

Digital Image Processing and Pattern Recognition

E1528



Lecture 3

Intensity Transformations and Spatial Filtering

INSTRUCTOR

DR / AYMAN SOLIMAN

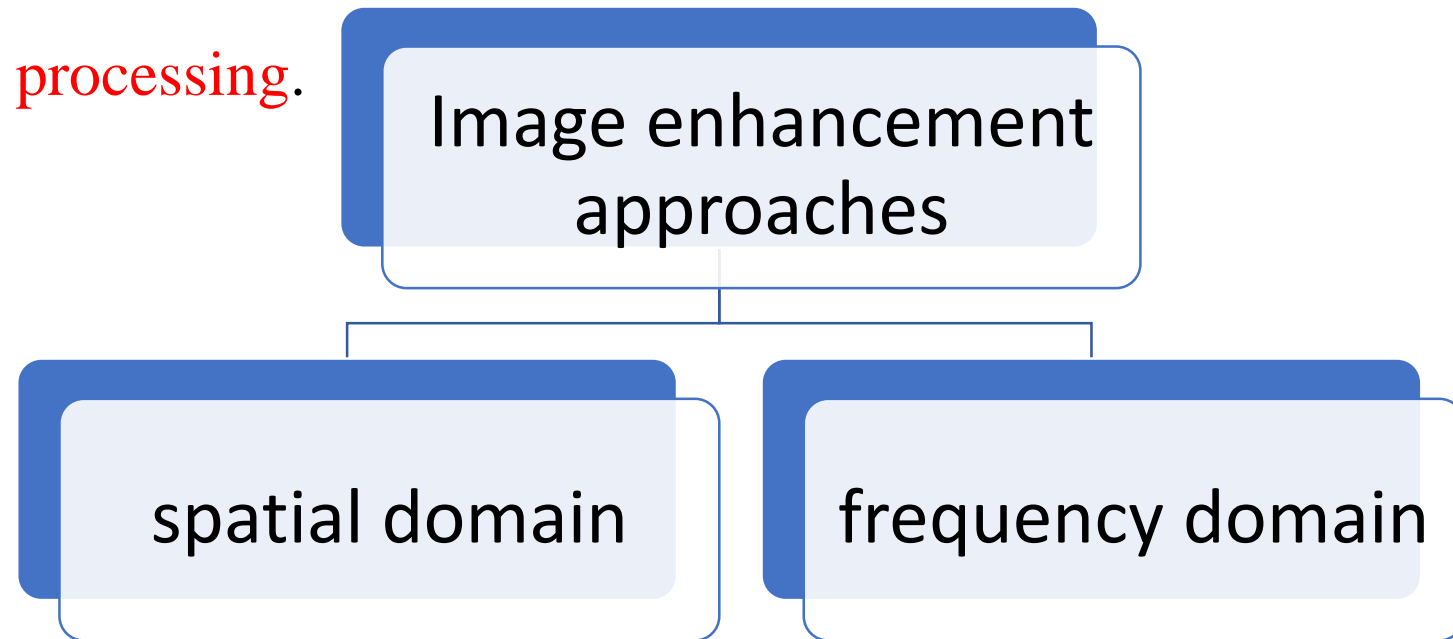
➤ Contents

- Some Basic Gray Level Transformations
- spatial domain Vs. Frequency domain
- Some Basic Gray Level Transformations
- Linear - Image Negatives
- Linear - Image identity
- Log Transformations
- Power-Law Transformations
- Histogram Processing
- Basics of Spatial Filtering
- Smoothing Linear Filters
- Order-Statistics Filters
- Sharpening Spatial Filters



➤ Some Basic Gray Level Transformations

- The principal **objective of enhancement** is to process an image so that the **result is more suitable** than the **original image** for a specific application.
- **Image enhancement** is one of the most interesting and visually appealing areas of **image processing**.



➤ **spatial domain Vs. Frequency domain**

- The term **spatial domain** refers to the image plane itself, and approaches in this category are based on direct manipulation of pixels in an image.
- **Frequency domain** processing techniques are based on modifying the Fourier transform of an image.
- There is **no general theory** of image enhancement.

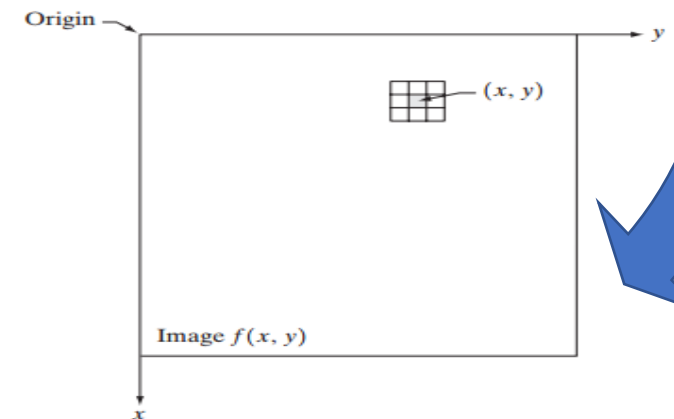
➤ Background

- Spatial domain processes will be denoted by the expression

$$g(x, y) = T[f(x, y)]$$

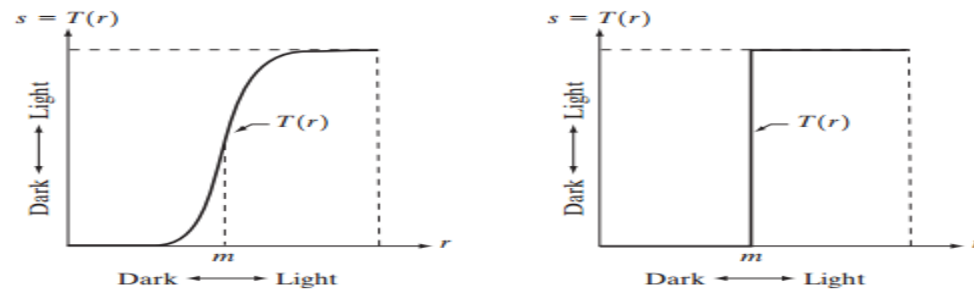
Where $f(x, y)$ is the **input image**, $g(x, y)$ is the **processed image**, and T is an **operator** on f , defined over some neighborhood of (x, y) .

In addition, T can operate on a set of input images, such as performing the pixel-by-pixel sum of K images for noise reduction

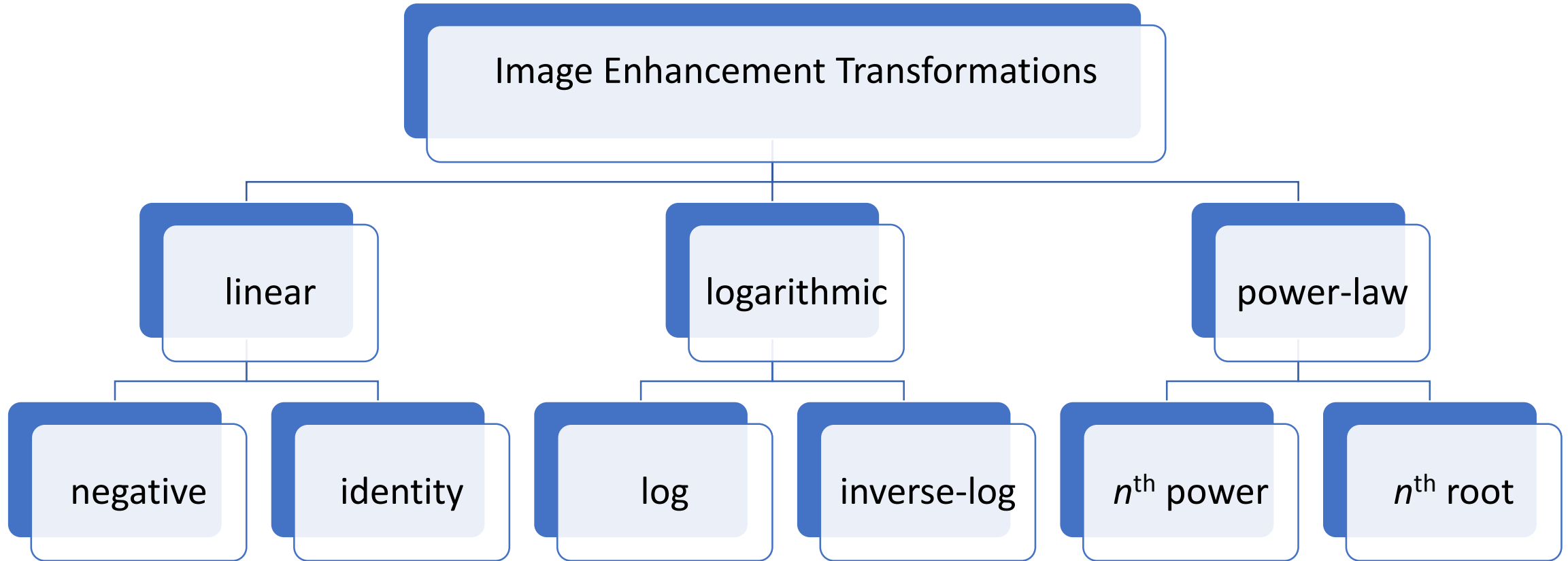


➤ Background

- The simplest form of T is when the neighborhood is of **size 1*1** (that is, a **single pixel**). In this case, g depends only on the value of f at (x, y) , and T becomes a **gray-level** (also called an **intensity** or **mapping**) transformation function of the form
$$s = T(r)$$
- where, for simplicity in notation, r and s are variables denoting, respectively, the gray level of $f(x, y)$ and $g(x, y)$ at any point (x, y) .



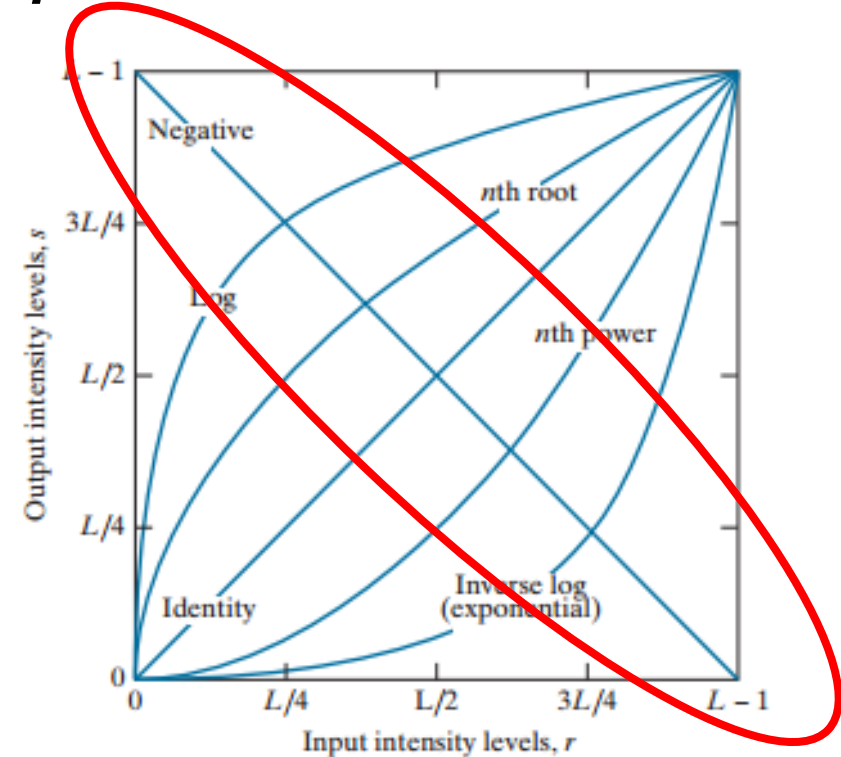
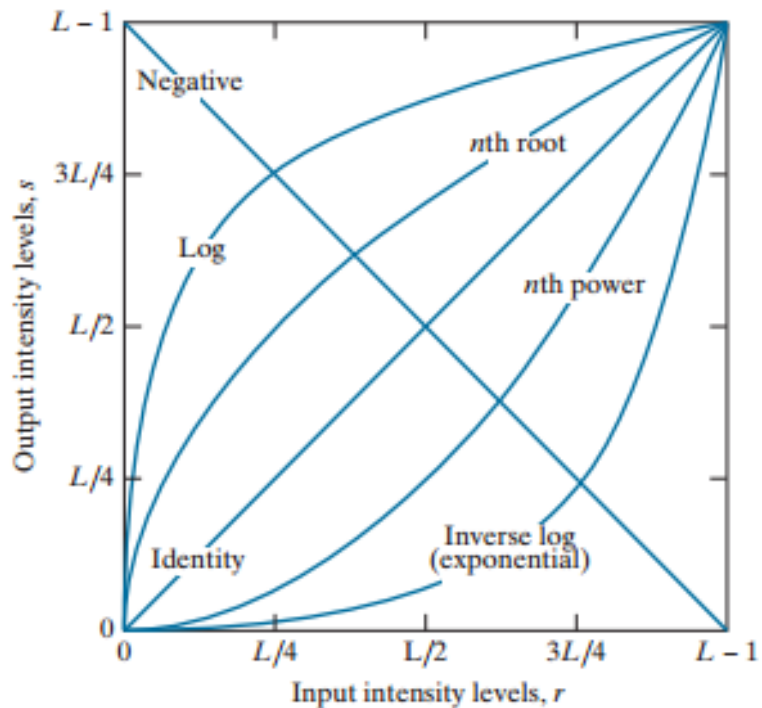
➤ Some Basic Gray Level Transformations



➤ Linear - Image Negatives

- The **negative** of an image with gray levels in the range $[0, L-1]$ is obtained by using the negative transformation shown below, which is given by the expression

$$s = L - 1 - r$$

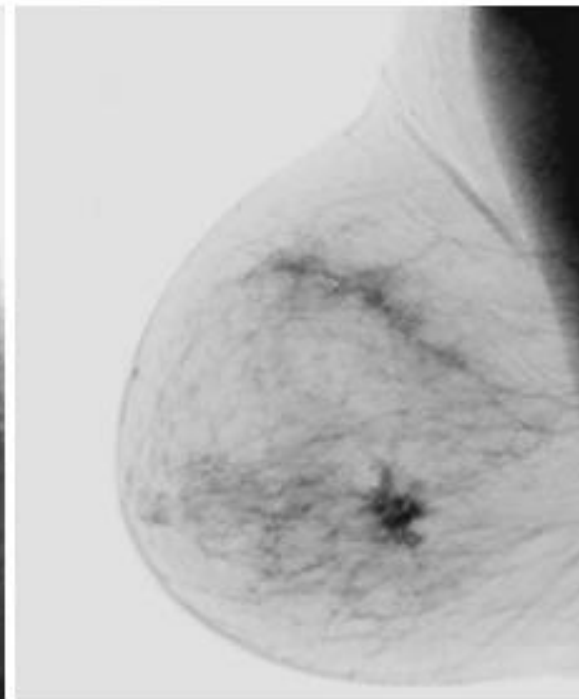
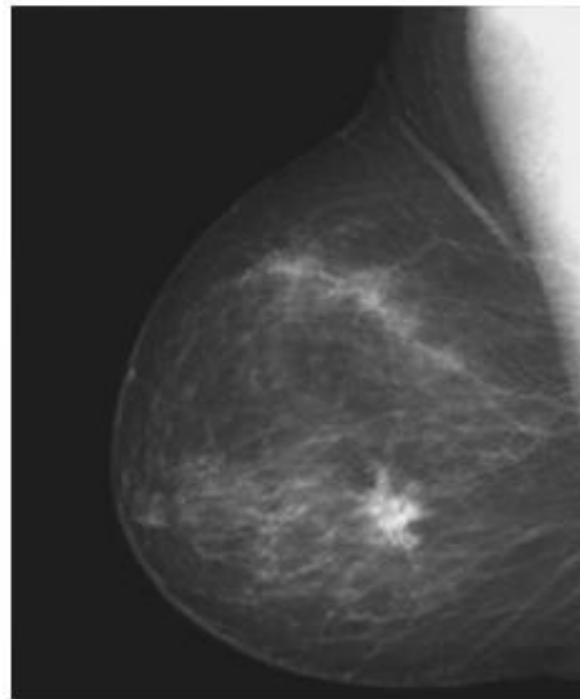


➤ Linear - Image Negatives

- The **original** image is a digital mammogram showing **a small lesion**.

Although the visual content is the same in both images,

- Note how much **easier** it is to **analyze** the breast tissue in the **negative** image in this case.

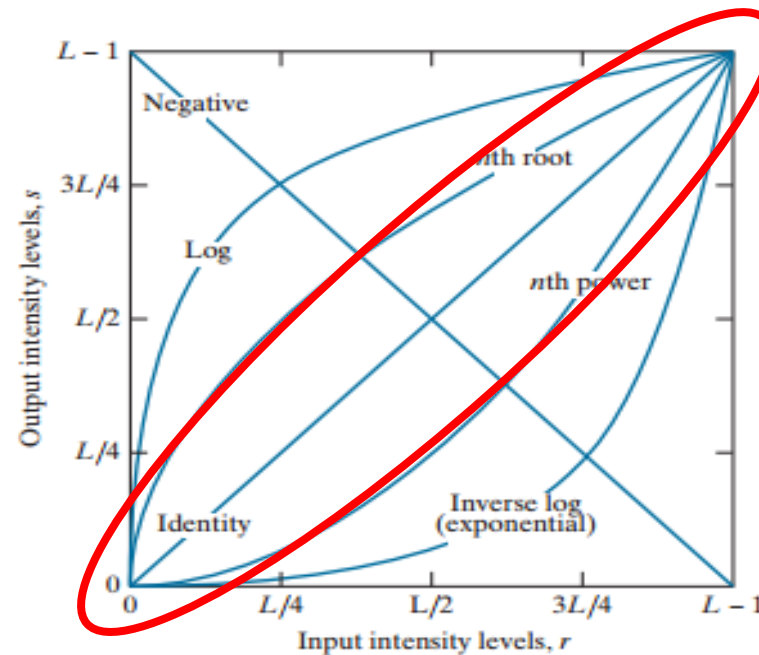


a b

(a) Original digital mammogram.
(b) Negative image obtained using the negative transformation in

➤ Linear - Image identity

- The identity function is the **trivial case** in which output intensities are **identical** to input intensities.
- It is included in the graph only for **completeness**.



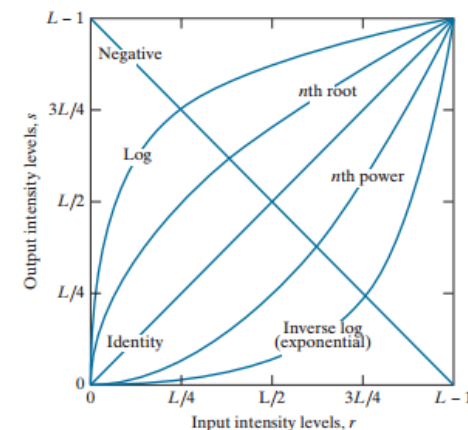
➤ Log Transformations

- The general form of the **log transformation** is

$$s = c \log(1 + r)$$

where c is a constant, and it is assumed that $r \geq 0$.

- The shape of the log curve in Fig. below shows that this transformation maps a narrow range of low gray-level values in the input image into a wider range of output levels.



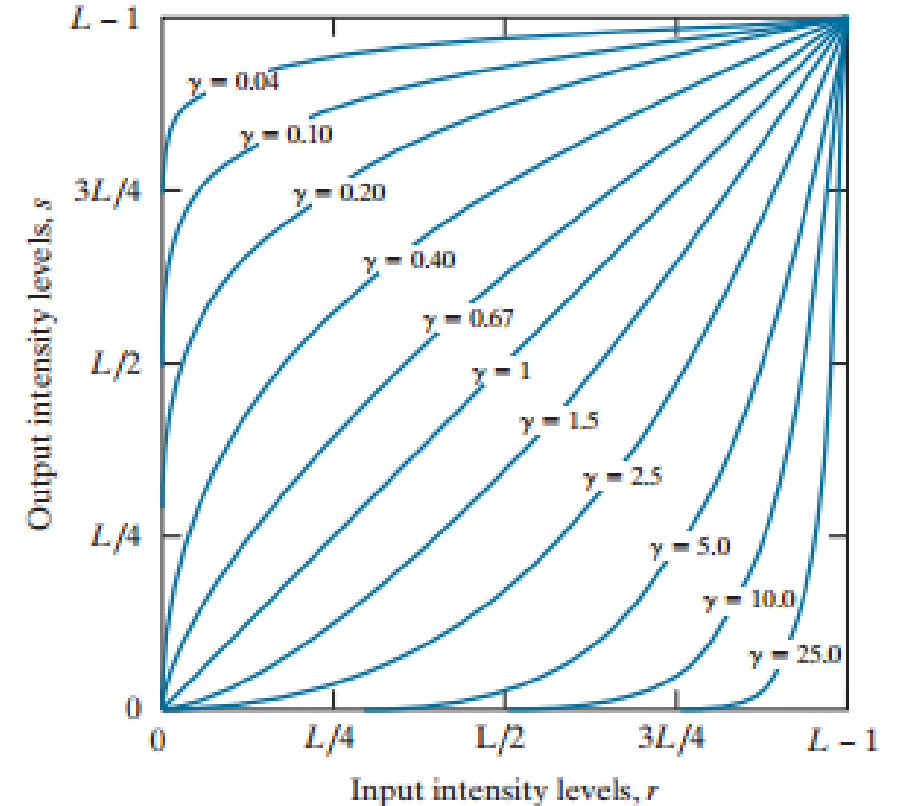
➤ Power-Law Transformations

- Power-law transformations have the basic form

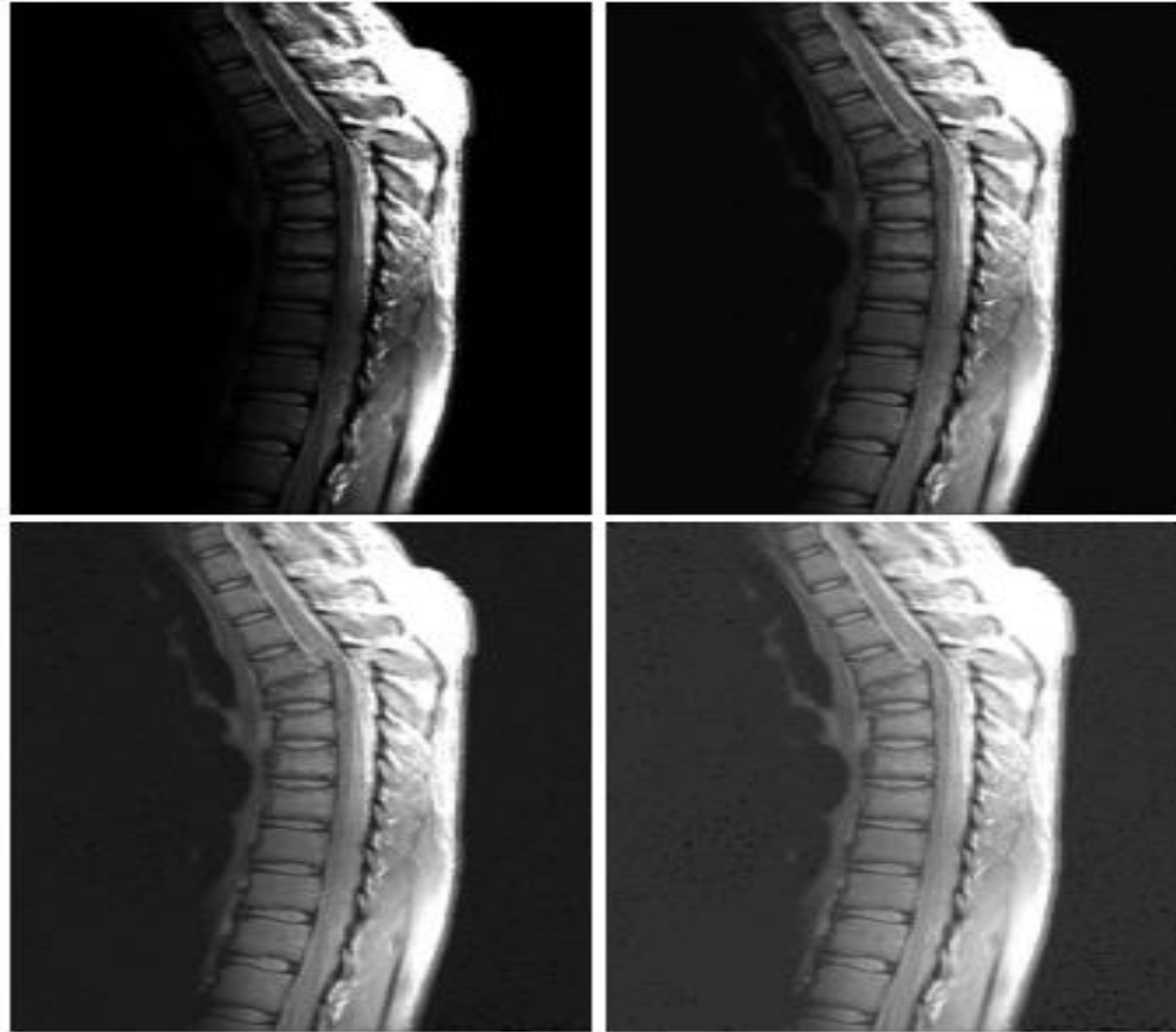
$$s = cr^\gamma$$

where c and γ are positive constants.

Sometimes this Eq. is written as $s = c(r + \epsilon)^\gamma$ to account for an offset (that is, a measurable output when the input is zero).



➤ Power-Law Transformations Example



a b
c d

(a) Magnetic resonance (MR) image of a fractured human spine.
(b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 0.6, 0.4,$ and $0.3,$ respectively. (Original image for this example courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

➤ Power-Law Transformations Example

a b
c d

(a) Aerial image.
(b)–(d) Results of
applying the
transformation in
Eq. (3.2-3) with
 $c = 1$ and
 $\gamma = 3.0, 4.0,$ and
 $5.0,$ respectively.
(Original image
for this example
courtesy of
NASA.)



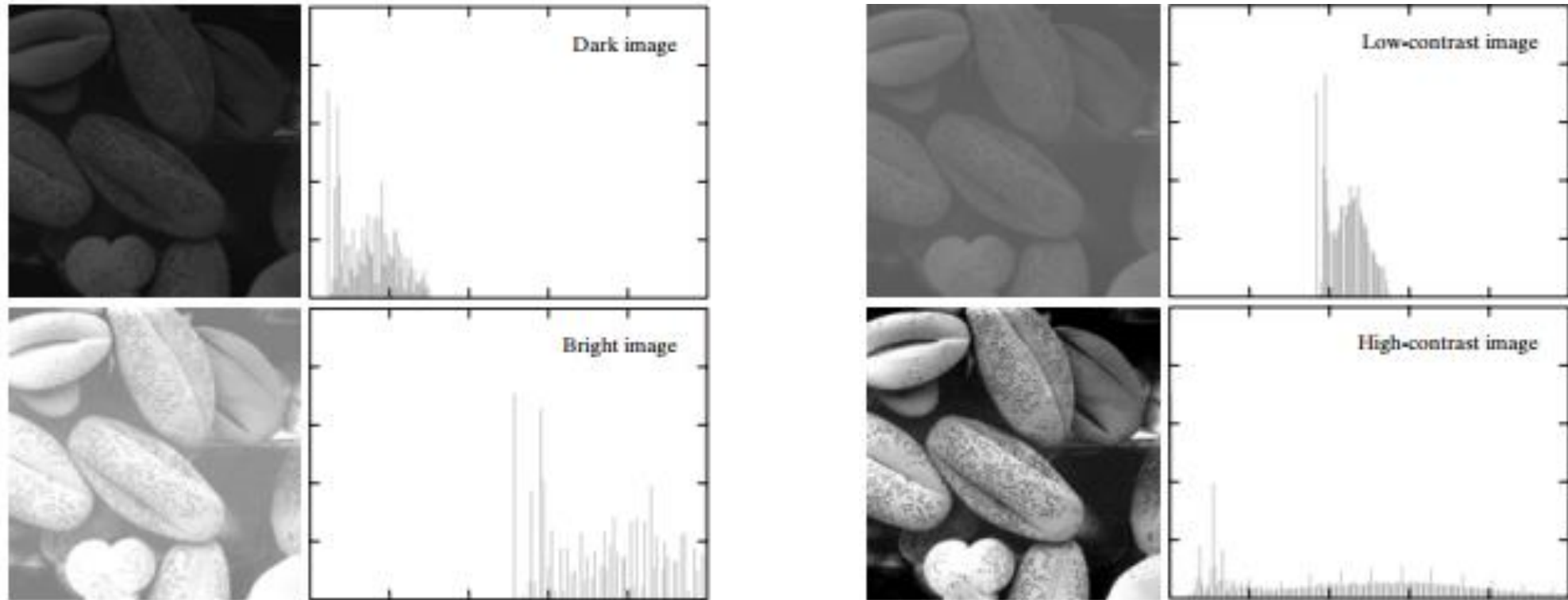
➤ Histogram Processing

- The histogram of a digital image with gray levels in the range $[0, L-1]$ is a **discrete function $h(r_k)=n_k$** , where r_k is the k_{th} **gray level** and n_k is the **number of pixels** in the image having gray level r_k .
- It is common practice to normalize a histogram by **dividing each of its values by the total number of pixels in the image, denoted by n** .
- Thus, a normalized histogram is given by $p(r_k)=n_k/n$, for $k=0, 1, \dots, L-1$.
- Loosely speaking, $p(r_k)$ **gives an estimate of the probability of occurrence of gray level r_k** . Note that the **sum** of all components of a normalized histogram is **equal to 1**.

➤ Histogram Processing

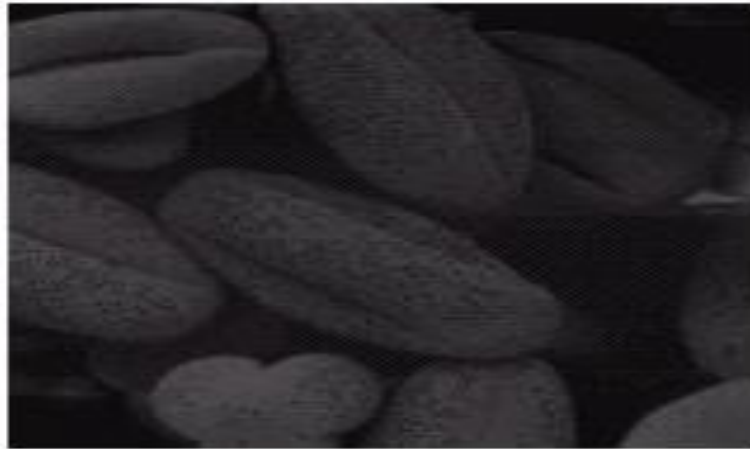
- Histograms are the **basis** for numerous **spatial domain processing techniques**.
- Histogram **manipulation** can be used **effectively** for **image enhancement**, In addition to providing useful image statistics.
- Histograms are **simple** to **calculate in software** and lend themselves to **economic hardware implementations**, thus making them a **popular tool** for **real-time** image processing.

➤ Histogram Processing Example



- Four basic image types: **dark**, **light**, **low contrast**, **high contrast**, and their corresponding histograms.

➤ 1. Histogram Equalization

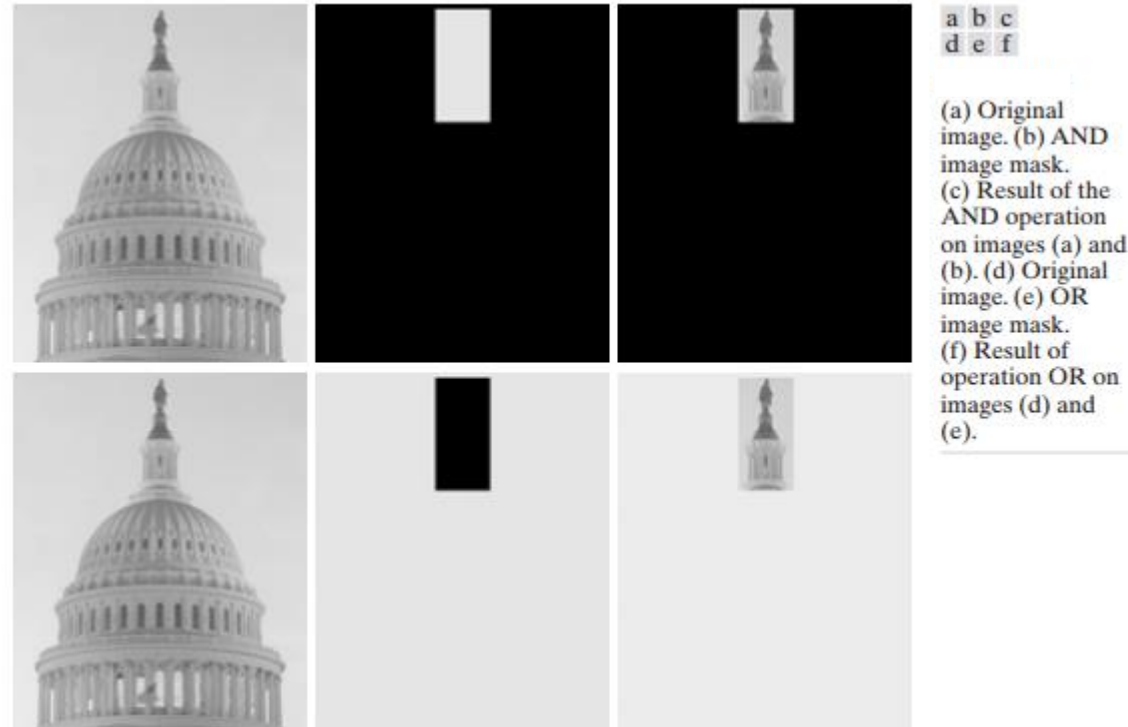


(a) Images

(b) Results of histogram equalization. (c) Corresponding histograms.

➤ Enhancement Using Arithmetic/Logic Operations

- **Arithmetic/logic** operations involving images are performed on a **pixel-by-pixel** basis between **two or more** images (this **excludes** the logic operation **NOT**, which is performed on a **single** image).

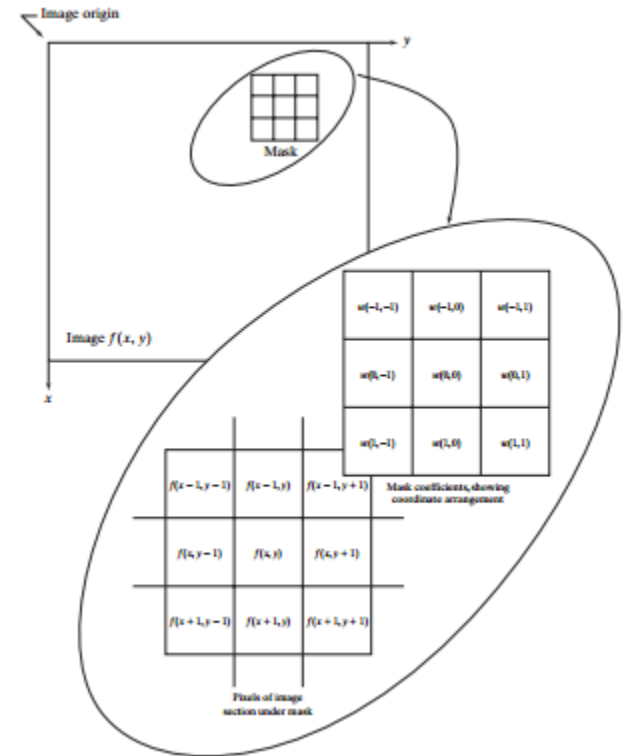


➤ Basics of Spatial Filtering

- As mentioned, some neighborhood operations work with the values of the image pixels in the **neighborhood** and the **corresponding values of a sub-image** that has the same dimensions as the neighborhood.
- The sub-image is called a **filter, mask, kernel, template, or window**, with the first **three** terms being the most prevalent terminology.
- The values in a filter sub-image are referred to as **coefficients**, rather than **pixels**.

➤ Mechanics of spatial filtering

- The process consists **simply of moving** the filter **mask** from **point to point** in an **image**. At each point (x, y) , the response of the filter at that point is **calculated** using a **predefined relationship**.
- For **linear spatial** filtering, the response is given by a **sum of products** of the filter **coefficients** and the corresponding **image pixels** in the area spanned by the filter mask.



➤ **Smoothing Spatial Filters**

- Smoothing filters are used for **blurring** and for **noise reduction**.
- **Blurring** is used in **preprocessing** steps, such as **removal of small details** from an image prior to (large) object extraction, and bridging of small gaps in lines or curves.
- **Noise reduction** can be **accomplished** by blurring with a **linear** filter and by **nonlinear** filtering.

➤ 1. Smoothing Linear Filters

- The output (response) of a smoothing, linear spatial filter is **simply** the **average** of the pixels contained in the neighborhood of the filter mask.
- These filters sometimes are called **averaging filters**.

$$\frac{1}{9} \times$$

1	1	1
1	1	1
1	1	1

$$\frac{1}{16} \times$$

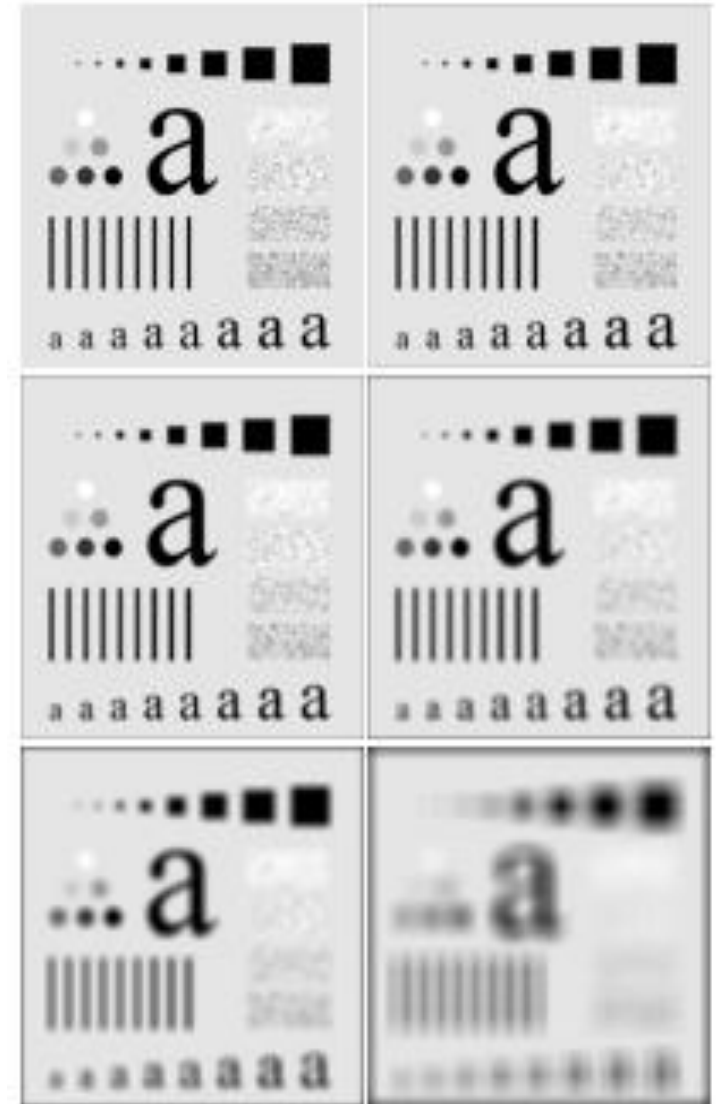
1	2	1
2	4	2
1	2	1

$$R = \frac{1}{\Sigma} \sum_{i=1}^n z_i$$

➤ 1. Smoothing Linear Filters Example

(a) Original image, of size 500*500 pixels.

(b) → (f) Results of smoothing with square averaging filter masks of sizes $n=3, 5, 9, 15,$ and $35,$ respectively.

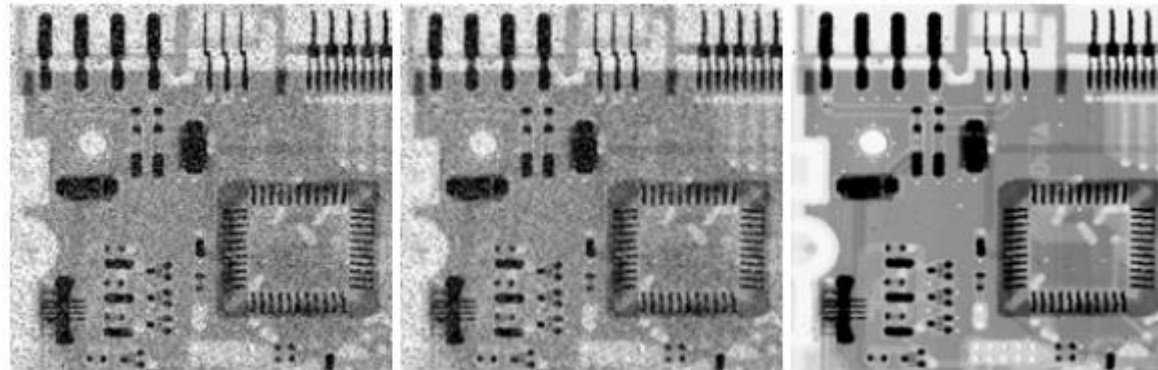


➤ 2. Order-Statistics Filters

- Order-statistics filters are **nonlinear** spatial filters whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result.
- The best-known example in this category is the **median filter**, which, as its name implies, replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel.
- **Median filters** are quite **popular** because, for certain types of **random noise**, they provide excellent **noise-reduction** capabilities, with considerably **less blurring than linear smoothing** filters of similar size.

➤ 2. Order-Statistics Filters - Median filter

- In order to **perform** median filtering at a **point** in an image, **we first sort** the values of the pixel in question and its neighbors, determine their **median**, and assign this value to that pixel.
- **For example**, suppose that a 3*3 neighborhood has values (10, 20, 20, 20, 15, 20, 20, 25, 100). These values are sorted as (10, 15, 20, 20, **20**, 20, 20, 25, 100), which results in a median of 20.



➤ Example:-

- (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3*3 averaging mask. (c) Noise reduction with a 3*3 median filter.

➤ **Sharpening Spatial Filters**

- The **principal objective** of sharpening is **to highlight fine detail** in an image or to enhance detail that has been blurred, either in error or as a natural effect of a particular method of image acquisition.
- Uses of image sharpening vary and include applications ranging from **electronic printing** and **medical imaging** to **industrial inspection** and **autonomous guidance** in military systems.

➤ 1. Use of Second Derivatives for Enhancement–The Laplacian

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\frac{\partial^2 f}{\partial x^2} = f(x + 1, y) + f(x - 1, y) - 2f(x, y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y + 1) + f(x, y - 1) - 2f(x, y)$$

$$\nabla^2 f = [f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1)] - 4f(x, y)$$

0	1	0	1	1	1
1	-4	1	1	-8	1
0	1	0	1	1	1
0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

Thank
you

