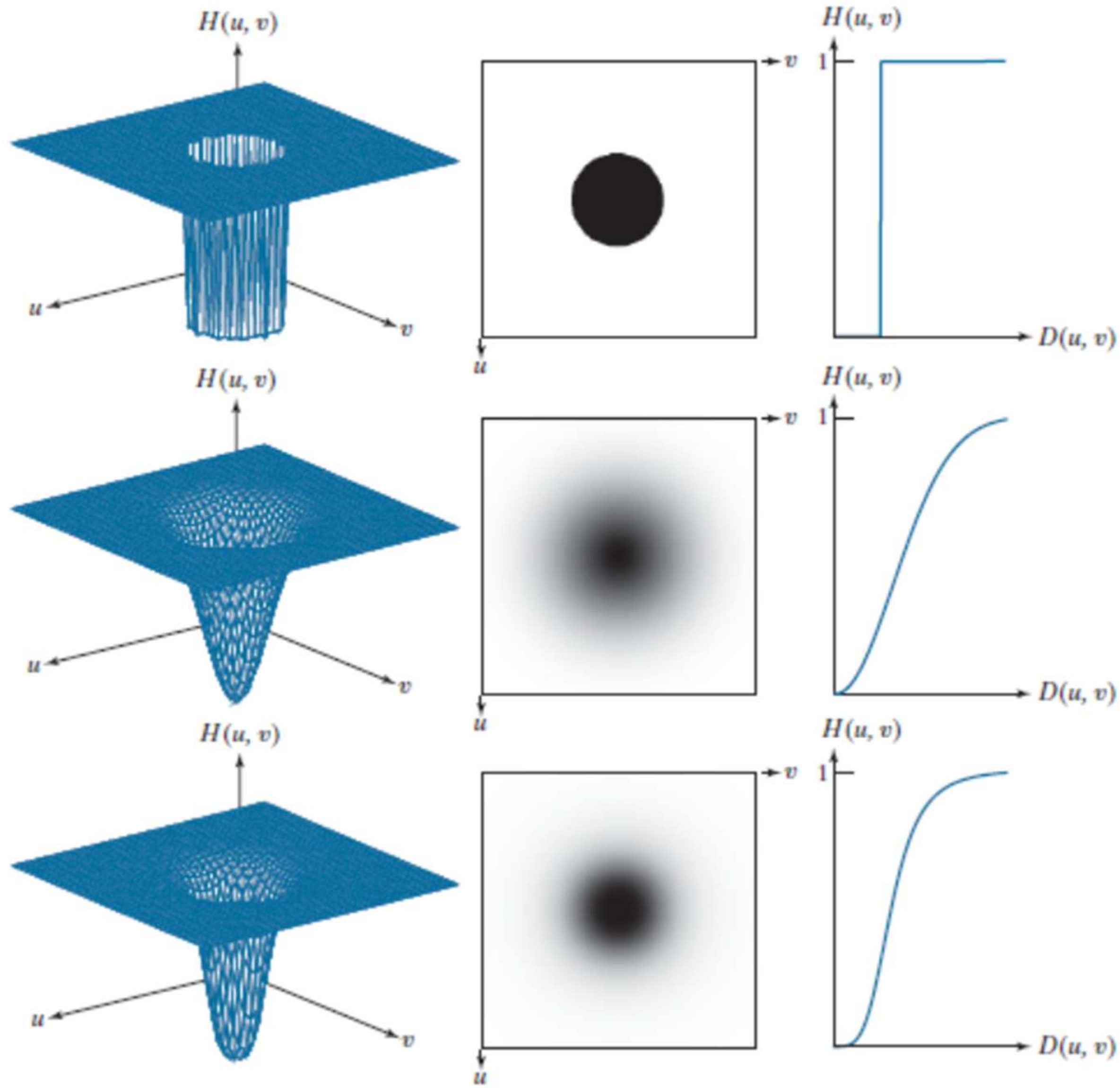


FIGURE 4.51 Top row: Perspective plot, image, and, radial cross section of an IHPF transfer function. Middle and bottom rows: The same sequence for GHPF and BHPF transfer functions. (The thin image borders were added for clarity. They are not part of the data.)

Ideal High-Pass Filtering (IHPF)

- This filter completely removes low-frequency components (smooth regions) and retains only high-frequency components (edges and textures).
- It acts as a binary filter:
  - Frequencies below a cutoff D_0 are completely removed.
  - Frequencies above D_0 are completely retained.

Mathematical Formula

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

Where:

- $D(u,v)$ is the distance from the center of the frequency domain.
- D_0 is the cutoff frequency.

Pros and Cons

✅ Strong edge enhancement

❌ Causes ringing artifacts (due to sudden cut-off of frequencies)

1. Ideal High-Pass Filtering (IHPF)

- This filter completely removes low-frequency components (smooth regions) and retains only high-frequency components (edges and textures).
- It acts as a binary filter:
  - Frequencies below a cutoff D_0 are completely removed.
  - Frequencies above D_0 are completely retained.

Mathematical Formula

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

Where:

- $D(u,v)$ is the distance from the center of the frequency domain.
- D_0 is the cutoff frequency.

Pros and Cons

✅ Strong edge enhancement

❌ Causes ringing artifacts (due to sudden cut-off of frequencies)

2. Butterworth High-Pass Filtering (BHPF)

- A smooth version of the ideal filter, reducing abrupt frequency transitions.
- The filter function gradually increases from 0 to 1 around D_0, avoiding sharp cutoffs.

$$H(u,v) = \frac{1}{1 + \left(\frac{D_0}{D(u,v)}\right)^{2n}}$$

Where:

- n is the order of the filter (higher n = sharper transition).
- D_0 is the cutoff frequency.

Pros and Cons

✅ Controls edge sharpness without artifacts

✅ More stable than the ideal high-pass filter

❌ Requires tuning of parameters n and D_0

3. Gaussian High-Pass Filtering (GHPF)

- Uses a Gaussian function to smoothly attenuate low frequencies while preserving high frequencies.
- The transition between low and high frequencies is smoothest, avoiding artifacts like ringing.

Mathematical Formula

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

Where:

- D_0 is the cutoff frequency.

Pros and Cons

✅ No ringing artifacts
✅ Smooth filtering
❌ Slightly lower edge sharpness compared to Butterworth or Ideal

4. Laplacian High-Pass Filtering (LHPF)

- A sharpening filter that enhances high frequencies directly using the Laplacian operator in the frequency domain.
- It amplifies edges more aggressively than other filters.

Mathematical Formula

$$H(u, v) = -4\pi^2(u^2 + v^2)$$

Where:

- (u, v) are frequency domain coordinates.

Pros and Cons

✅ Strongest edge enhancement
✅ Captures fine details
❌ Amplifies noise significantly

Filters

Filter Type	Edge Enhancement	Noise Sensitivity	Artifacts (Ringing, Overshoot)	Smoothness
Ideal HPF	High	High	High (Ringing)	Sharp Cutoff
Butterworth HPF	Moderate to High	Moderate	Low to Moderate	Smooth
Gaussian HPF	Moderate	Low	None	Very Smooth
Laplacian HPF	Very High	Very High	Strong (Noise Amplified)	Sharp

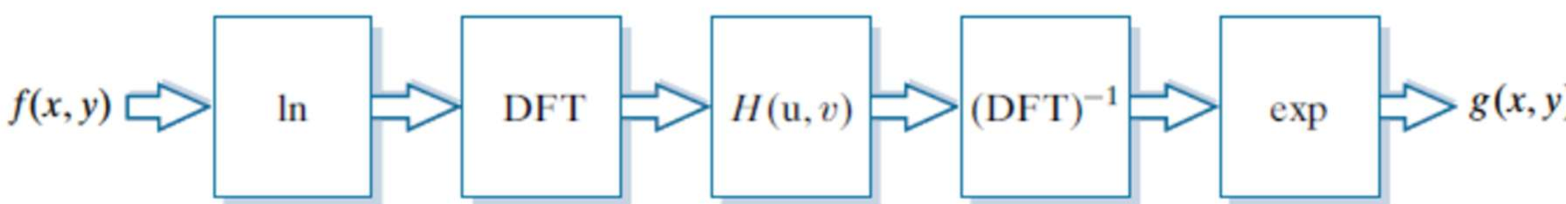
Summary

- Ideal HPF is the sharpest but introduces artifacts.
- Butterworth HPF is smoother and controllable.
- Gaussian HPF provides the smoothest sharpening without artifacts.
- Laplacian HPF offers extreme sharpening but amplifies noise

HOMOMORPHIC FILTERING

- Homomorphic filtering is a technique used in **image processing** to correct **non-uniform illumination** and enhance contrast by manipulating the **illumination and reflectance** components of an image.
- It operates in the **frequency domain** and uses a **high-pass filter** to attenuate low frequencies (illumination) while amplifying high frequencies (reflectance).
- An image $f(x, y)$ can be expressed as the product of its illumination, $i(x, y)$, and reflectance, $r(x, y)$, components:

FIGURE 4.58
Summary of steps in homomorphic filtering.



Quiz - 1

- _____________ is a technique in digital image processing used for image enhancement, particularly in improving contrast and correcting non-uniform illumination.
- A). Homomorphic filtering
- B). Isomorphic filtering
- C). Mesomorphic filtering
- D). Harmonic filtering

Quiz - 1

- ______________ is a technique in digital image processing used for image enhancement, particularly in improving contrast and correcting non-uniform illumination.
- A). Homomorphic filtering
- B). Isomorphic filtering
- C). Mesomorphic filtering
- D). Harmonic filtering

HOMOMORPHIC FILTERING

- **1. Image Model Representation**
- An image can be modeled as:
  - $f(x, y) = i(x, y)r(x, y)$
    - where:
    - i(x,y) → Illumination (low-frequency components)
    - r(x,y) → Reflectance (high-frequency components)
- To apply filtering, we take the **logarithm** to convert multiplication into addition:
  - ln f(x,y) = ln i(x,y) + ln r(x,y)
- **2. Transform to Frequency Domain**
- Applying the **Fourier Transform (FT)**:
  - F(u,v)=I(u,v)+R(u,v)
    - where F(u,v), I(u,v) and R(u,v) are the FT of ln f(x,y), ln i(x,y) and ln r(x,y) respectively

HOMOMORPHIC FILTERING

- 3. Frequency-Domain Filtering

We apply a **high-pass filter** $H(u,v)$ to suppress low frequencies (illumination) and enhance high frequencies (reflectance):

$$G(u,v) = H(u,v) \cdot F(u,v)$$

A common choice is the **Gaussian high-pass filter**:

$$H(u,v) = \gamma_H + (\gamma_L - \gamma_H)e^{-c(D(u,v)^2/D_0^2)}$$

where:

- $D(u,v) \rightarrow$ Distance from the frequency center
- $D_0 \rightarrow$ Cutoff frequency
- $\gamma_H, \gamma_L \rightarrow$ Parameters controlling enhancement

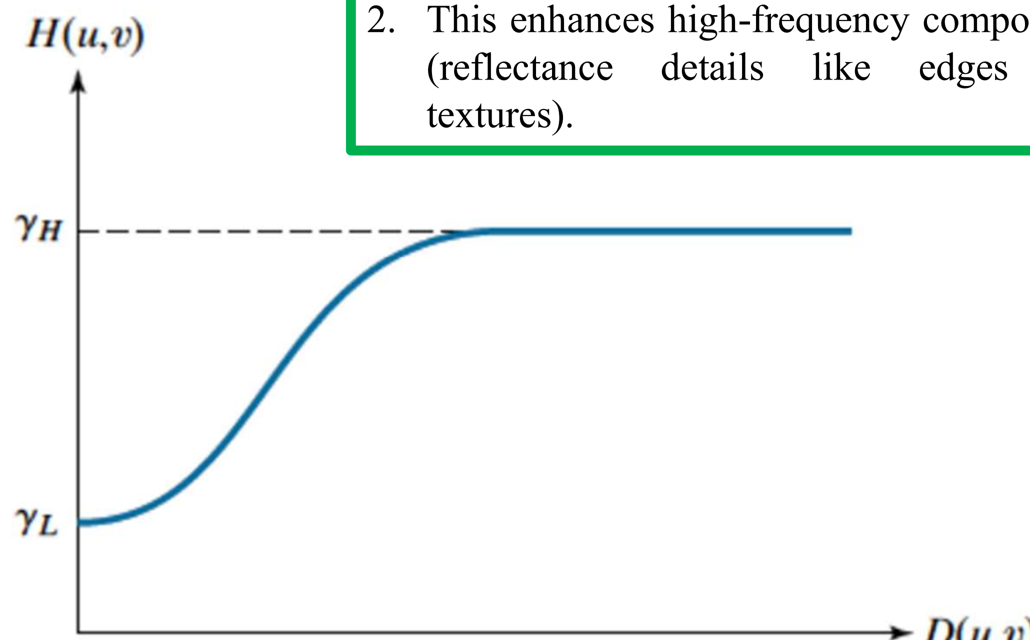
c is a constant controlling the steepness of the transition between high and low frequencies.

HOMOMORPHIC FILTERING

- 3. Frequency-Domain Filtering

$$H(u,v) = \gamma_H + (\gamma_L - \gamma_H)e^{-c(D(u,v)^2/D_0^2)}$$

FIGURE 4.59 Radial cross section of a homomorphic filter transfer function..



Lower Frequencies (D(u,v) ≈ 0):

1. The filter response starts at γ_L, which is a low gain factor.
2. This means low-frequency components (illumination variations) are attenuated, reducing uneven lighting effects.

Higher Frequencies (D(u,v) → large values):

1. The filter response saturates at γ_H, a higher gain factor.
2. This enhances high-frequency components (reflectance details like edges and textures).

HOMOMORPHIC FILTERING

4. Inverse Transform and Exponentiation

After filtering, we take the **Inverse Fourier Transform (IFT)**:

$$g(x, y) = \mathcal{F}^{-1}[G(u, v)]$$

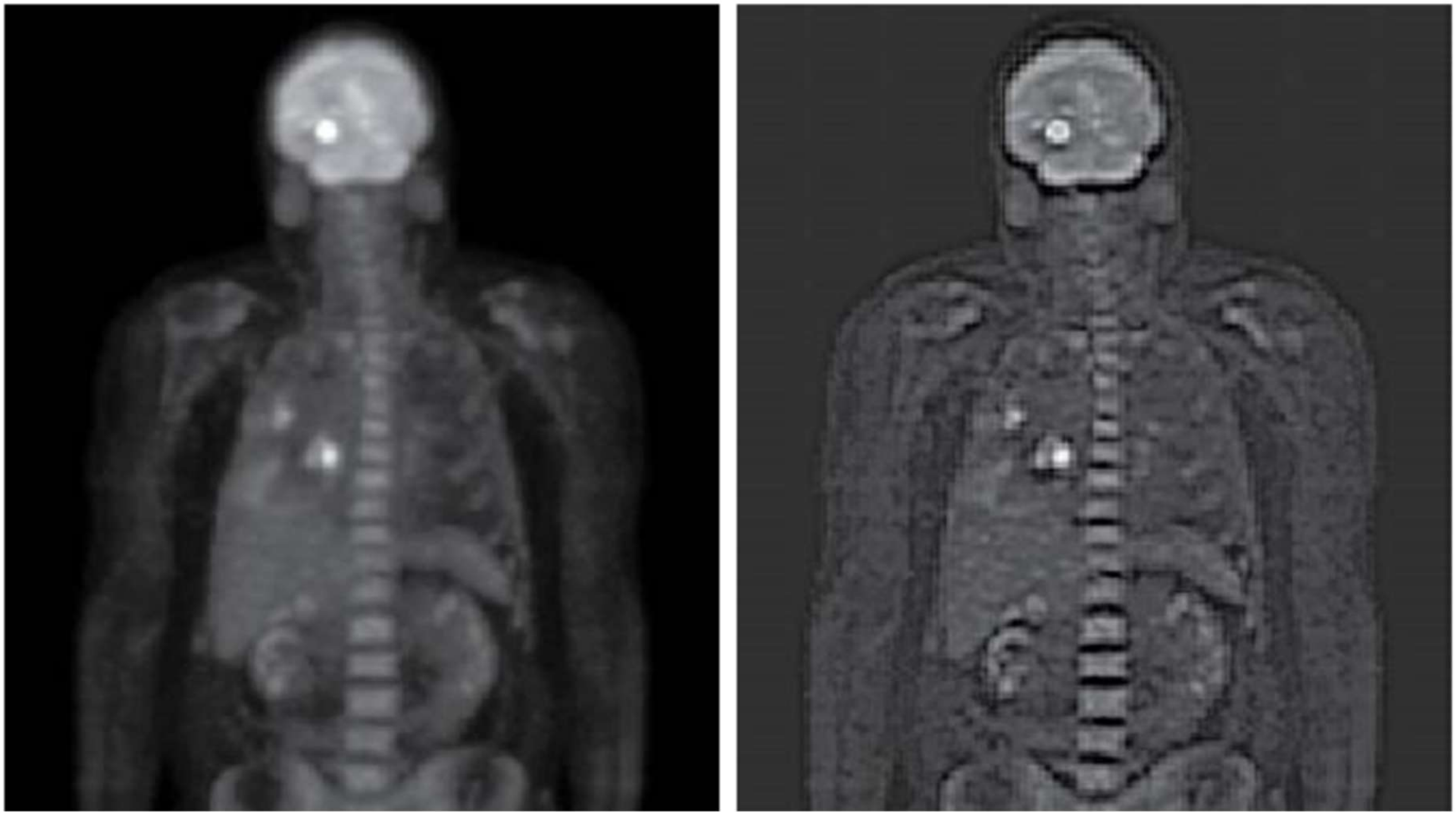
Then, apply **exponentiation**:

$$f'(x, y) = e^{g(x,y)}$$

This restores the image with enhanced details and improved contrast.

APPLICATIONS OF HOMOMORPHIC FILTERING

1. **Medical Imaging** → Enhancing X-ray, MRI, and CT scan images.
2. **Document Enhancement** → Improving old, degraded, or unevenly illuminated text images.
3. **Satellite Image Processing** → Enhancing details in remote sensing images.
4. **Face Recognition** → Improving image contrast in facial images.



a b

FIGURE 4.60
(a) Full body PET scan. (b) Image enhanced using homomorphic filtering. (Original image courtesy of Dr. Michael E. Casey, CTI Pet Systems.)

Position Emission Tomography