Image Processing and Pattern Recognition (IPPR)

Lecture 2

Basanta Joshi, PhD

Asst. Prof., Depart of Electronics and Computer Engineering

Program Coordinator, MSc in Information and Communication Engineering

Member, Laboratory for ICT Research and Development (LICT)

Member, Research Management Cell (RMC)

Institute of Engineering

basanta@ioe.edu.np

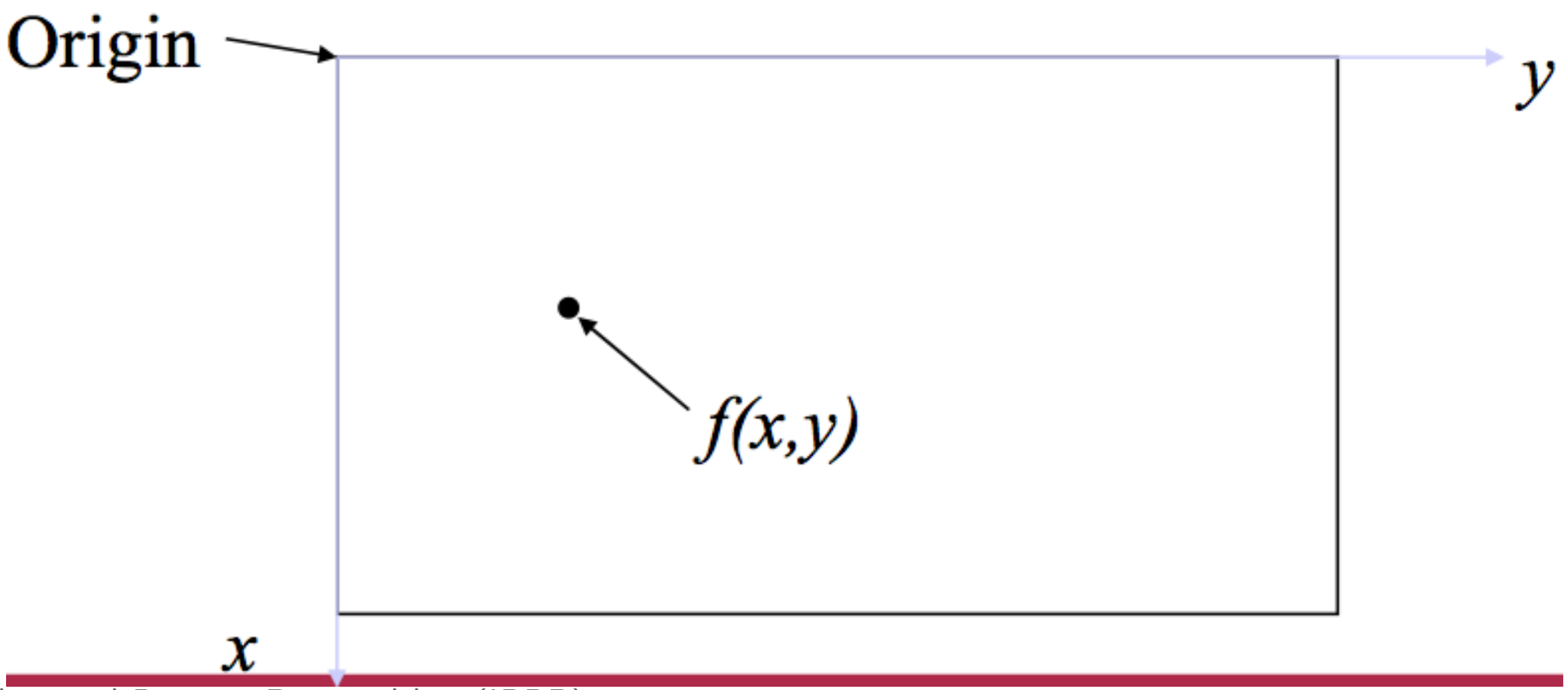
http://www.basantajoshi.com.np

https://scholar.google.com/citations?user=iocLiGcAAAAJ
https://www.researchgate.net/profile/Basanta_Joshi2

Digital image representation

Monochrome image (or simply image) refers to a 2- dimensional light intensity function *f(x,y)*

- *x* and *y* denote spatial coordinates
- the value of *f(x,y)* at *(x,y)* is proportional to the brightness (or *gray level*) of the image at that point



Digital image representation

A digital image is an image *f(x,y)* that has been discretized both in spatial coordinates and in brightness

- Considered as a matrix whose row and column indices represent a point in the image
- The corresponding matrix element value represents the gray level at that point
- The elements of such an array are referred to as:
  - image elements
    - picture elements (pixels or pels)

Steps in image processing

The *problem domain* in this example consists of pieces of mail and the objective is to read the address on each piece

Step 1: image acquisition

– Acquire a digital image using an image sensor

  • a monochrome or color TV camera: produces an entire image of the problem domain every 1/30 second

  • a line-scan camera: produces a single image line at a time, motion past the camera produces a 2-dimensional image

– If not digital, an *analog-to-digital* conversion process is required

– The nature of the image sensor (and the produced image) are determined by the application

  • Mail reading applications rely greatly on line-scan cameras

  • CCD and CMOS imaging sensors are very common in many applications

Steps in image processing

- Step 2: preprocessing

  – Key function: improve the image in ways that increase the chance for success of the other processes

  – In the mail example, may deal with contrast enhancement, removing noise, and isolating regions whose texture indicates a likelihood of alphanumeric information

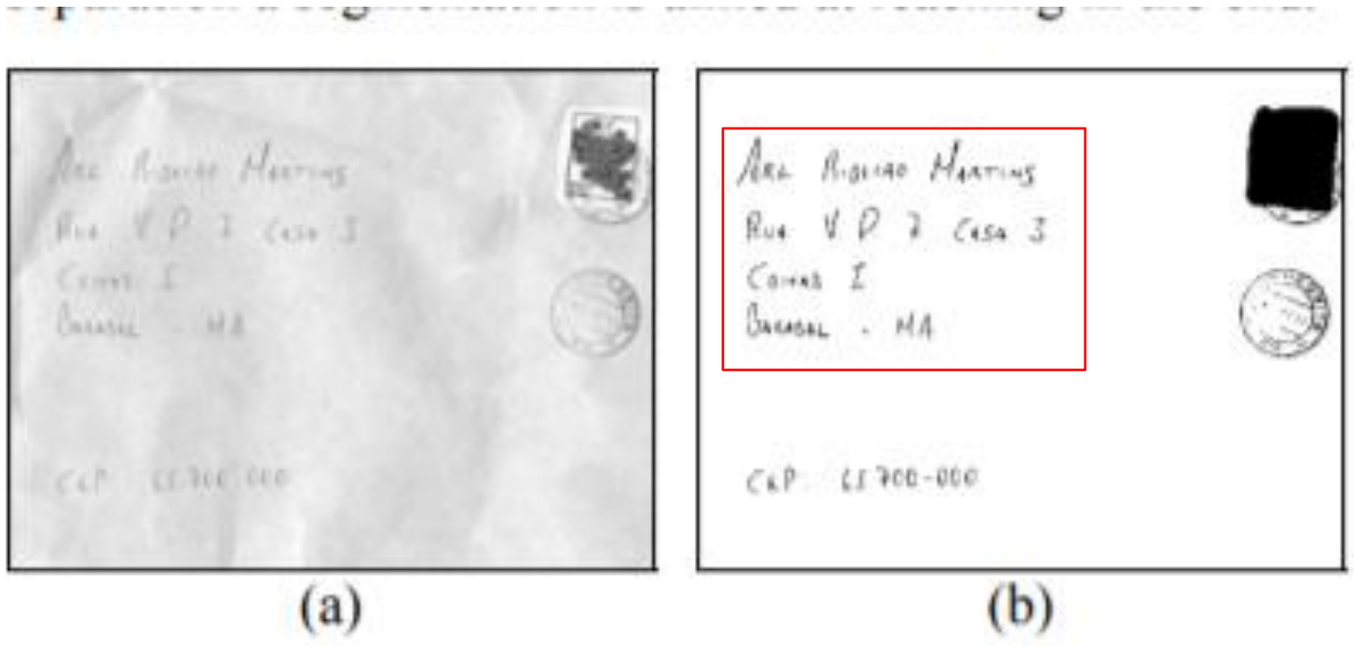


Figure 1. (a) Original image of a postal envelope; (b) Aimed segmented image from (a).

Steps in image processing

- Step 3: segmentation
  - Broadly defined: breaking an image into its constituent parts
  - In general, one of the most difficult tasks in image processing
    - Good segmentation simplifies the rest of the problem
    - Poor segmentation make the task impossible
  - Output is usually raw pixel data: may represent region boundaries, points in the region itself, etc.
    - Boundary representation can be useful when the focus is on external shape characteristics (e.g. corners, rounded edges, etc.)
    - Region representation is appropriate when the focus is on internal properties (e.g. texture or skeletal shape)
  - For the mail problem (character recognition) both representations can be necessary

Steps in image processing

- Step 4: representation & description
  - Representation: transforming raw data into a form suitable for computer processing
  - Description (also called feature extraction) deals with extracting features that result in some quantitative information of interest or features which are basic for differentiating one class of objects from another
  - In terms of character recognition, *descriptors* such as lakes (holes) and bays help differentiate one part of the alphabet from another

Steps in image processing

- Step 5: recognition & interpretation

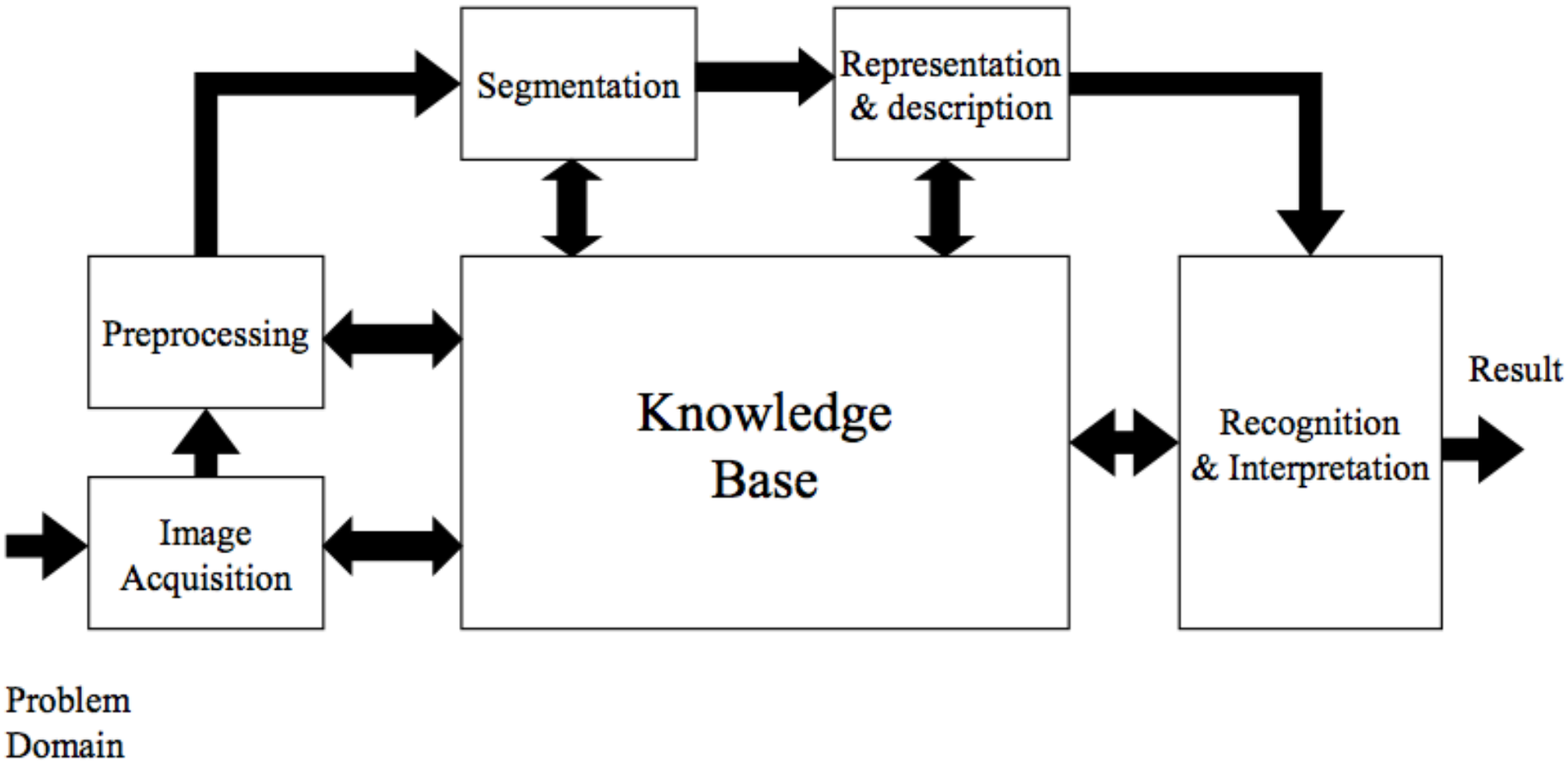
  – Recognition: The process which assigns a label to an object based on the information provided by its descriptors

    A may be the alphanumeric character A

  – Interpretation: Assigning meaning to an ensemble of recognized objects

    35487-0286 may be a ZIP code

Steps in image processing



Steps in image processing

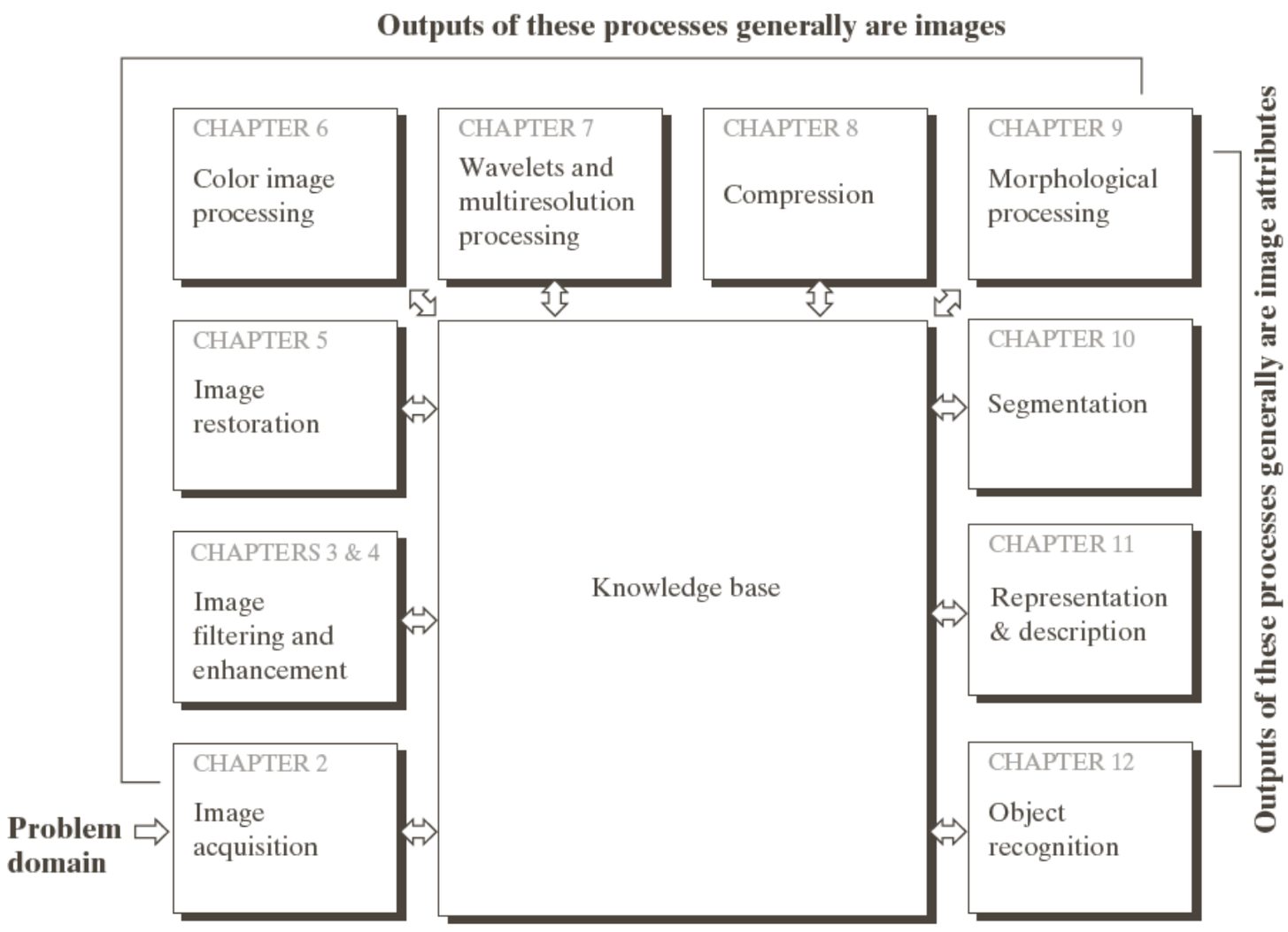


FIGURE 1.23 Fundamental steps in digital image processing. The chapter(s) indicated in the boxes is where the material described in the box is discussed.

A Knowledge Base

- Knowledge about a problem domain is coded into an image processing system in the form of a *knowledge database*
  - May be simple:
    - detailing areas of an image expected to be of interest
  - May be complex
    - A list of all possible defects of a material in a vision inspection system
  - Guides operation of each processing module
  - Controls interaction between modules
  - Provides feedback through the system

Steps in an image processing system

- Not all image processing systems would require all steps/processing modules
  - Image enhancement for human visual perception may not go beyond the preprocessing stage
- A knowledge database may not be required
- Processing systems which include recognition and interpretation are associated with image analysis systems in which the objective is autonomous (or at least partially automatic)

A simple imaging model

- An image is a 2-D light intensity function *f(x,y)*
- As light is a form of energy

  $0 < f(x,y) < \infty$

- f(x,y) may be expressed as the product of 2 components

$f(x,y)=i(x,y)r(x,y)$

- i(x,y) is the illumination: $0 < i(x,y) < \infty$
  - Typical values: 9000 foot-candles sunny day, 100 office room, 0.01 moonlight
- r(x,y) is the reflectance: $0 < r(x,y) < 1$
  - r(x,y)=0 implies total absorption
  - r(x,y)=1 implies total reflectance
  - Typical values: 0.01 black velvet, 0.80 flat white paint, 0.93 snow

A simple imaging model

- The intensity of a monochrome image *f* at (*x,y*) is the *gray level* (*l*) of the image at that point

$$L_{\min} \le l \le L_{\max}$$

- In practice $L_{min}=i_{min}\ r_{min}$ and $L_{max}=i_{max}\ r_{max}$
- As a guideline $L_{min} \approx 0.005$ and $L_{max} \approx 100$ for indoor image processing applications
- The interval [L_{min}, L_{max}] is called the *gray scale*
- Common practice is to shift the interval to [0,L] where l=0 is considered black and l=L is considered white. All intermediate values are shades of gray

Sampling and Quantization

- To be suitable for computer processing an image, f(x,y) must be digitized both spatially and in amplitude
- Digitizing the spatial coordinates is called *image sampling*
- Amplitude digitization is called gray-level quantization
- f(x,y) is approximated by equally spaced samples in the form of an NxM array where each element is a discrete quantity

$$f(x,y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ . & & & \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

Sampling and Quantization

- Common practice is to let N and M be powers of two; N=2^n and M=2^k
- And G=2^m where G denotes the number of gray levels
- The assumption here is that gray levels are equally space in the interval [0,L]
- The number of bits, b, necessary to store the image is then

$$b = NxMxm$$

$$or\ if\ N = M$$

$$b = N^2 m$$

- For example, a 128x128 image with 64 gray levels would require 98,304 bits of storage

Sampling and Quantization

Number of storage bits for various values of N and k.

N/k	1 ($L = 2$)	2 ($L = 4$)	3 ($L = 8$)	4 ($L = 16$)	5 ($L = 32$)	6 ($L = 64$)	7 ($L = 128$)	8 ($L = 256$)
32	1,024	2,048	3,072	4,096	5,120	6,144	7,168	8,192
64	4,096	8,192	12,288	16,384	20,480	24,576	28,672	32,768
128	16,384	32,768	49,152	65,536	81,920	98,304	114,688	131,072
256	65,536	131,072	196,608	262,144	327,680	393,216	458,752	524,288
512	262,144	524,288	786,432	1,048,576	1,310,720	1,572,864	1,835,008	2,097,152
1024	1,048,576	2,097,152	3,145,728	4,194,304	5,242,880	6,291,456	7,340,032	8,388,608
2048	4,194,304	8,388,608	12,582,912	16,777,216	20,971,520	25,165,824	29,369,128	33,554,432
4096	16,777,216	33,554,432	50,331,648	67,108,864	83,886,080	100,663,296	117,440,512	134,217,728
8192	67,108,864	134,217,728	201,326,592	268,435,456	335,544,320	402,653,184	469,762,048	536,870,912

- How many samples and gray levels are required for a “good” approximation?
- The resolution (the degree of discernible detail) depends strongly on these two parameters

Effects of reducing spatial resolution

256x256

128x128

64x64

32x32

Pixel replication occurs as resolution is decreased

Effects of reducing gray levels

256

128

64

32

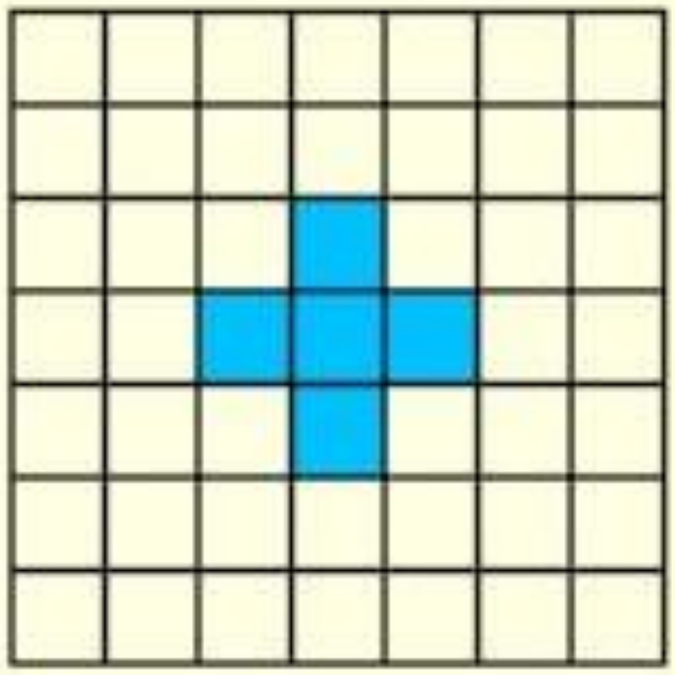
Ridgelike structures develop as gray level is decreased: false contours

Basic relationships between pixels

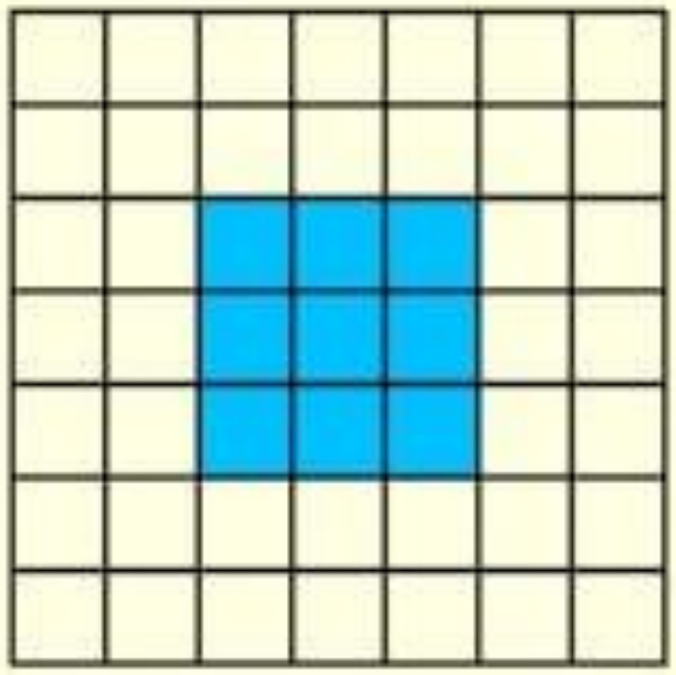
- An image is denoted by: f(x,y)
- Lowercase letters (e.g. *p, q*) will denote individual pixels
- A subset of f(x,y) is denoted by *S*
- Neighbors of a pixel:
  - A pixel *p* at (x,y) has 4 horizontal/vertical neighbors at
    - (x+1,y),(x-1,y),(x,y+1)and(x,y-1)
    - called the *4-neighbors* of *p*: $N_4(p)$
  - A pixel *p* at (x,y) has 4 diagonal neighbors at
    - (x+1,y+1),(x+1,y-1),(x-1,y+1)and(x-1,y-1)
    - called the *diagonal–neighbors* of *p*:$N_D(p)$
  - The *4-neighbors* and the *diagonal-neighbors* of *p* are called the *8-neighbors* of *p*: $N_8(p)$

Types of Pixels Neighborhoods

- **Basic Relationships Between Pixels**



4-connected

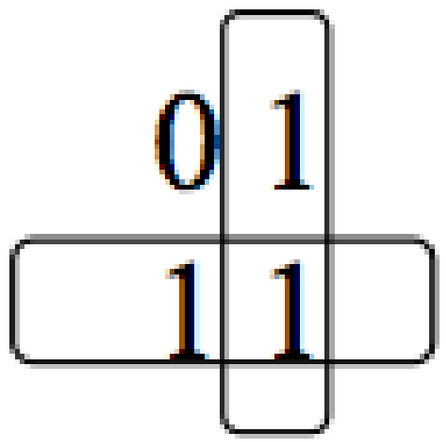


8-connected

6-connected

Connectivity between pixels

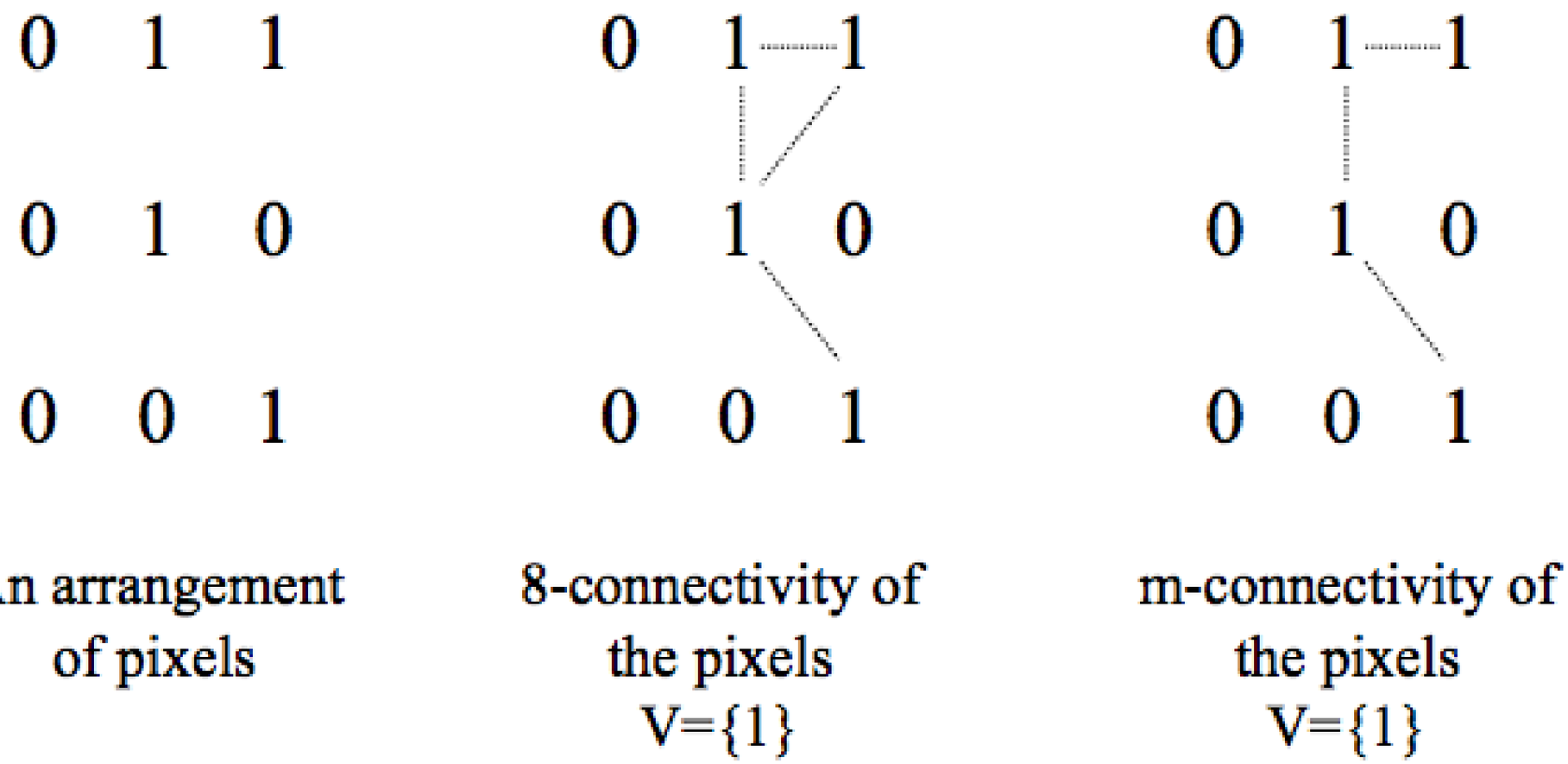
- Connectivity is an important concept in establishing boundaries of object and components of regions in an image
- When are two pixels connected?
  - If they are adjacent in some sense (say they are 4-neighbors)
  - and, if their gray levels satisfy a specified criterion of similarity (say they are equal)
- Example: given a binary image (e.g. gray scale = [0,1]), two pixels may be 4-neighbors but are not considered connected unless they have the same value



Connectivity between pixels

- Let *V* be the set of values used to determine connectivity
  - For example, in a binary image, *V*={1} for the connectivity of pixels with a value of 1
  - In a gray scale image, for the connectivity of pixels with a range of intensity values of, say, 32 to 64, it follows that *V*={32,33,...,63,64}
  - Consider three types of connectivity
    - *4-connectivity*: Pixels *p* and *q* with values from *V* are *4-connected* if *q* is in the set $N_4(p)$
    - *8-connectivity*: Pixels *p* and *q* with values from *V* are *8-connected* if *q* is in the set $N_8(p)$
    - *m-connectivity (mixed)*: Pixels *p* and *q* with values from *V* are *m-connected* if
      - *q* is in the set $N_4(p)$,*or*
      - *q* is in the set $N_D(p)$ *and* the set $N_4(p) \cap N_4(q)$ is empty (This is the set of pixels that are 4-neighbors of *p* and *q* and whose values are from *V*)

Connectivity between pixels



An arrangement of pixels

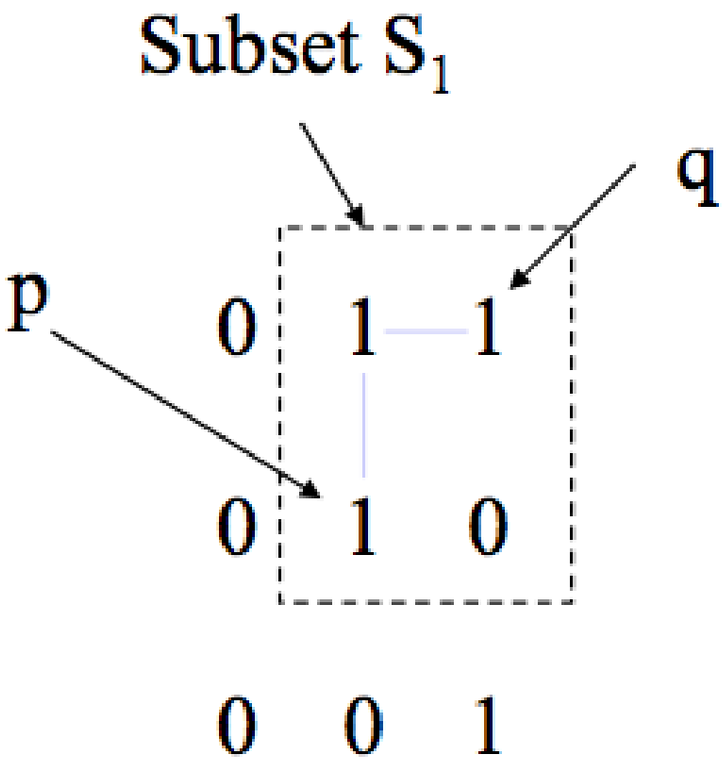
8-connectivity of the pixels V={1}

m-connectivity of the pixels V={1}

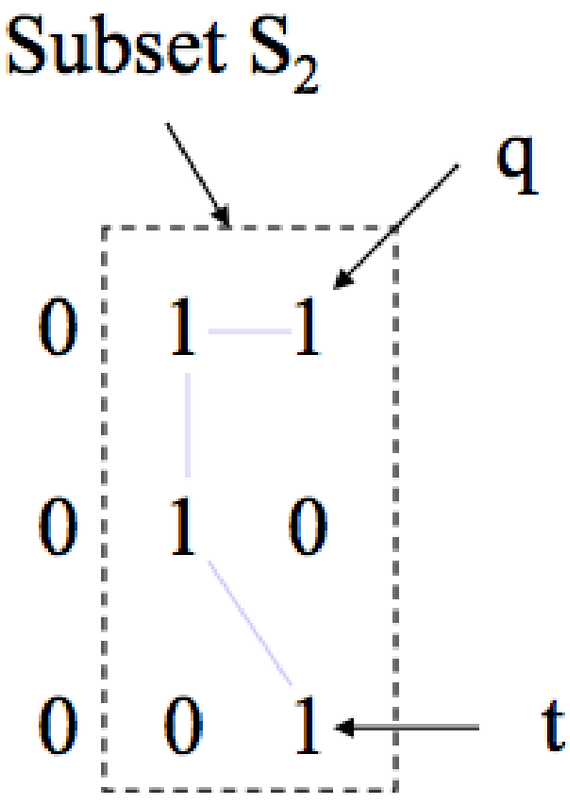
Pixel adjacencies and paths

- Pixel *p* is adjacent to *q* if they are connected
  - We can define 4-, 8-, or m-adjacency depending on the specified type of connectivity
- Two image subsets S_1 and S_2 are adjacent if some pixel in S_1 is adjacent to S_2
- A path from *p* at (*x,y*) to *q* at (*s,t*) is a sequence of distinct pixels with coordinates (x_0,y_0), (x_1,y_1),....., (x_n,y_n)
  - Where $(x_0,y_0)=(x,y)$ and $(x_n,y_n)=(s,t)$ and
  - (x_i,y_i) is adjacent to (x_{i-1},y_{i-1}) for $1<= i <= n$
  - n is the *length* of the path
- If *p* and *q* are in *S*, then *p* is connected to *q* in *S* if there is a path from *p* to *q* consisting entirely of pixels in *S*

Example paths



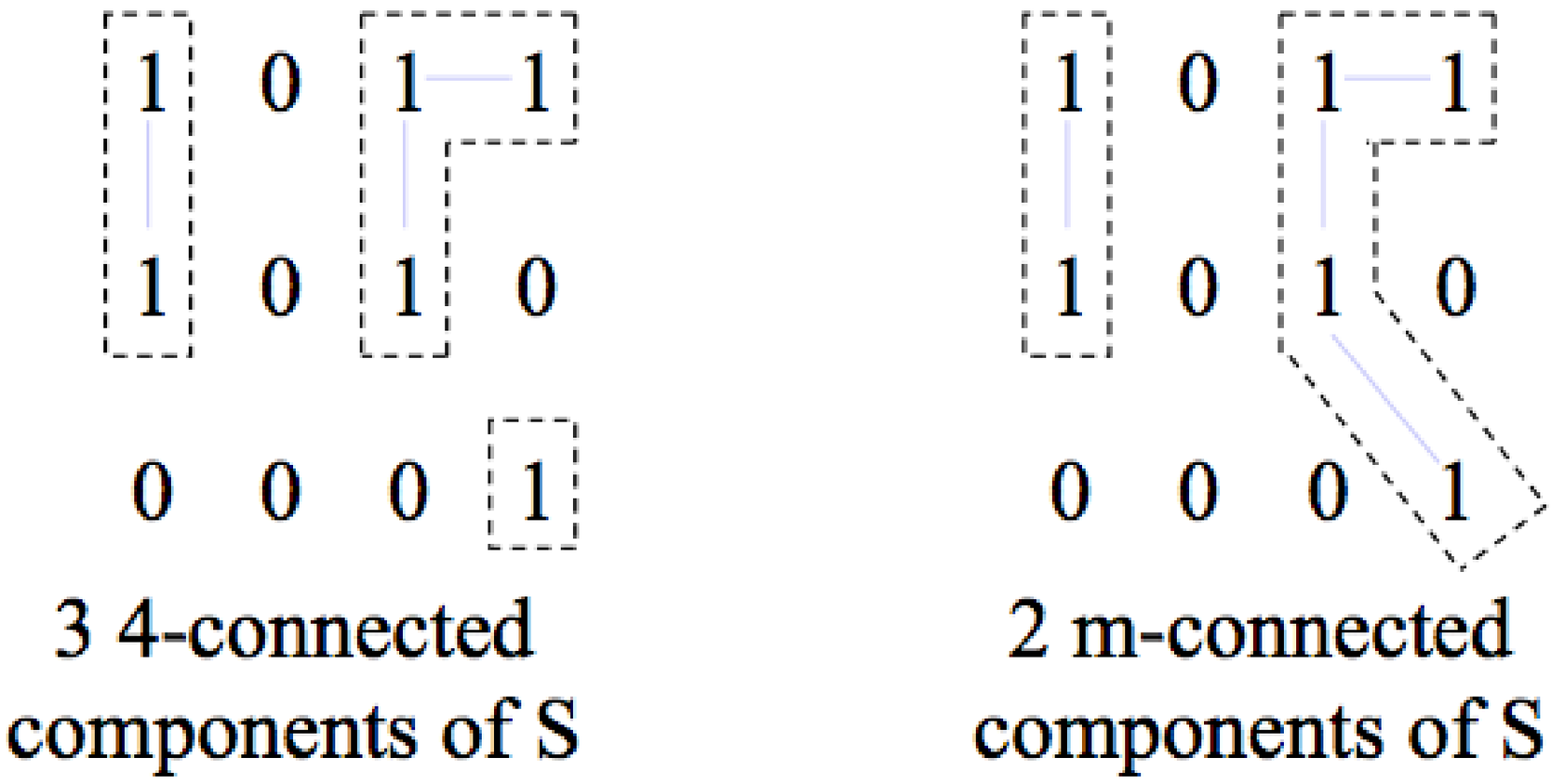
A 4-connected path from p to q (n=2). p and q are connected in S_1



An m-connected path from t to q (n=3). t and q are connected in S_2

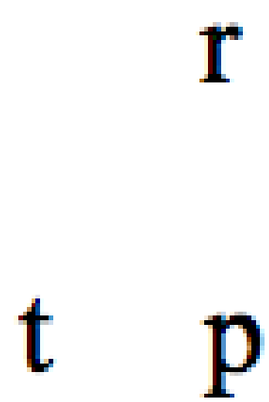
Connected components

- For any pixel *p* in *S*, the set of pixels connected to *p* form a connected component of *S*
- Distinct connected components in *S* are said to be *disjoint*



Labeling 4-connected components

• Consider scanning an image pixel by pixel from left to right and top to bottom



– Assume, for the moment, we are interested in 4-connected components

– Let *p* denote the pixel of interest, and *r* and *t* denote the upper and left neighbors of *p*, respectively

– The nature of the scanning process assures that *r* and *t* have been encountered (and labeled if 1) by the time *p* is encountered

Labeling 4-connected components

- Consider the following procedure

if p=0 continue to the next position

if r=t=0 assign a new label to p (Ln)

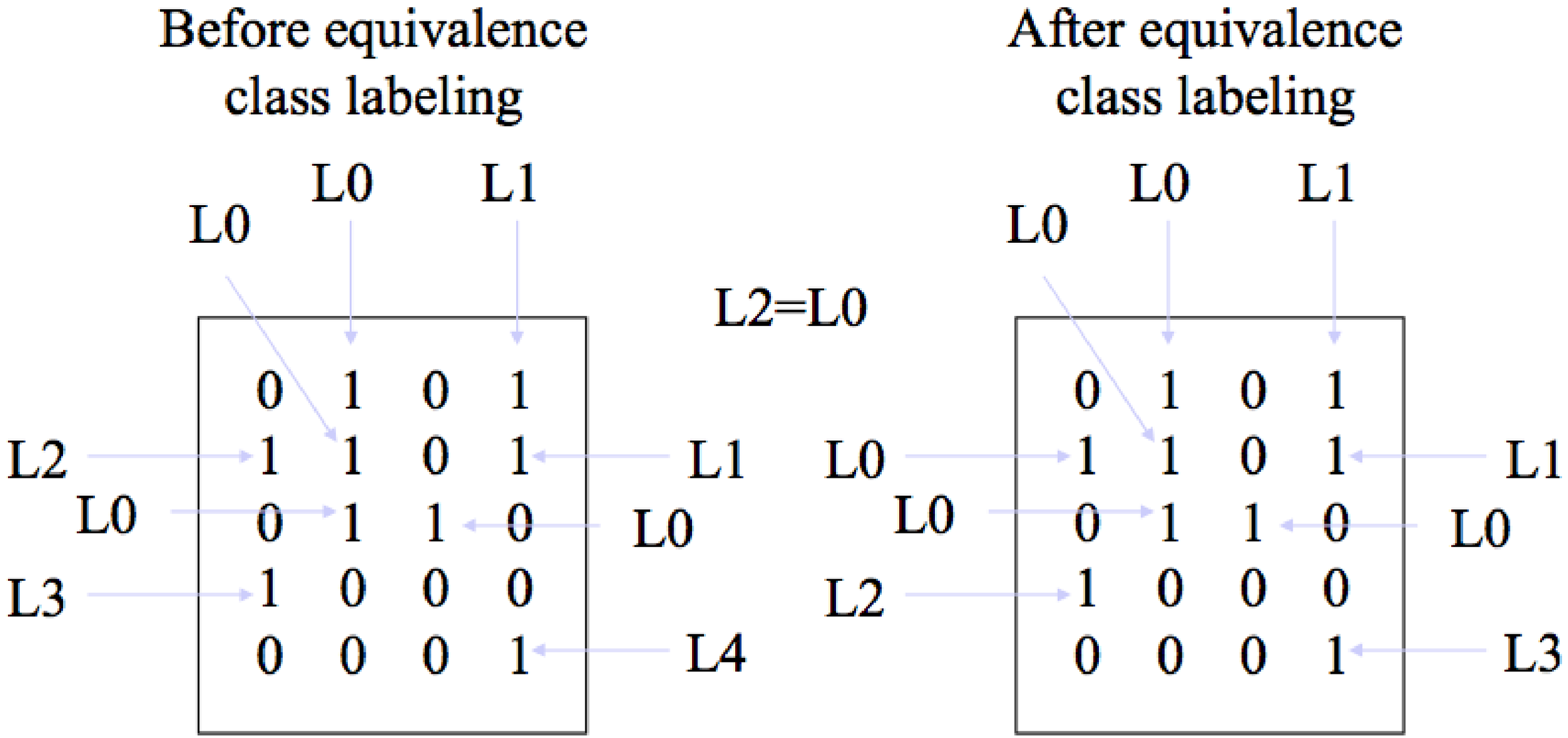
if r=t=1 and they have the same label, assign that label to p

if only one of r and t are 1, assign its label to p

if r=t=1 and they have different labels, assign one label to p and note that the two labels are equivalent (that is r and t are connected through p)

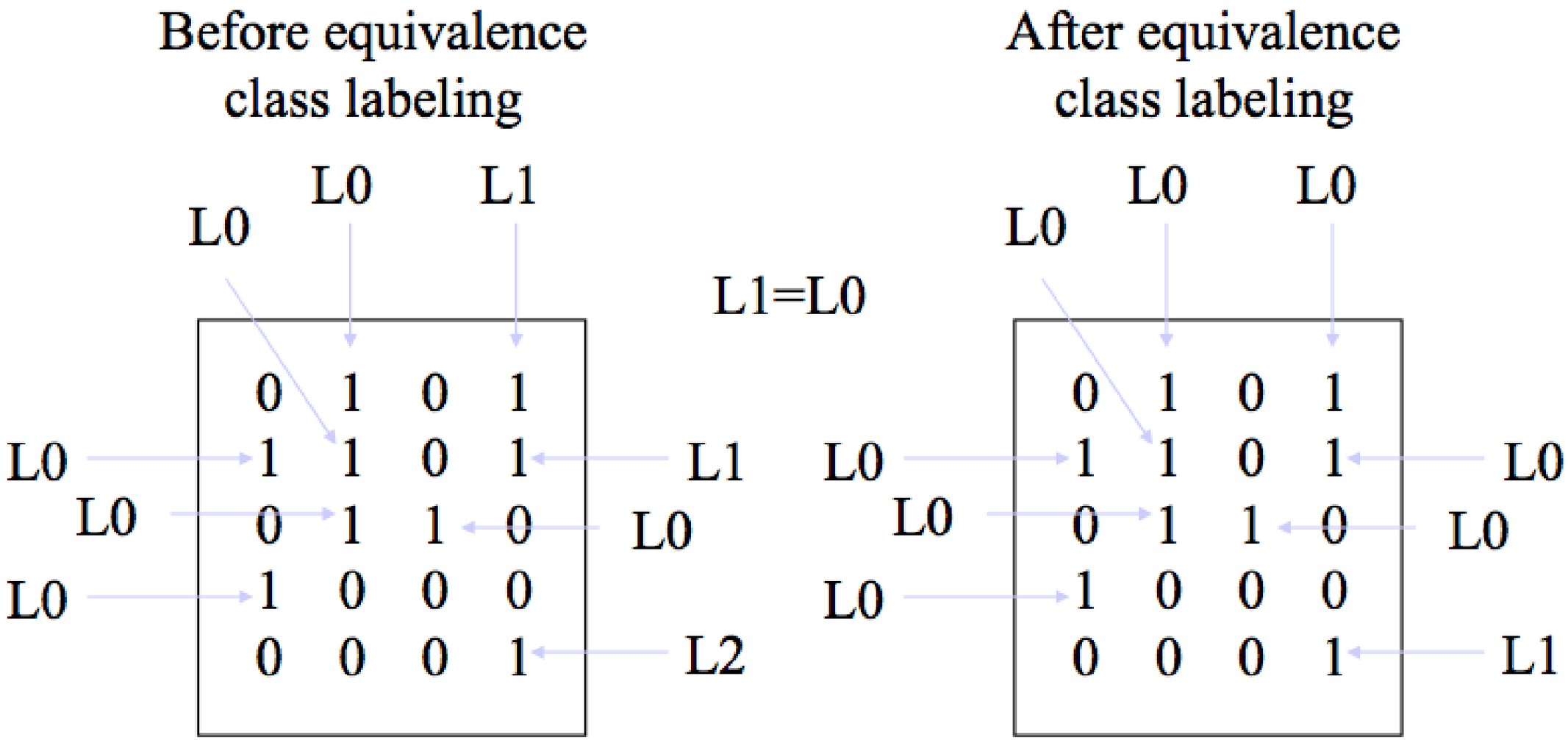
At the end of the scan, sort pairs of equivalent labels into equivalence classes and assign a different label to each class

Labeling 4-connected components (example)



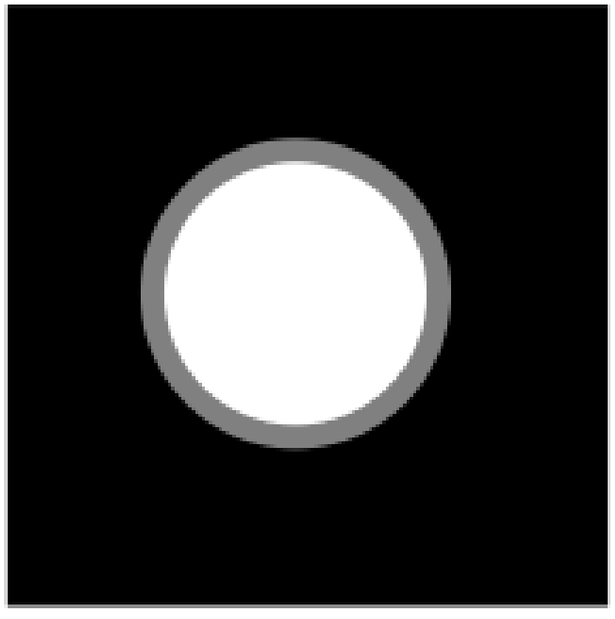
Labeling 8-connected components

- Proceed as in the 4-connected component labeling case, but also examine two upper diagonal neighbors (*q* and *s*) of *p*

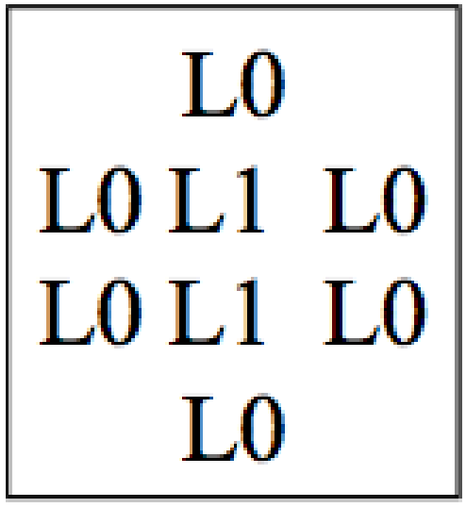


Labeling connected components in non-binary images

- The 4-connected or 8-connected labeling schemes can be extended to gray level images
- The set *V* may be used to connect into a component only those pixels within a specified range of pixel values



000	000	000	000	000
000	000	128	000	000
000	128	255	128	000
000	128	255	128	000
000	000	128	000	000
000	000	000	000	000



Distance measures

- Given pixels *p*, *q*, and *z* at (*x*,*y*), (*s*,*t*) and (*u*,*v*) respectively,
- *D* is a *distance function* (or *metric*) if:
  - D(p,q) ≥ 0 (D(p,q)=0 iff p=q),
  - D(p,q) = D(q,p), and
  - D(p,z) ≤ D(p,q) + D(q,z).
- The *Euclidean distance* between *p* and *q* is given by:

$$D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2}$$

- The pixels having distance less than or equal to some value *r* from (x,y) are the points contained in a disk of radius *r* centered at (x,y)

Distance measures

- The D_4 distance (also called the *city block distance*) between *p* and *q* is given by:

$$D_4(p,q) = |x-s| + |y-t|$$

- The pixels having a D_4 distance less than some *r* from (x,y) form a diamond centered at (x,y)
- Example: pixels where $D_4 \leq 2$

```
      2
    2 1 2
  2 1 0 1 2
    2 1 2
      2
```

Note: Pixels with $D_4=1$ are the 4-neighbors of (x,y)

Distance measures

- The D_8 distance (also called the *chessboard distance*) between *p* and *q* is given by:

$$D_8(p,q) = \max(|x-s|, |y-t|)$$

- The pixels having a D_8 distance less than some *r* from (x,y) form a square centered at (x,y)
- Example: pixels where $D_8 \leq 2$

2 2 2 2 2
2 1 1 1 2
2 1 0 1 2
2 1 1 1 2
2 2 2 2 2

Note: Pixels with D_8=1 are the 8-neighbors of (x,y)

Distance measures and connectivity

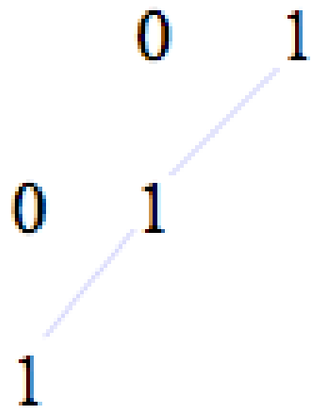
- The D_4 distance between two points *p* and *q* is the shortest 4-path between the two points
- The D_8 distance between two points *p* and *q* is the shortest 8-path between the two points
- D_4 and D_8 may be considered, regardless of whether a connected path exists between them, because the definition of these distances involves only the pixel coordinates
- For *m-connectivity*, the value of the distance (the length of the path) between two points depends on the values of the pixels along the path

Distance measures and m-connectivity

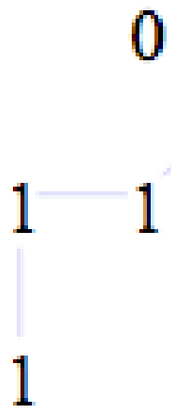
p_3 p_4
p_1 p_2
p

- Consider the given arrangement of pixels and assume
  - p, p_2 and p_4 =1
  - p_1 and p_3 can be 0 or 1
- If V={1} and p_1 and p_3 are 0, the m-distance (p, p_4) is 2 If either p_1 or p_3 are 1, the m-distance (p, p_4) is 3 If p_1 and p_3 are 1, the m-distance (p, p_4) is 4

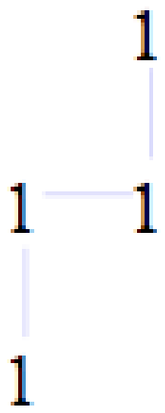
M-connectivity example



m-distance(p,p_4)=2



m-distance(p,p_4)=3



m-distance(p,p_4)=4

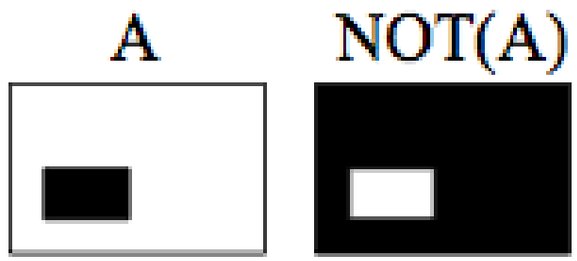
Arithmetic & logic operations

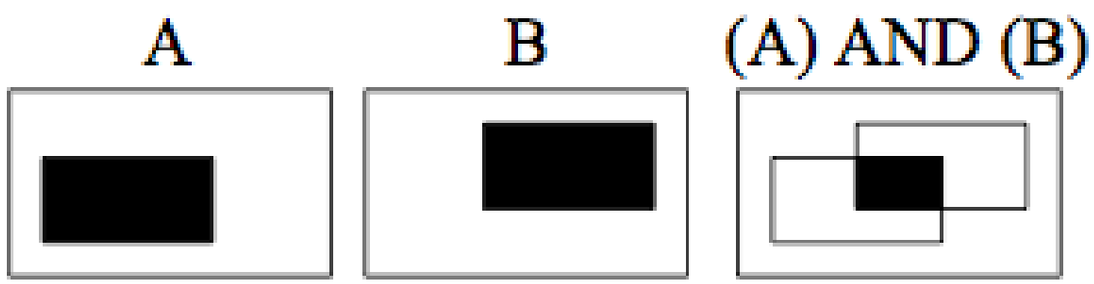
- Arithmetic & logic operations on images used extensively in most image processing applications
  - May cover the entire image or a subset Arithmetic operation between pixels *p* and *q* are defined as:
  - Addition: (*p*+*q*)
    - Used often for image averaging to reduce noise
  - Subtraction: (*p*-*q*)
    - Used often for static background removal
  - Multiplication: (*p***q*) (or *pq*, $p \times q$)
    - Used to correct gray-level shading
  - Division: ($p \div q$) (or *p*/*q*)
    - As in multiplication

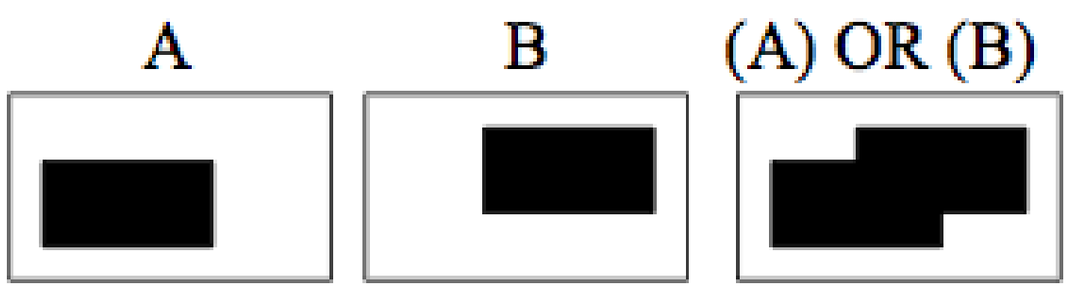
Logic operations

- Logic operation between pixels *p* and *q* are defined as:
  - AND: *p* AND *q* (also p·q)
  - OR: *p* OR *q* (also p+q)
  - COMPLEMENT: NOT*q* (also *q*’)
- Form a *functionally complete* set
- Applicable to binary images
- Basic tools in binary image processing, used for:
  - Masking
  - Feature detection
  - Shape analysis

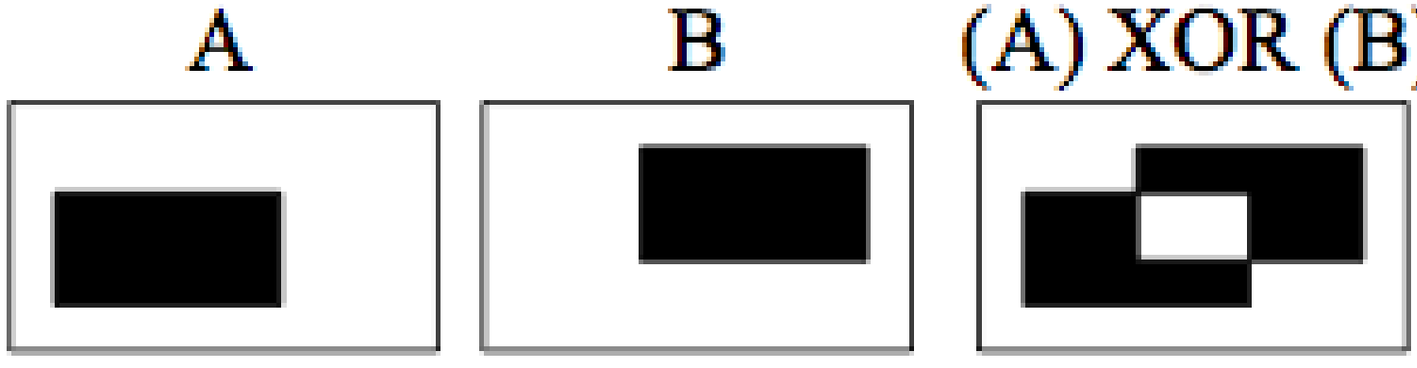
Examples of logic operations

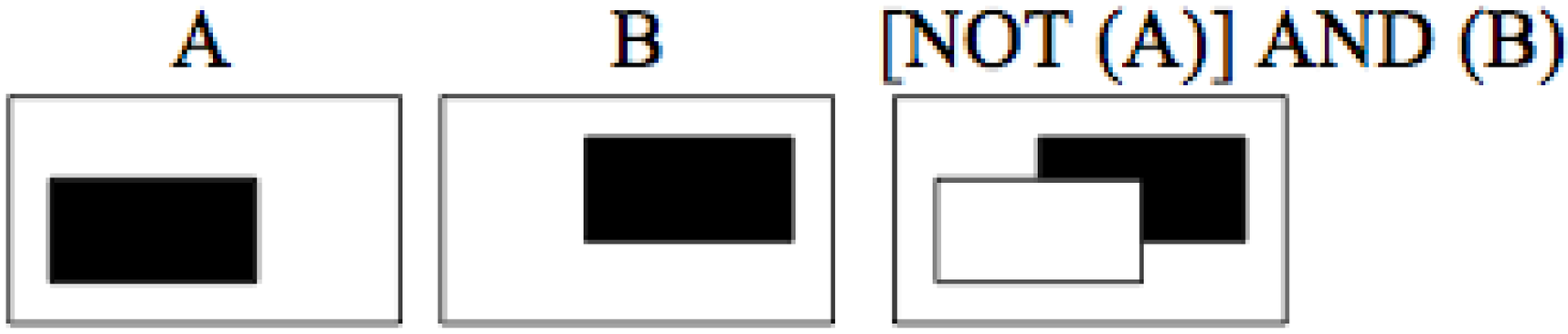






Examples of logic operations





Neighborhood-oriented operations

- Arithmetic and logical operations may take place on a subset of the image
  - Typically neighborhood oriented
- Formulated in the context of *mask* operations (also called *template*, *window* or *filter* operations)
- Basic concept:let the value of a pixel be a function of its (current) gray level and the gray level of its neighbors (in some sense)

Neighborhood-oriented operations

- Consider the following subset of pixels in an image
- Suppose we want to filter the image by replacing the value at Z_5 with the average value of the pixels in a 3x3 region centered around Z_5
- Perform an operation of the form:

$$z = \frac{1}{9}(z_1 + z_2 + \ldots + z_9) = \frac{1}{9}\sum_{i=1}^{9} z_i$$

	Z_1	Z_2	Z_3
	Z_4	Z_5	Z_6
	Z_7	Z_8	Z_9

- and assign to z_5 the value of z

Neighborhood-oriented operations

- In the more general form, the operation may look like:

$$z = (w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9) = \sum_{i=1}^{9} w_i z_i$$

- This equation is widely used in image processing
- Proper selection of coefficients (weights) allows for operations such as
  - noise reduction
  - region thinning
  - edge detection

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

What Is Image Enhancement?

Image enhancement is the process of making images more useful

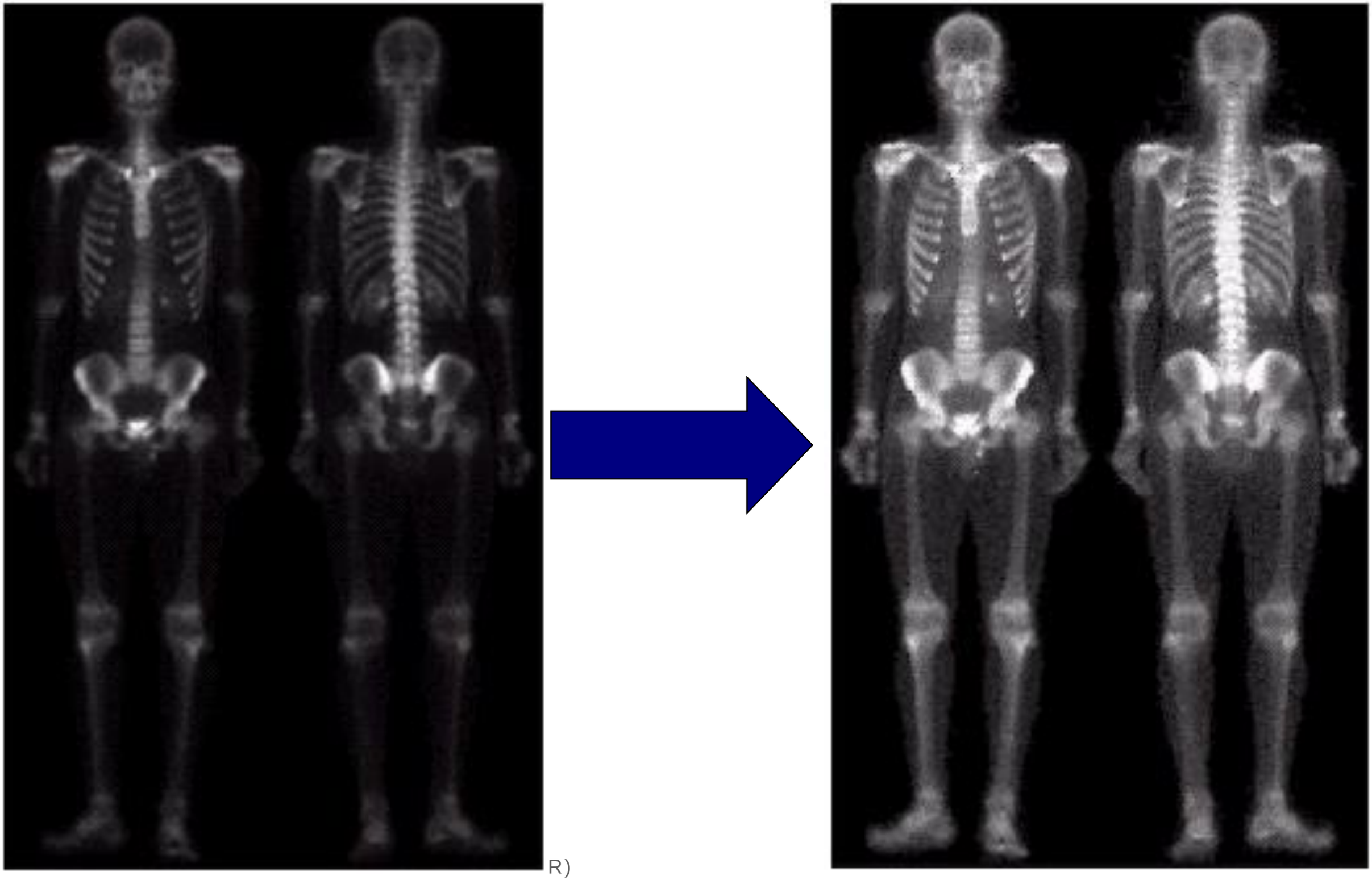
The reasons for doing this include:

- Highlighting interesting detail in images
- Removing noise from images
- Making images more visually appealing

Image Enhancement Examples

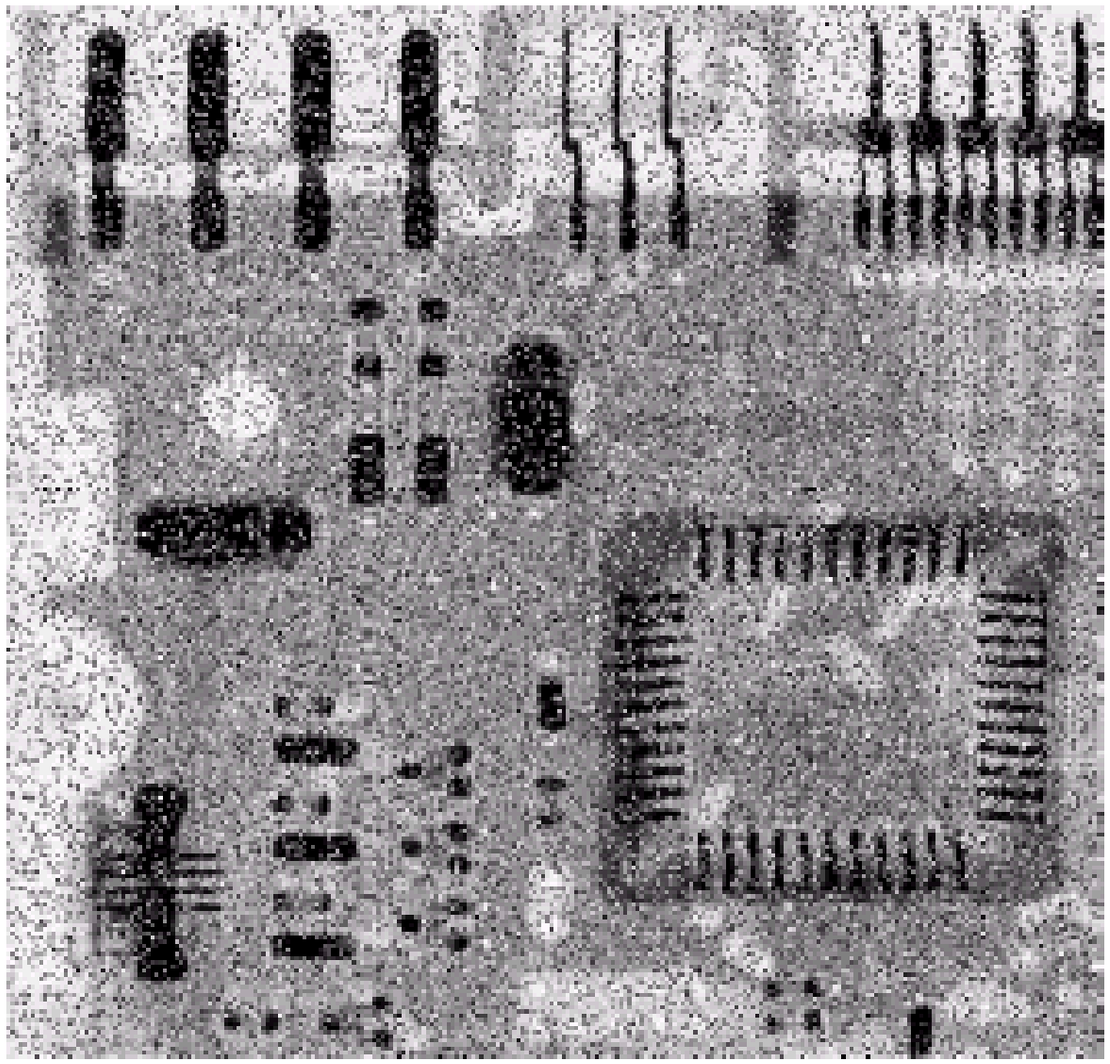
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

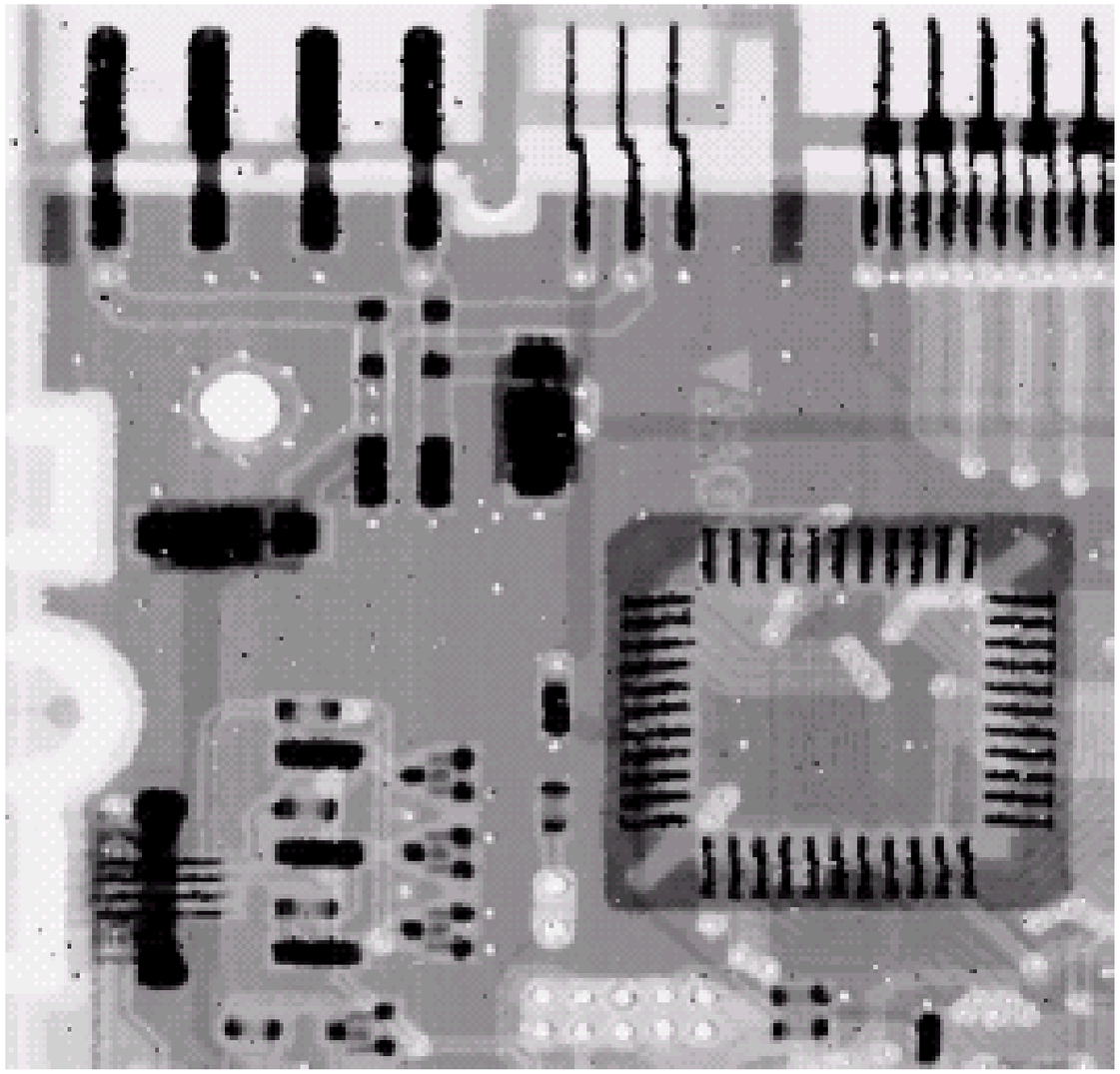
Image Enhancement Examples (cont…)



R)

Image Enhancement Examples (cont…)





Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Image Enhancement Examples (cont…)

Spatial & Frequency Domains

There are two broad categories of image enhancement techniques

- Spatial domain techniques
  - Direct manipulation of image pixels
- Frequency domain techniques
  - Manipulation of Fourier transform or wavelet transform of an image

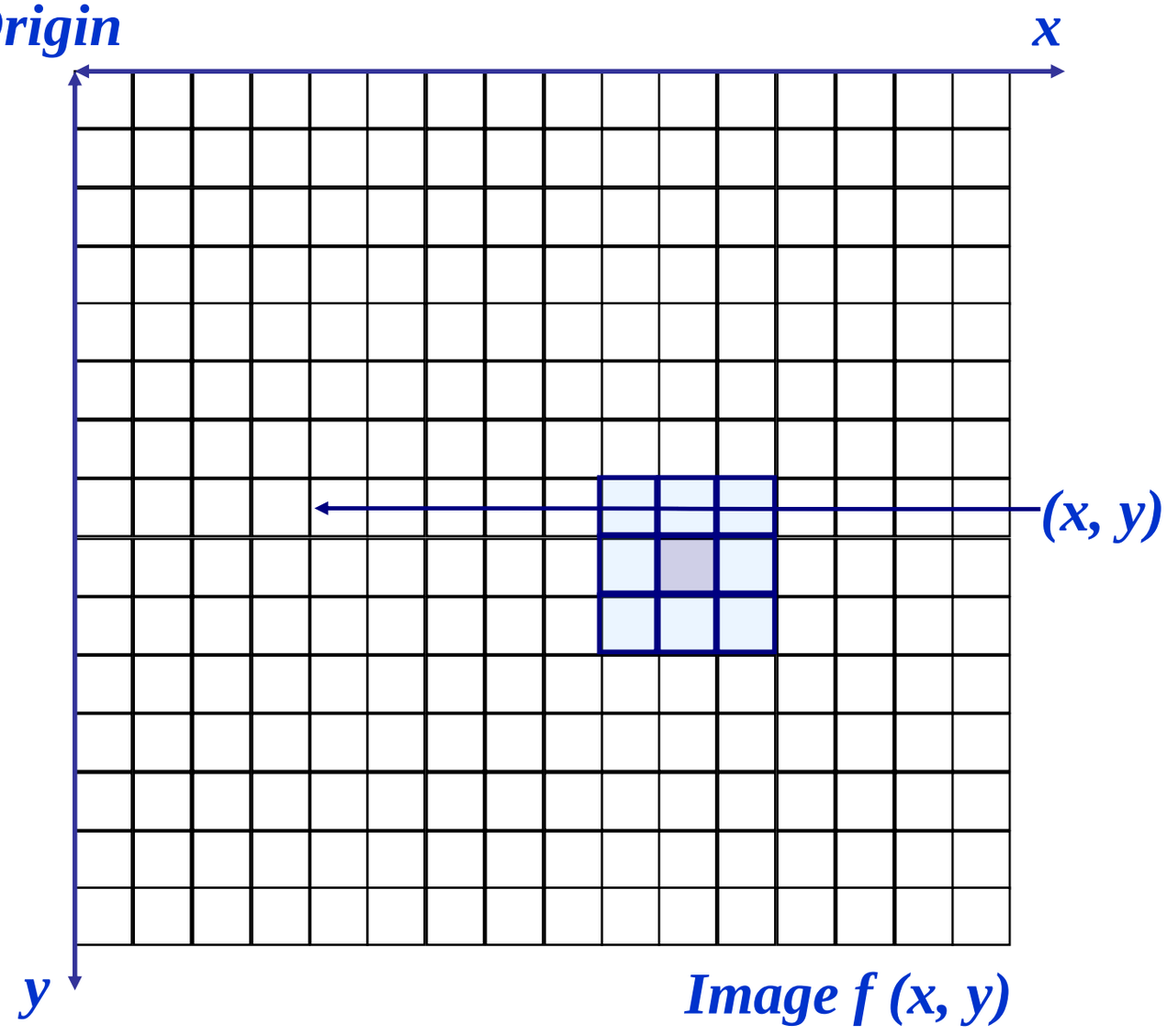
For the moment we will concentrate on techniques that operate in the spatial domain

Basic Spatial Domain Image Enhancement

Most spatial domain enhancement operations can be reduced to the form

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

where $f(x, y)$ is the input image, $g(x, y)$ is the processed image and T is some operator defined over some neighbourhood of (x, y)



Point Processing

The simplest spatial domain operations occur when the neighbourhood is simply the pixel itself

In this case T is referred to as a *grey level transformation function* or a *point processing operation*

Point processing operations take the form

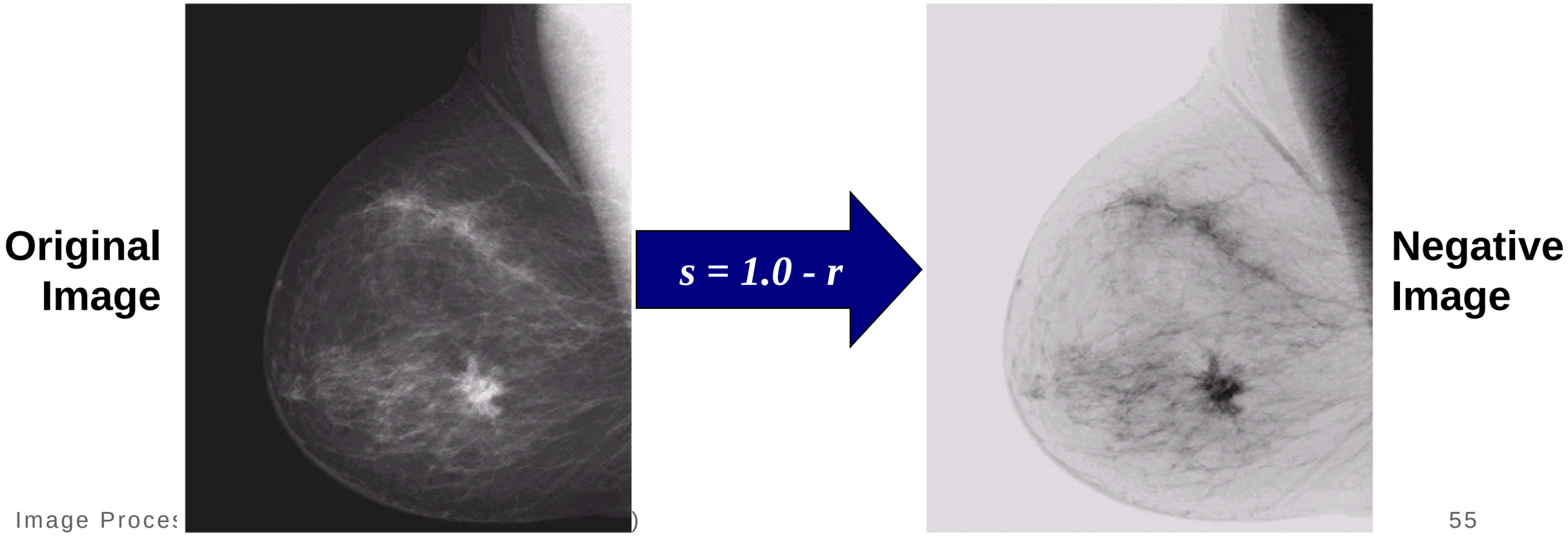
$$s = T(r)$$

where s refers to the processed image pixel value and r refers to the original image pixel value

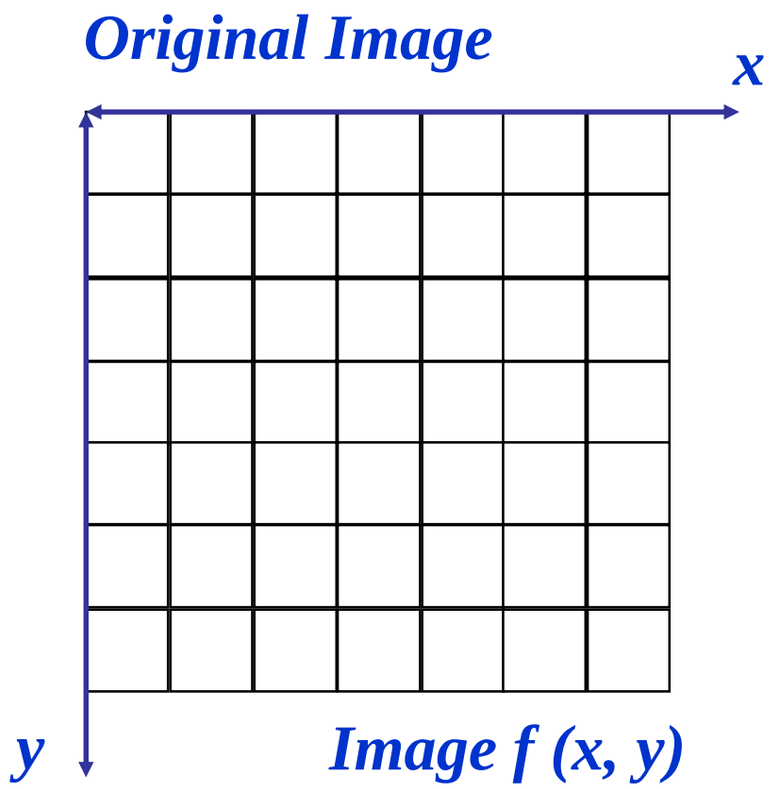
Point Processing Example: Negative Images

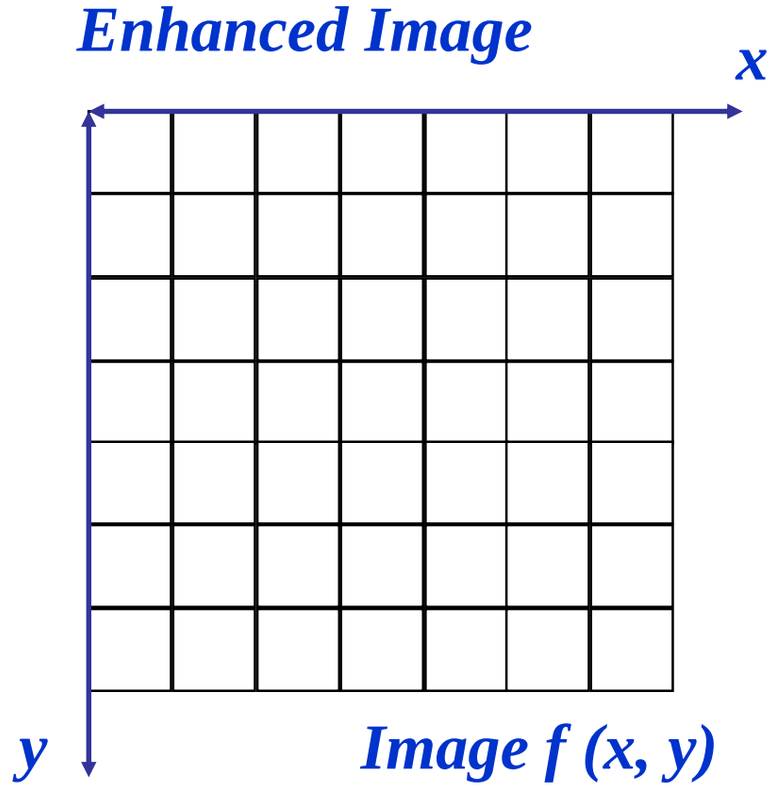
Negative images are useful for enhancing white or grey detail embedded in dark regions of an image

– Note how much clearer the tissue is in the negative image of the mammogram below



Point Processing Example: Negative Images (cont…)



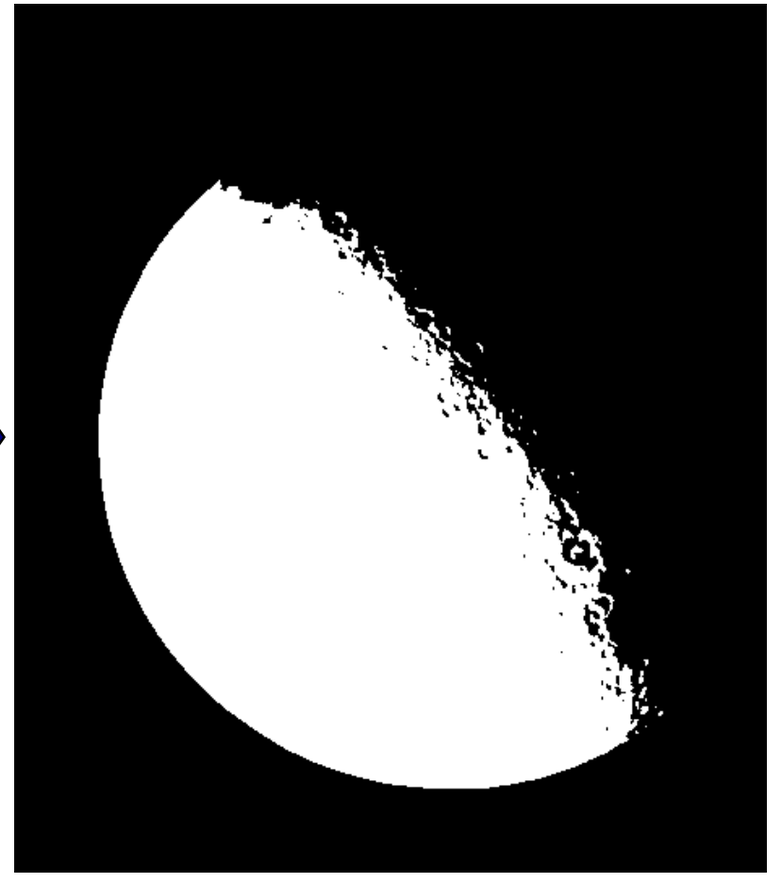


$$s = intensity_{max} - r$$

Point Processing Example: Thresholding

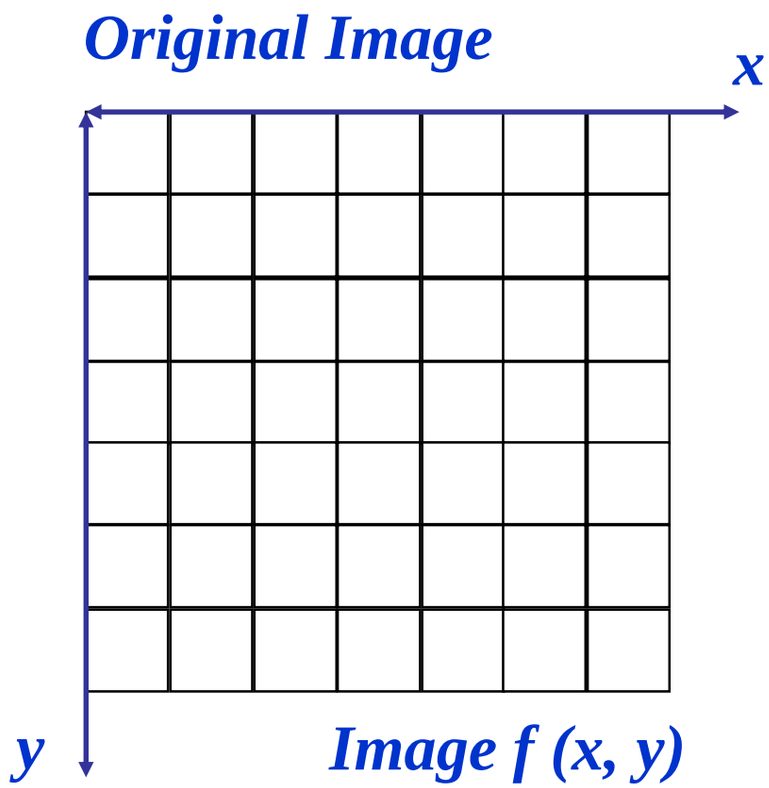
Thresholding transformations are particularly useful for segmentation in which we want to isolate an object of interest from a background

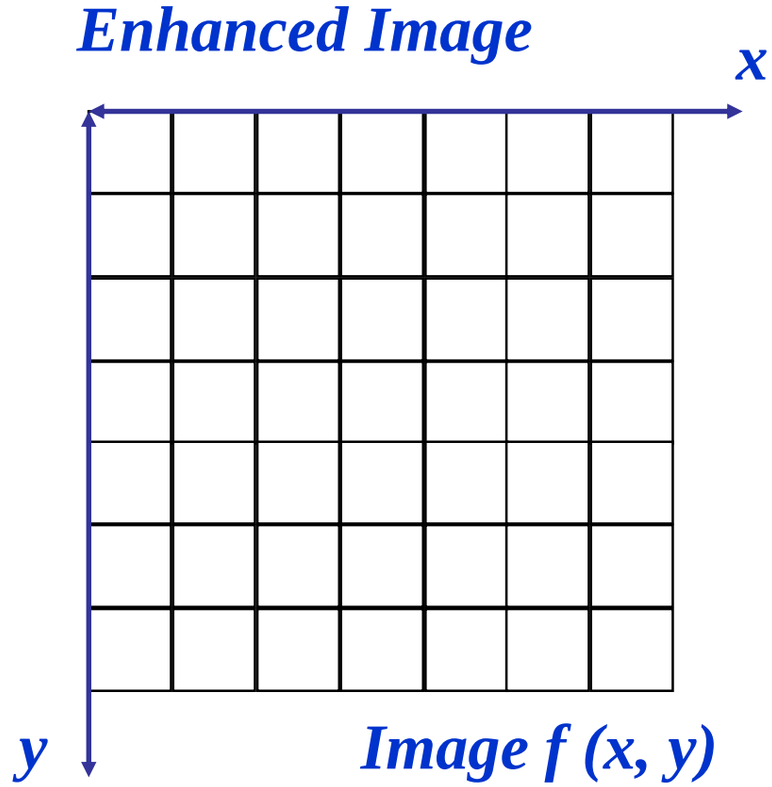
$$s = \begin{cases} 1.0 & r > \text{threshold} \\ 0.0 & r <= \text{threshold} \end{cases}$$



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

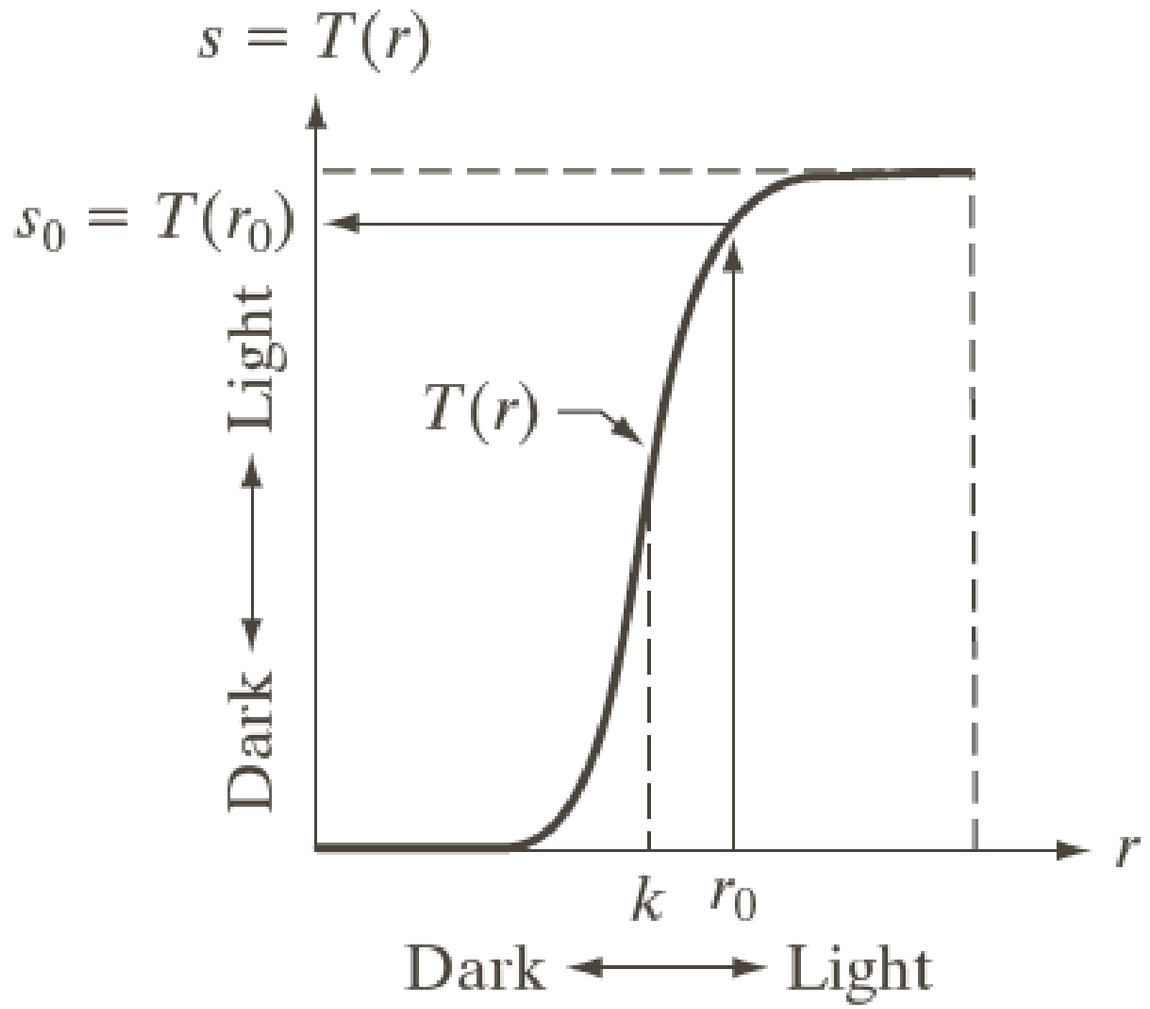
Point Processing Example: Thresholding (cont…)

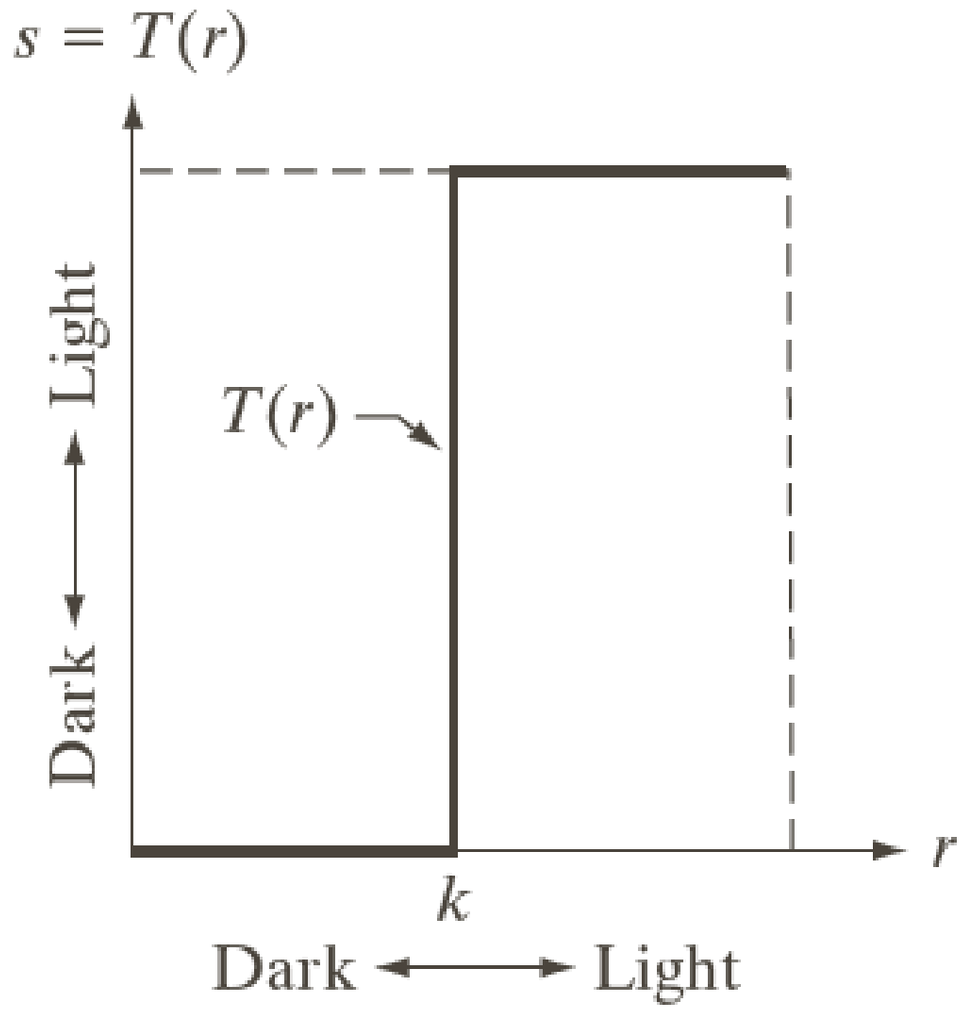




$$s = \begin{cases} 1.0 & r > threshold \\ 0.0 & r <= threshold \end{cases}$$

Intensity Transformations



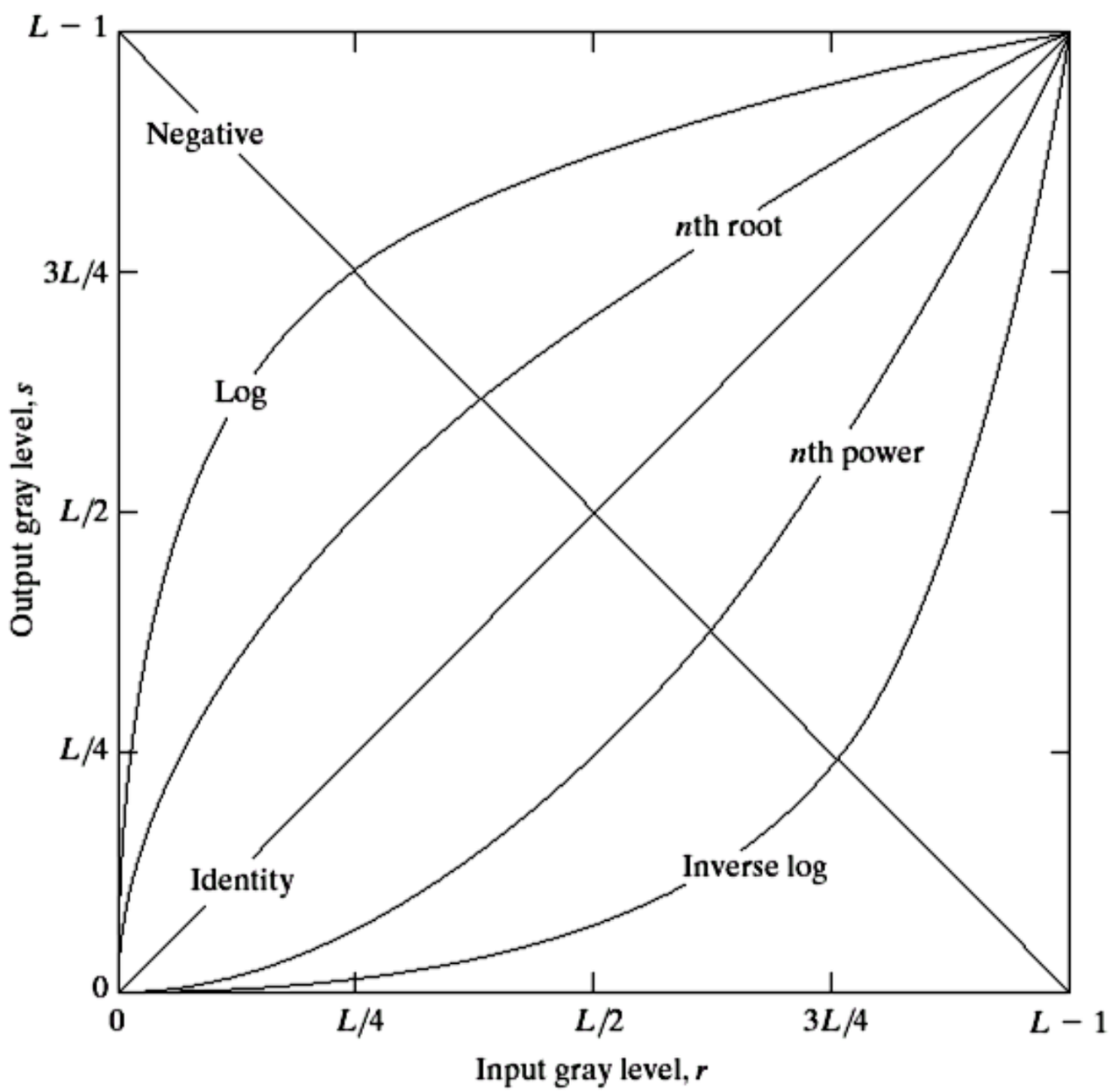


Basic Grey Level Transformations

There are many different kinds of grey level transformations

Three of the most common are shown here

- Linear
  - Negative/Identity
- Logarithmic
  - Log/Inverse log
- Power law
  - n^{th} power/n^{th} root



Logarithmic Transformations

The general form of the log transformation is

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

The log transformation maps a narrow range of low input grey level values into a wider range of output values

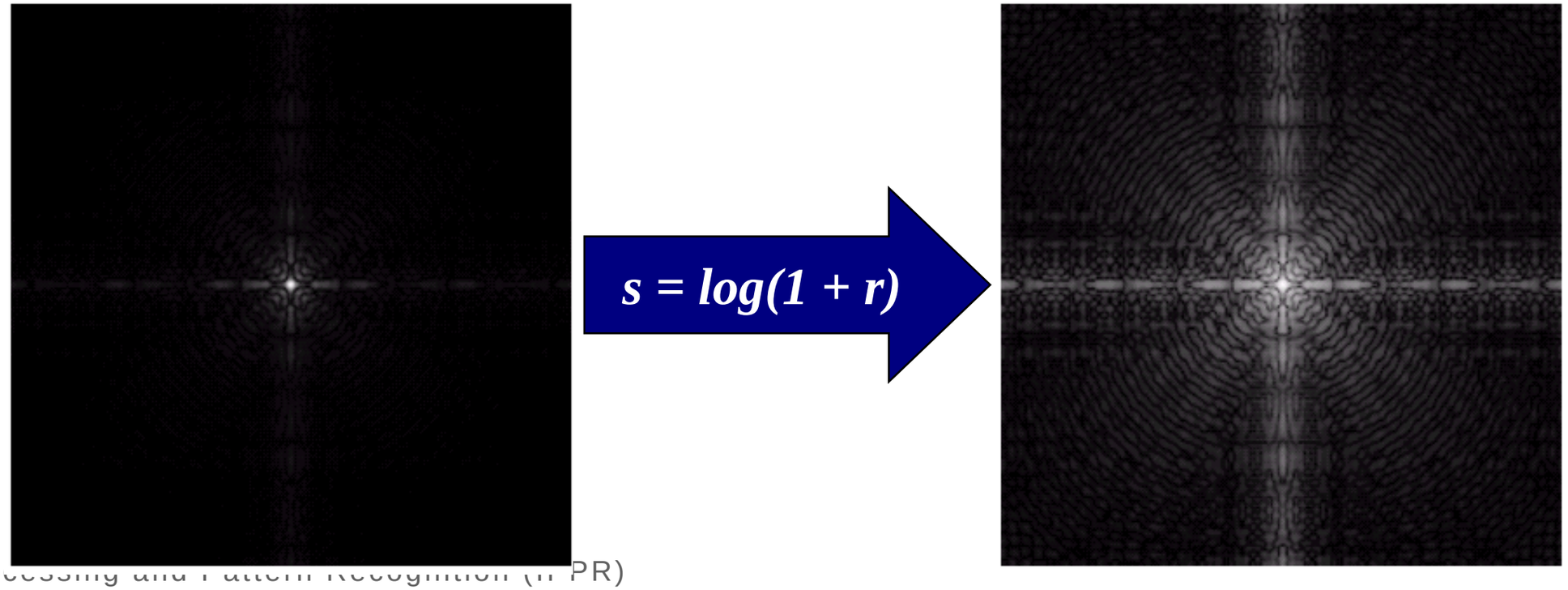
The inverse log transformation performs the opposite transformation

Compresses the dynamic range of images with large variations in pixel values

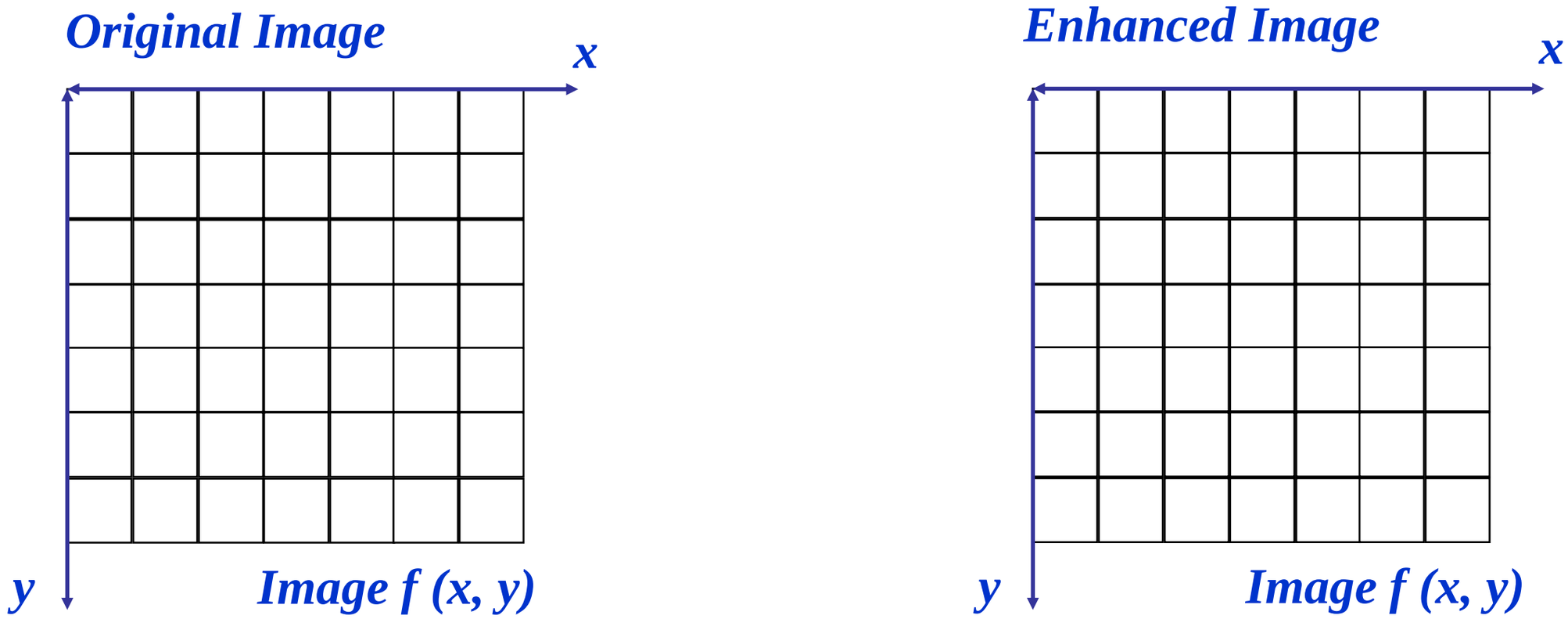
Logarithmic Transformations (cont…)

Log functions are particularly useful when the input grey level values may have an extremely large range of values

In the following example the Fourier transform of an image is put through a log transform to reveal more detail



Logarithmic Transformations (cont…)



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

We usually set c to 1

Grey levels must be in the range [0.0, 1.0]

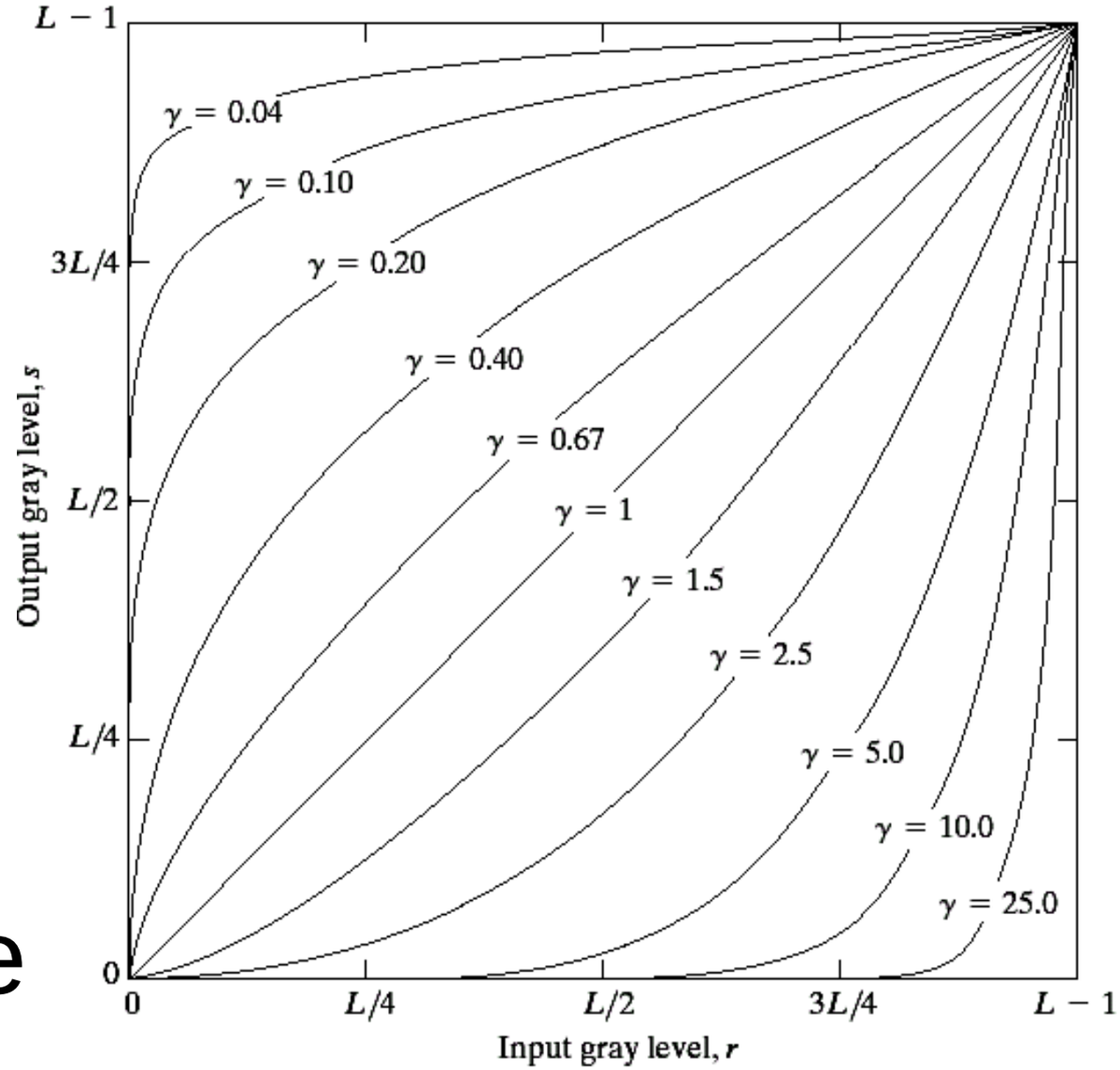
Power Law Transformations

Power law transformations have the following form

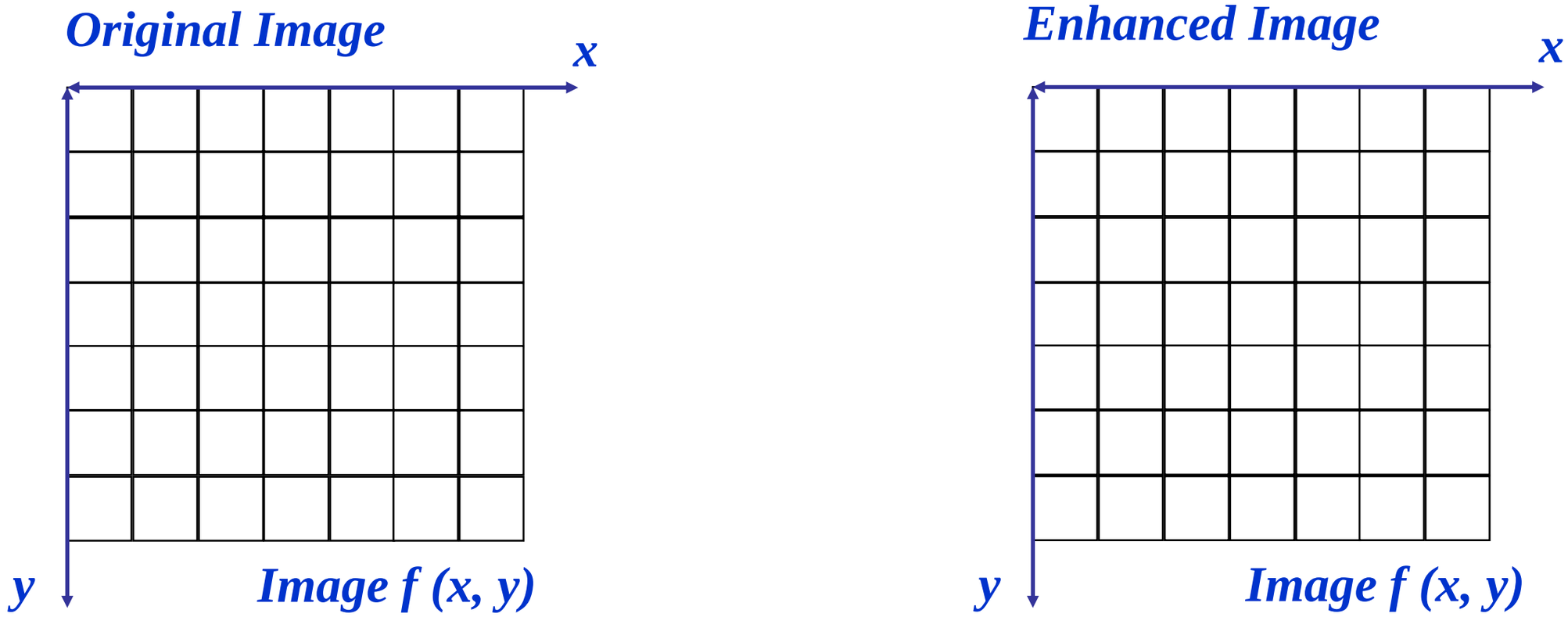
$$s = c * r^{\gamma}$$

Map a narrow range of dark input values into a wider range of output values or vice versa

Varying γ gives a whole family of curves



Power Law Transformations (cont…)



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

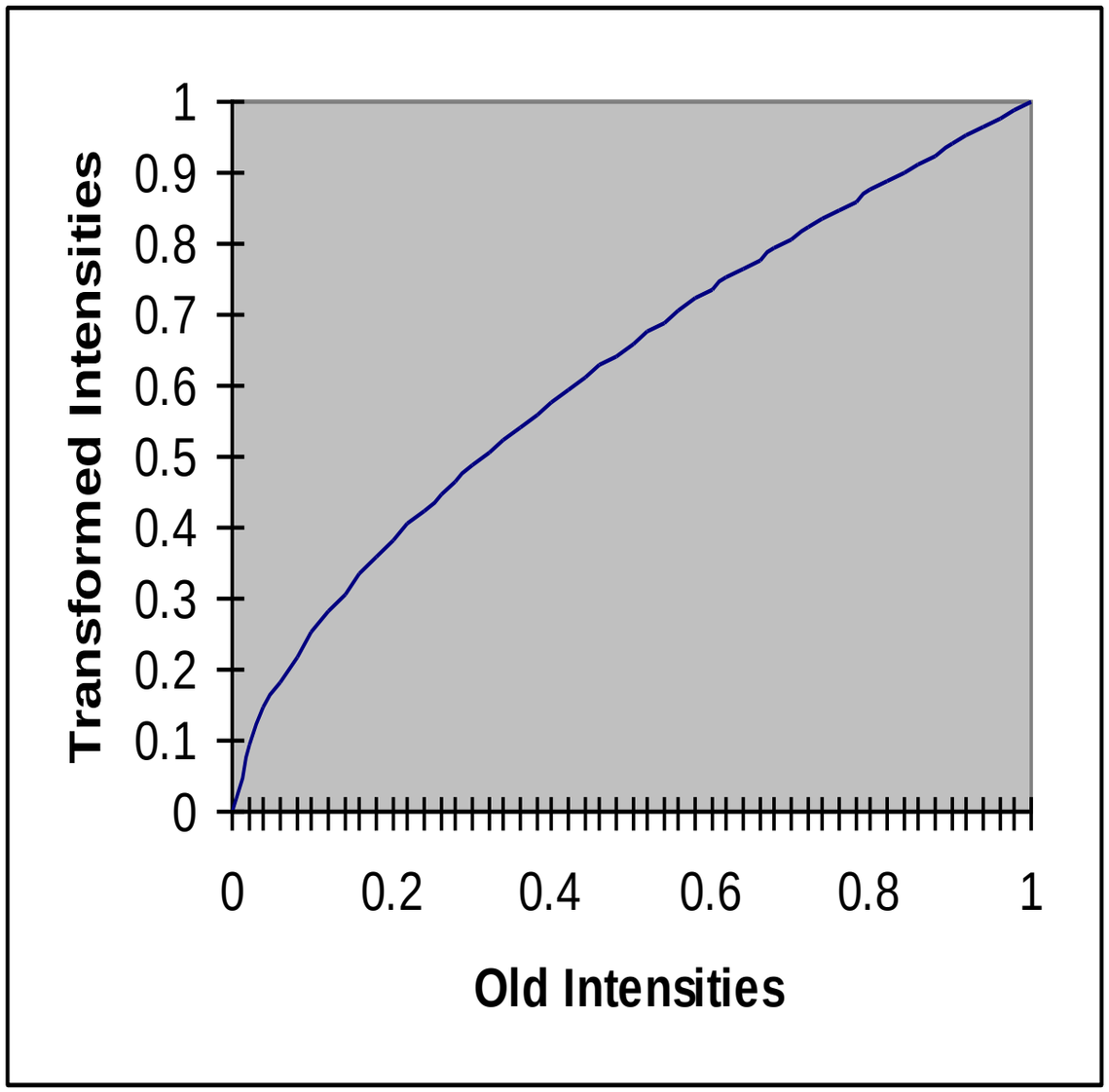
We usually set c to 1

Grey levels must be in the range [0.0, 1.0]

Power Law Example

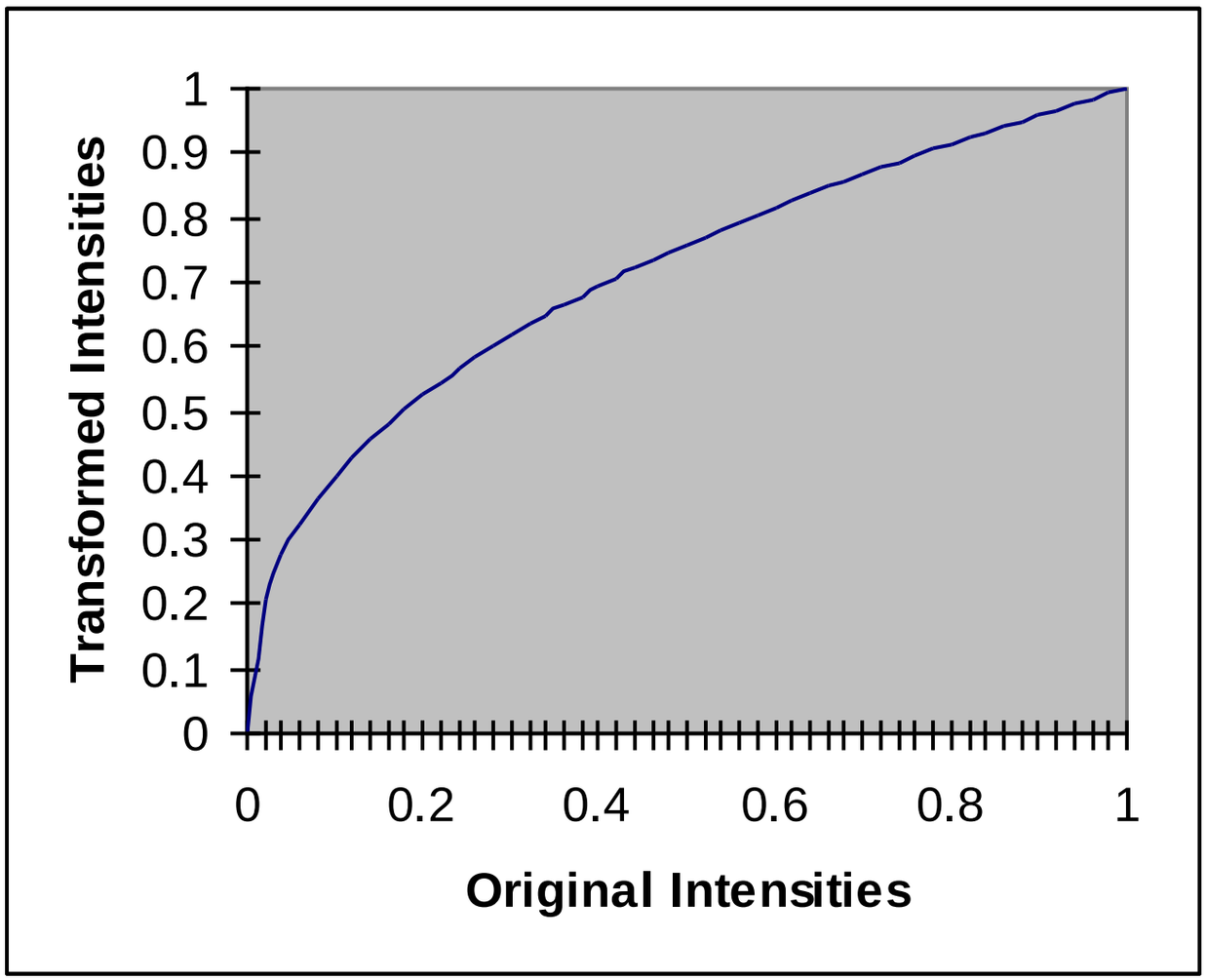
Power Law Example (cont…)

$$\gamma = 0.6$$



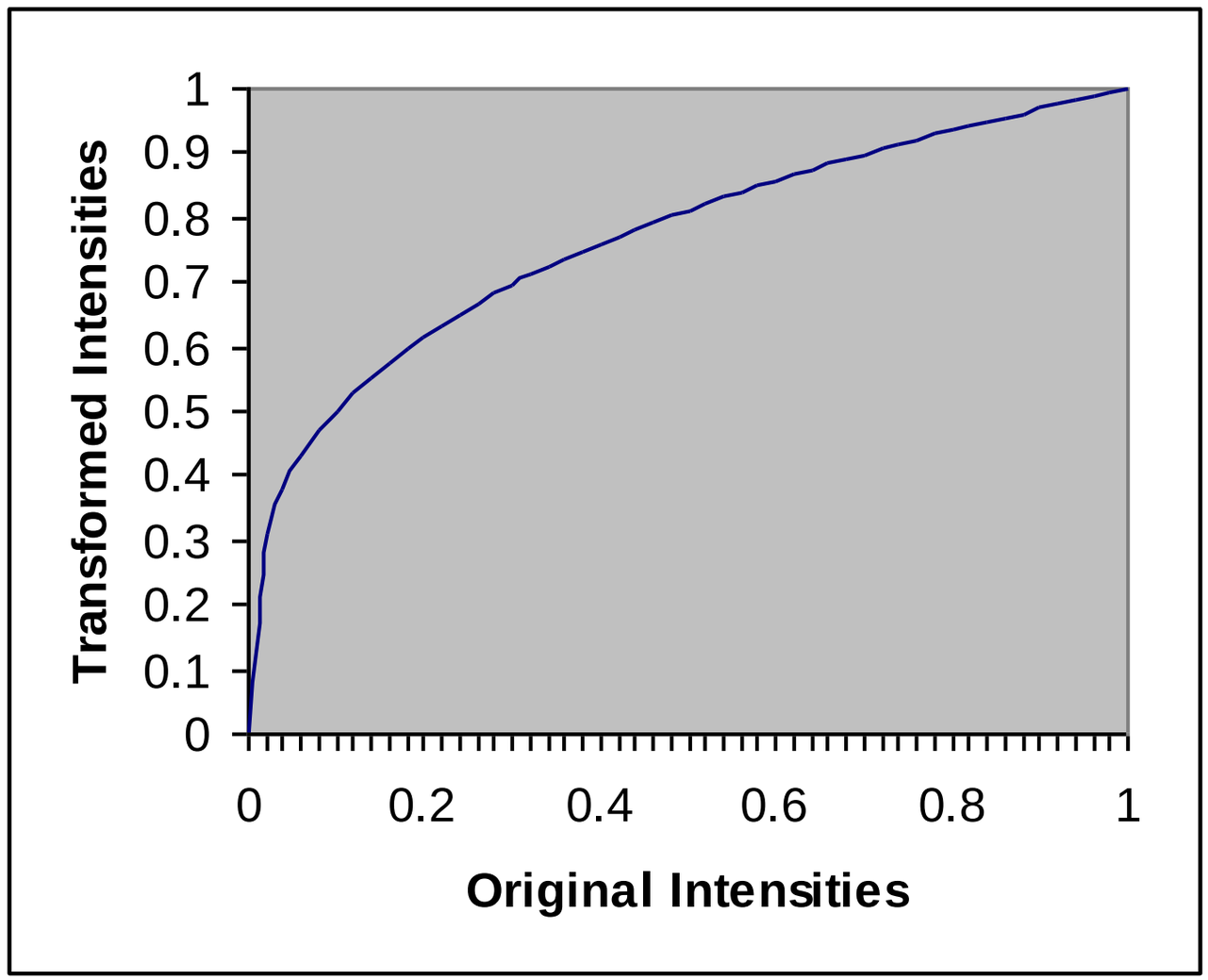
Power Law Example (cont…)

$$\gamma = 0.4$$



Power Law Example (cont…)

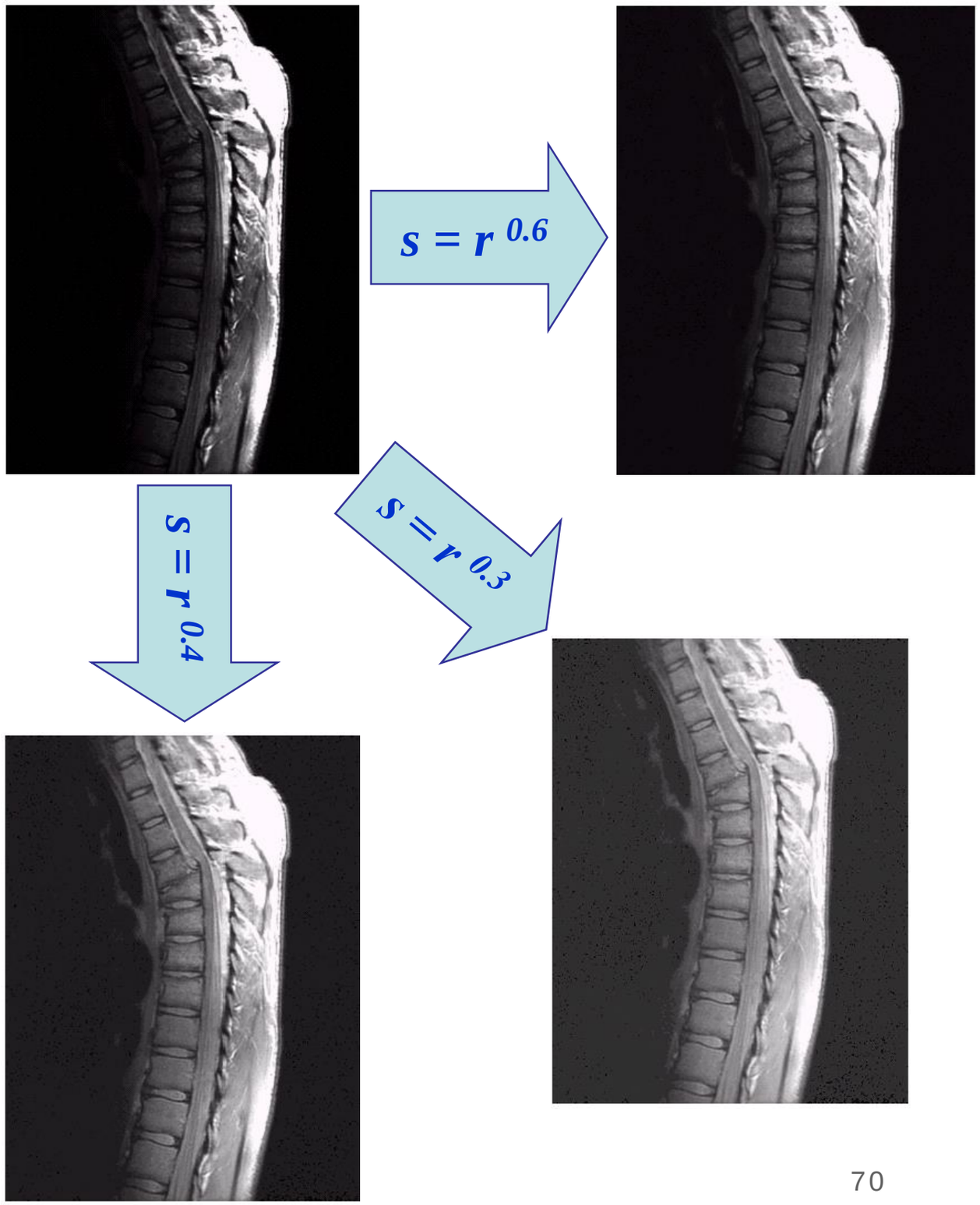
$\gamma = 0.3$



Power Law Example (cont…)

The images to the right show a magnetic resonance (MR) image of a fractured human spine

Different curves highlight different detail

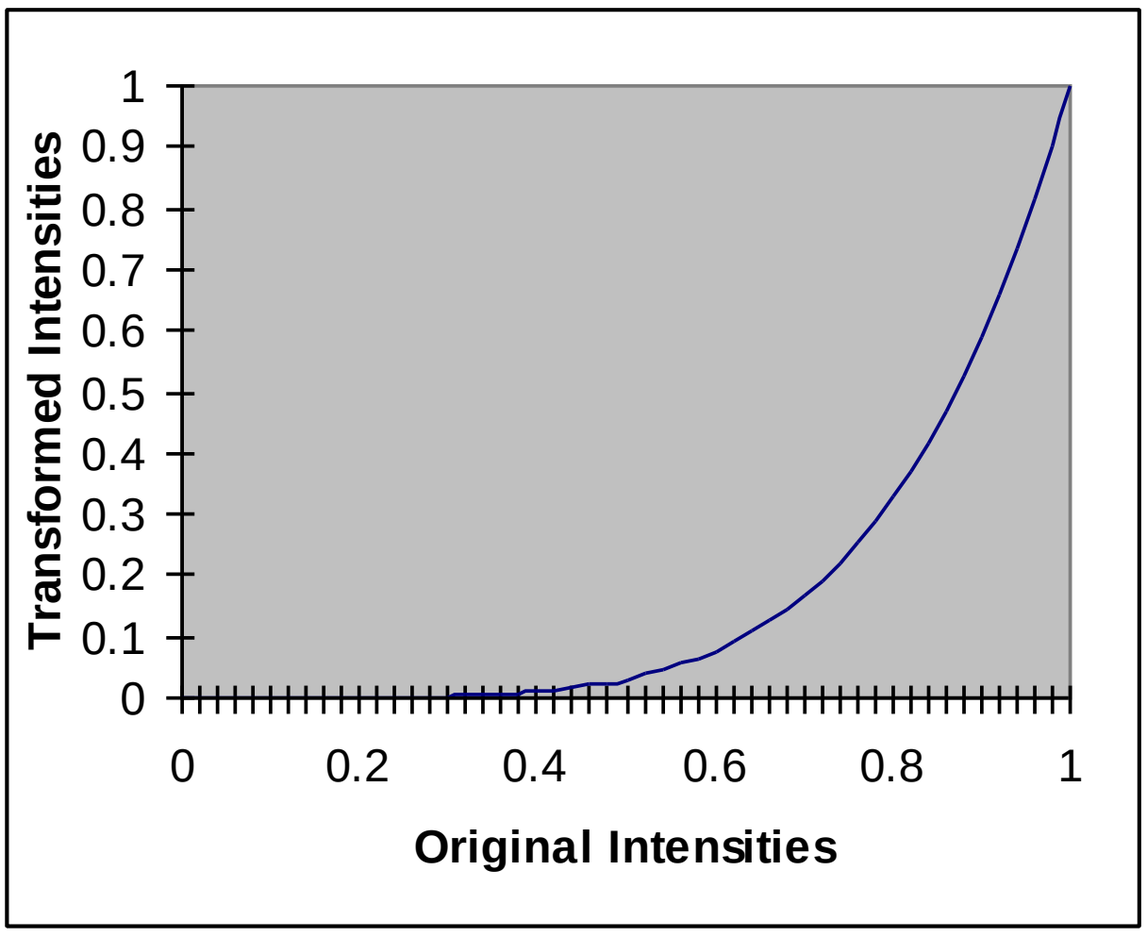


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Power Law Example

Power Law Example (cont…)

$$\gamma = 5.0$$

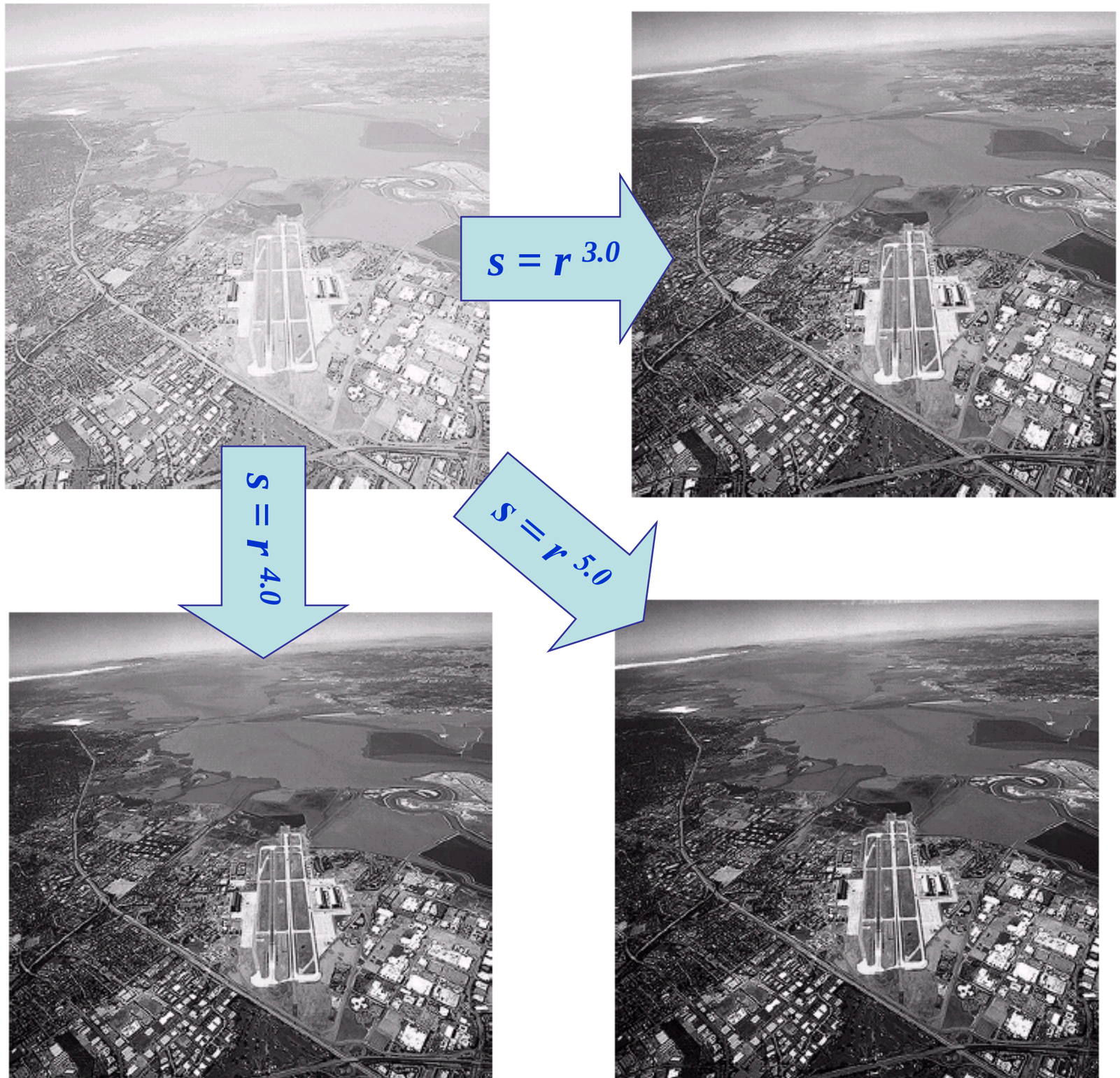


Power Law Transformations (cont…)

An aerial photo of a runway is shown

This time power law transforms are used to darken the image

Different curves highlight different detail



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Gamma Correction

Many devices used for image capture, display and printing respond according to a power law

• The exponent in the power-law equation is referred to as gamma

• The process of correcting for the power-law response is referred to as gamma correction

• Example: – CRT devices have an intensity-to-voltage response that is

a power function (exponents typically range from 1.8-2.5)

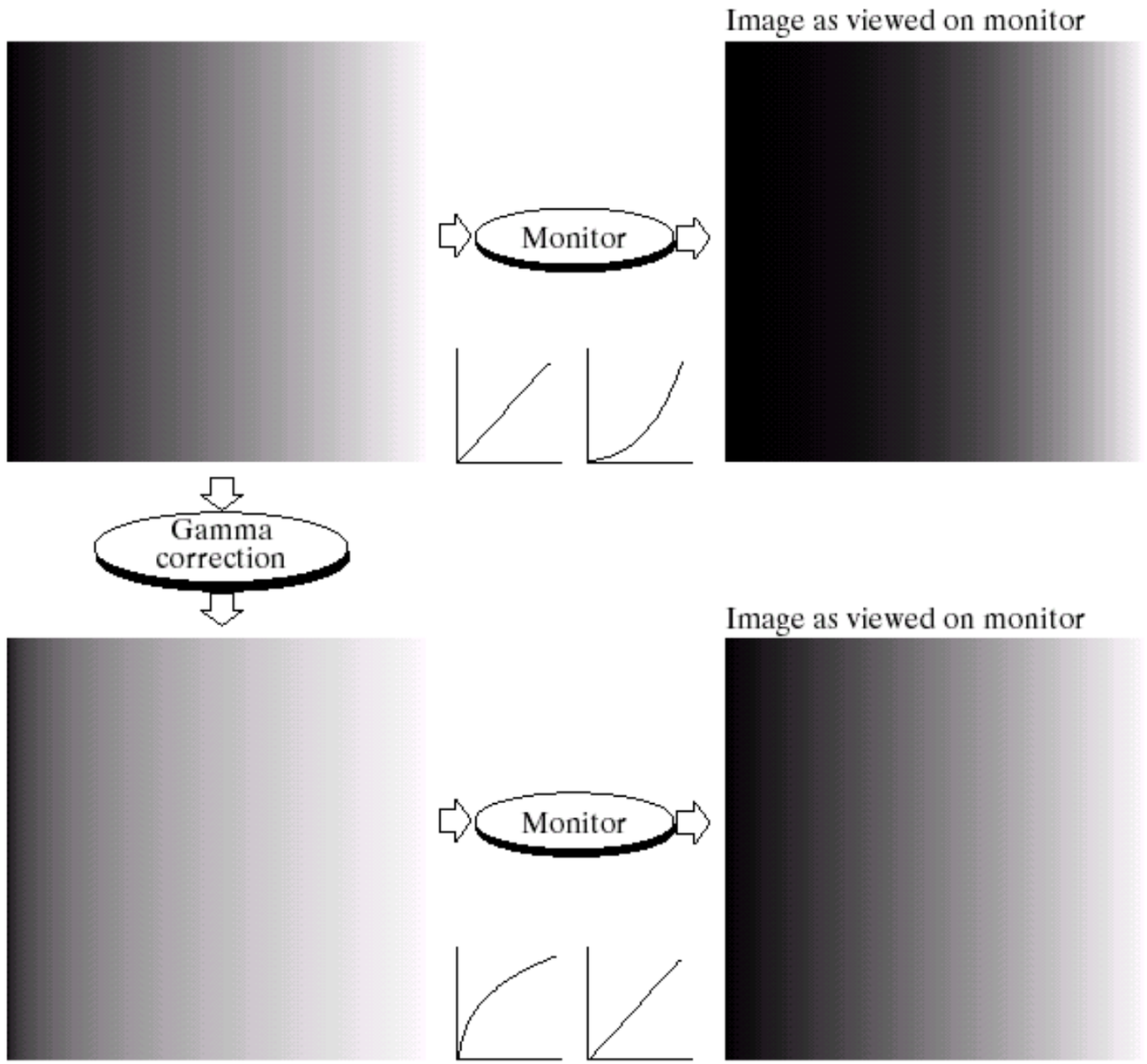
– Gamma correction in this case could be achieved by applying the transformation $s=r_1/2.5=r^{0.4}$

Gamma Correction

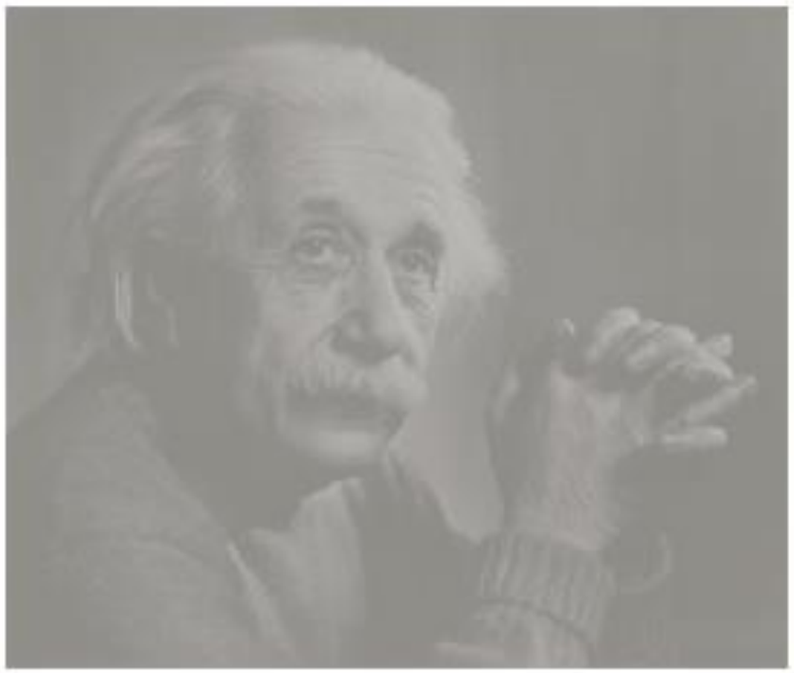
Many of you might be familiar with gamma correction of computer monitors

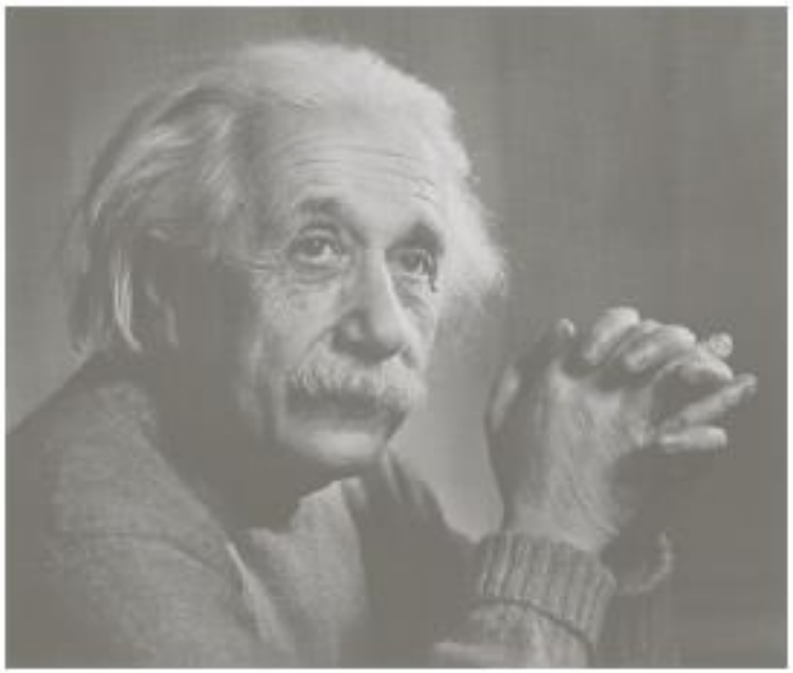
Problem is that display devices do not respond linearly to different intensities

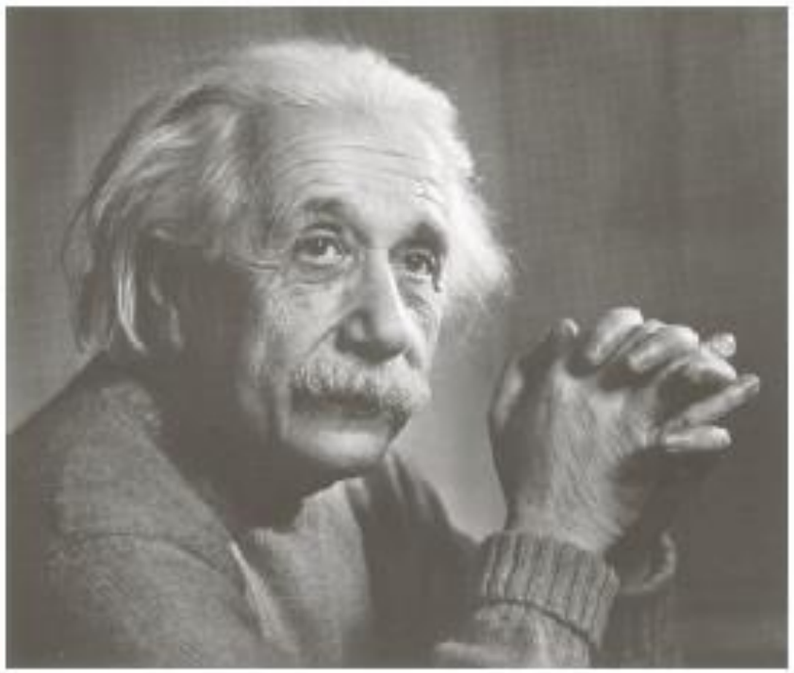
Can be corrected using a log transform



More Contrast Issues





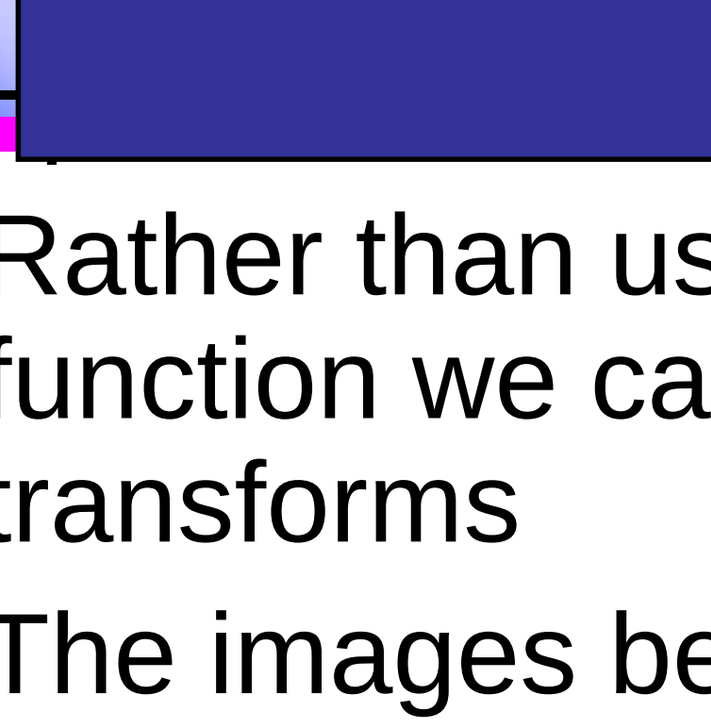


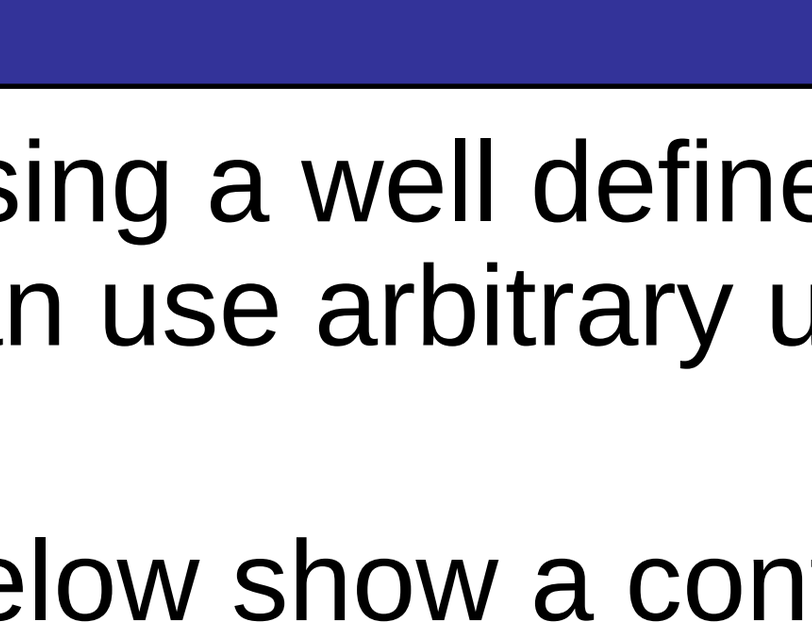
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

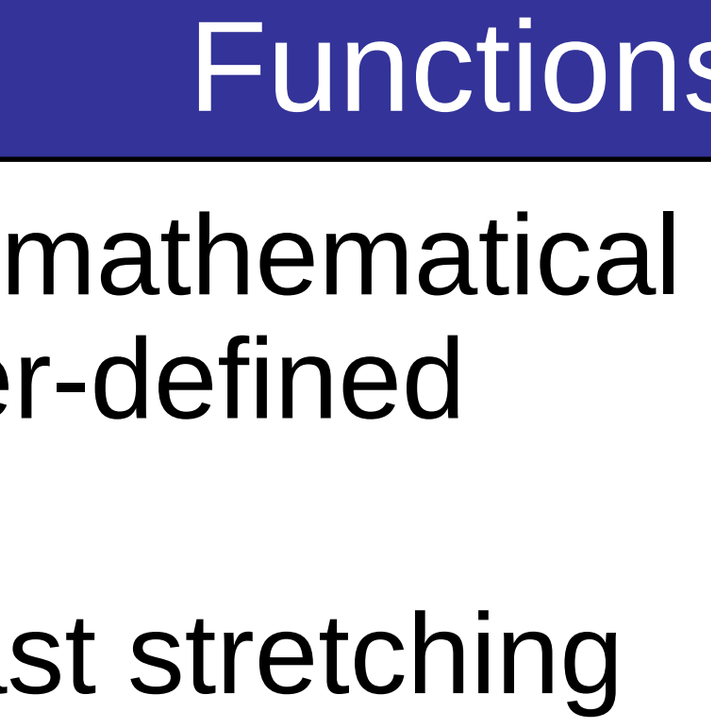
Piecewise Linear Transformation Functions

Rather than using a well defined mathematical function we can use arbitrary user-defined transforms

The images below show a contrast stretching linear transform to add contrast to a poor quality image







Piecewise Linear Transformation Functions

- Rather than using a well defined mathematical function we can use arbitrary user-defined transforms
- Contrast stretching expands the range of intensity levels in an image so it spans a given (full) intensity range
- Control points (r1,s1) and (r2,s2) control the shape of the transform T(r)
- $r_1=r_2$, $s_1=0$ and $s_2=L-1$ yields a thresholding function

The contrast stretched image shown in the previous slide is obtained using the transformation obtained from the equation of the line having following points

- $(r_1,s_1)=(r_{min},0)$ and $(r_2, s_2)=(r_{max},L-1)$

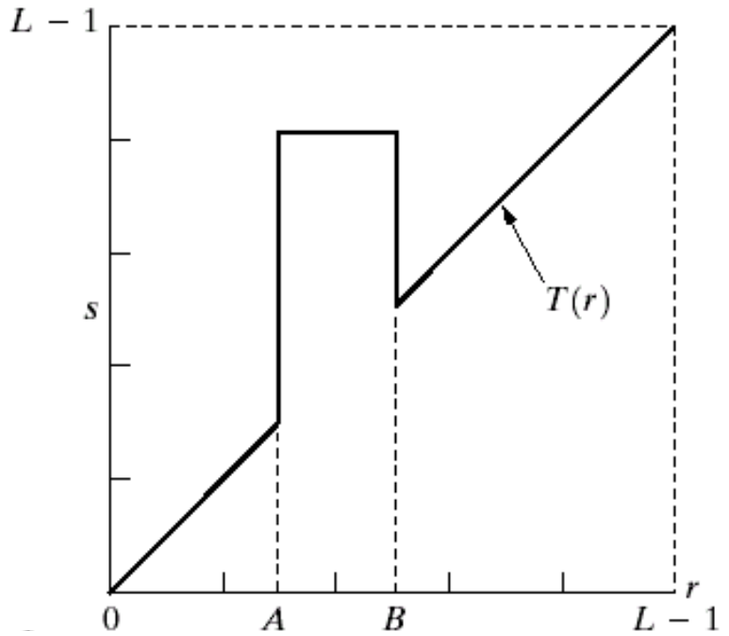
Gray Level Slicing

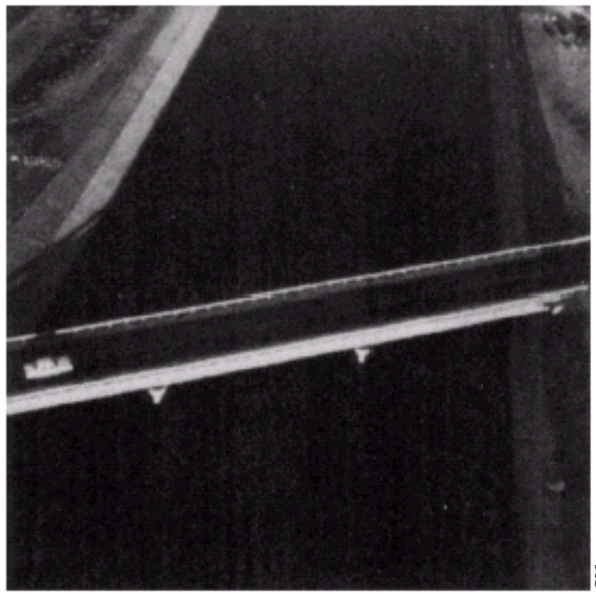
- Used to highlight a specific range of intensities in an image that might be of interest
- Two common approaches
  - Set all pixel values within a range of interest to one value (white) and all others to another value (black)
    - Produces a binary image
  - Brighten (or darken) pixel values in a range of interest and leave all others unchanged

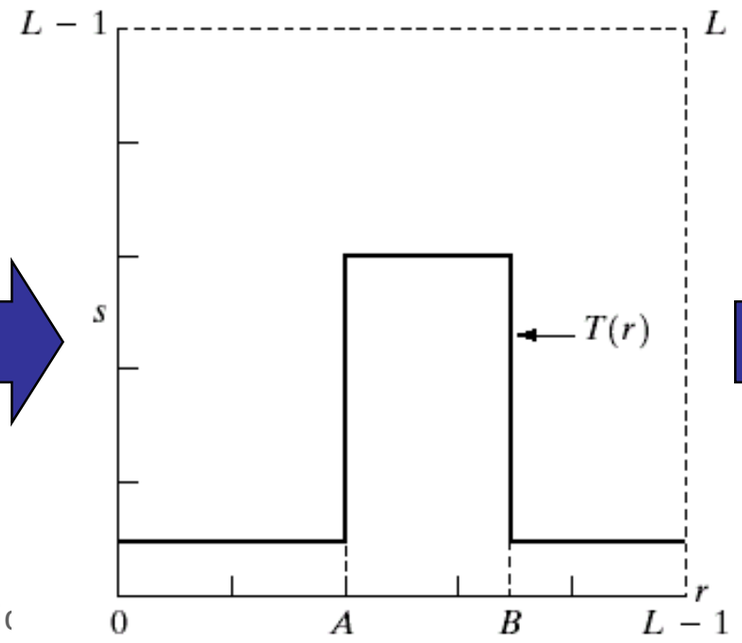
Gray Level Slicing

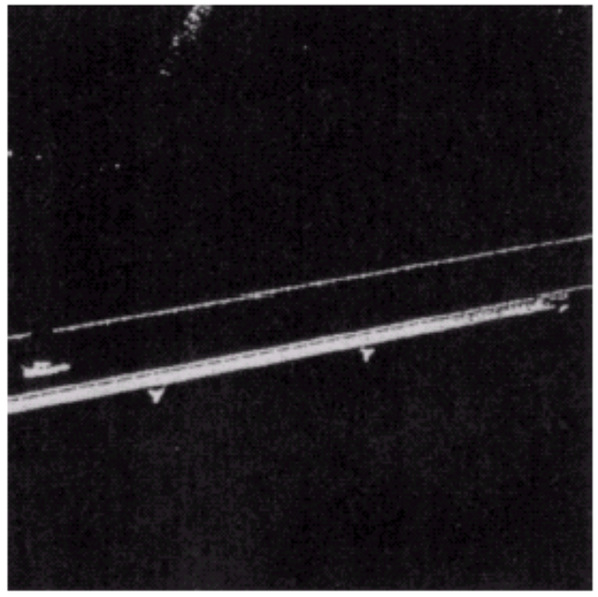
Highlights a specific range of grey levels

- Similar to thresholding
- Other levels can be suppressed or maintained
- Useful for highlighting features in an image





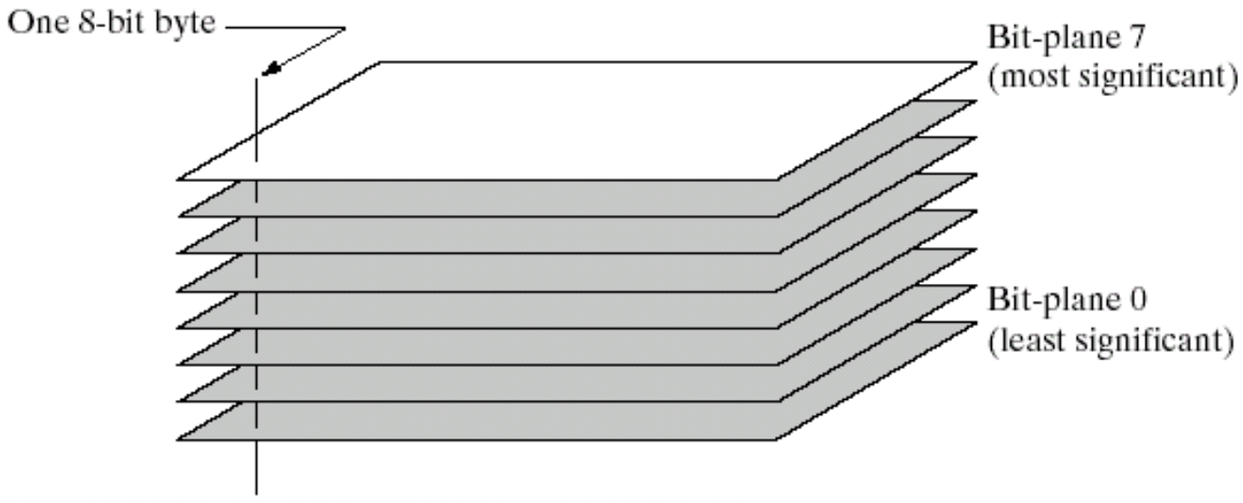


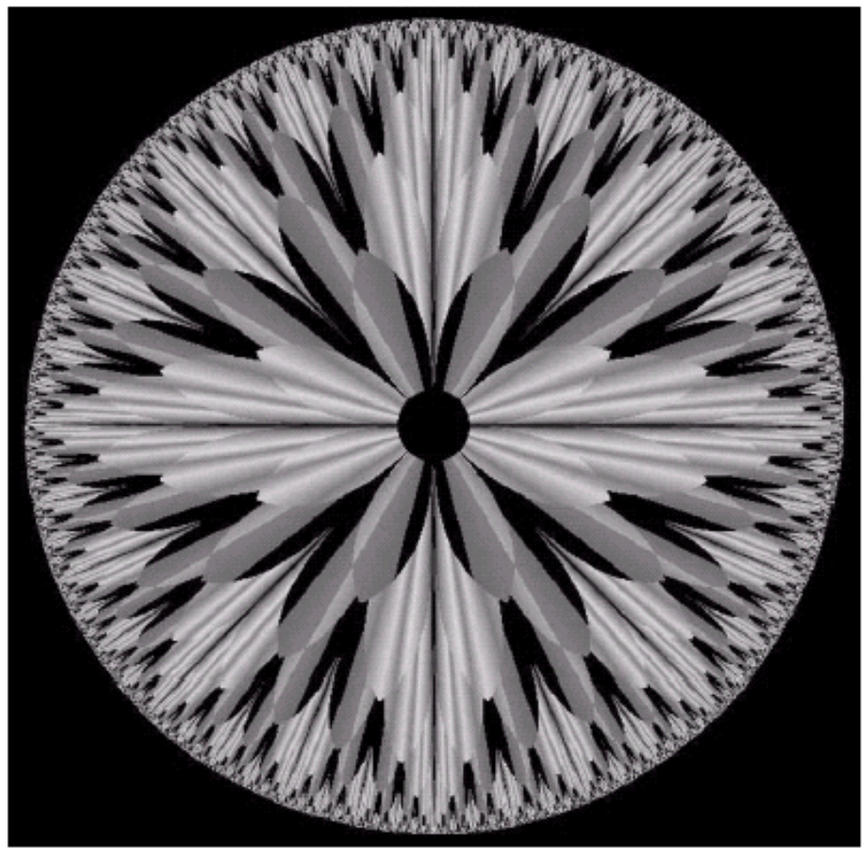


Bit Plane Slicing

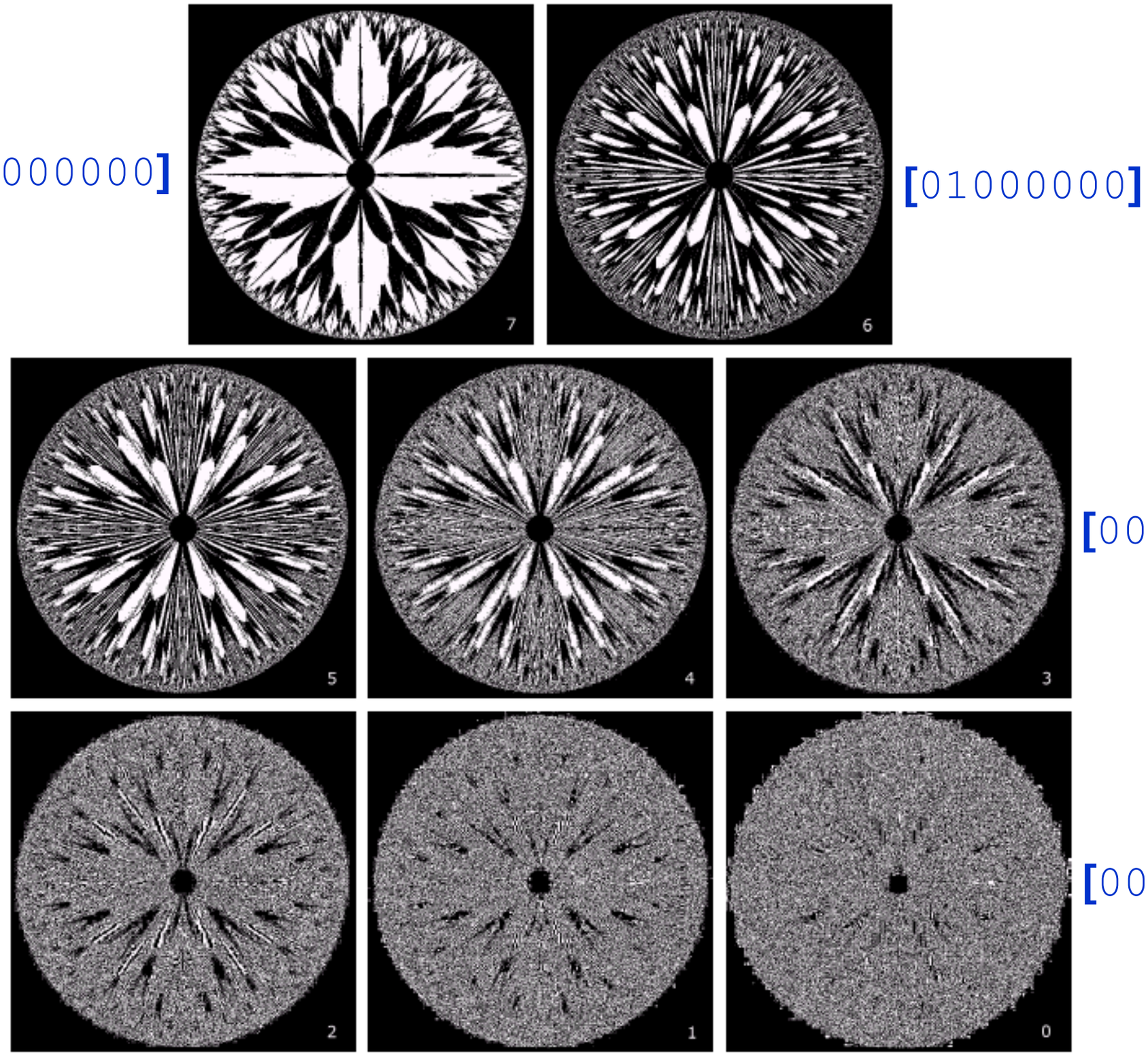
Often by isolating particular bits of the pixel values in an image we can highlight interesting aspects of that image

- Higher-order bits usually contain most of the significant visual information
- Lower-order bits contain subtle details

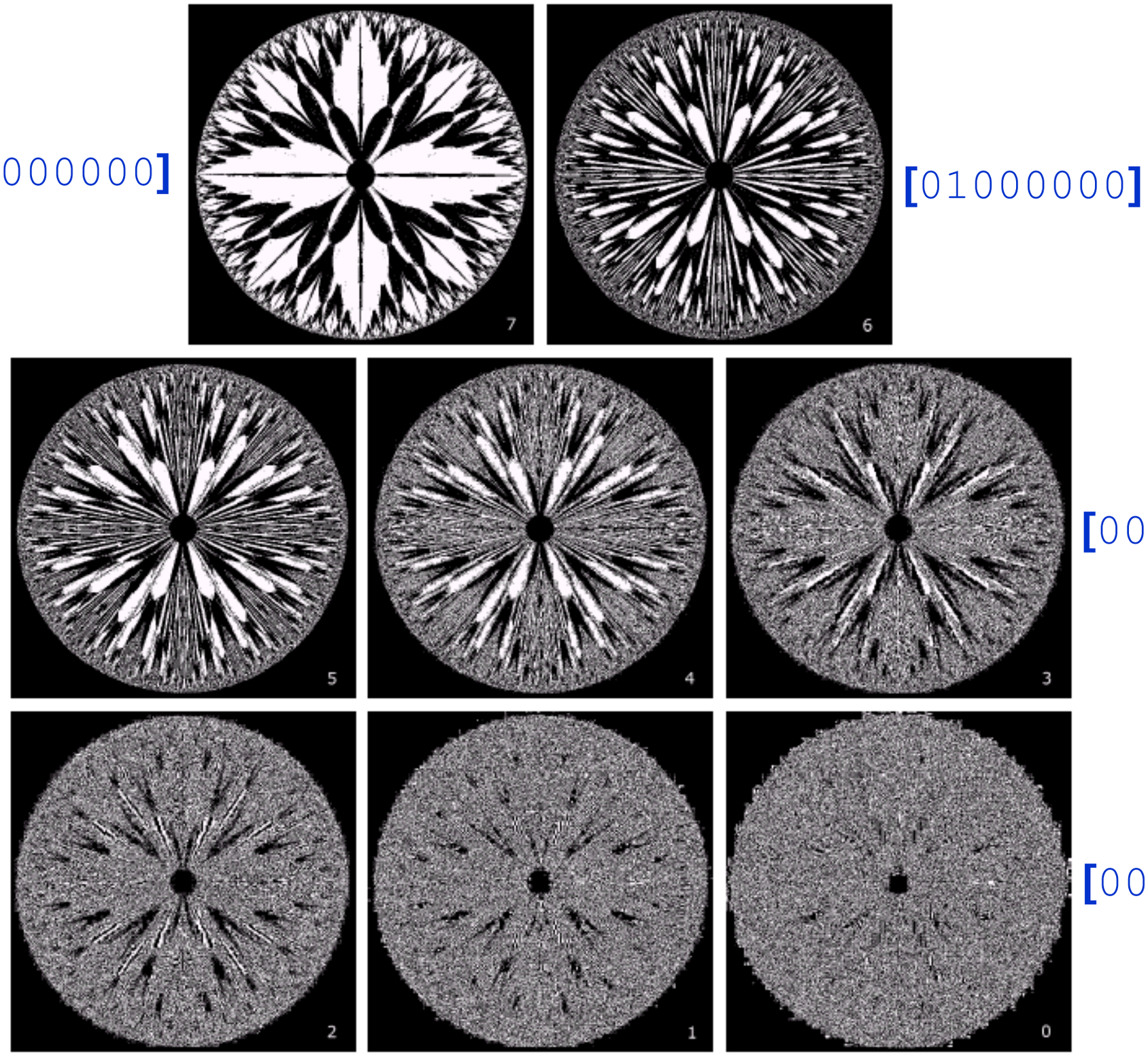




Bit Plane Slicing (cont…)



Bit Plane Slicing (cont…)



Bit Plane Slicing (cont…)

a b c
d e f
g h i

FIGURE 3.14 (a) An 8-bit gray-scale image of size 500 × 1192 pixels. (b) through (i) Bit planes 1 through 8, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image.

Bit Plane Slicing (cont…)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

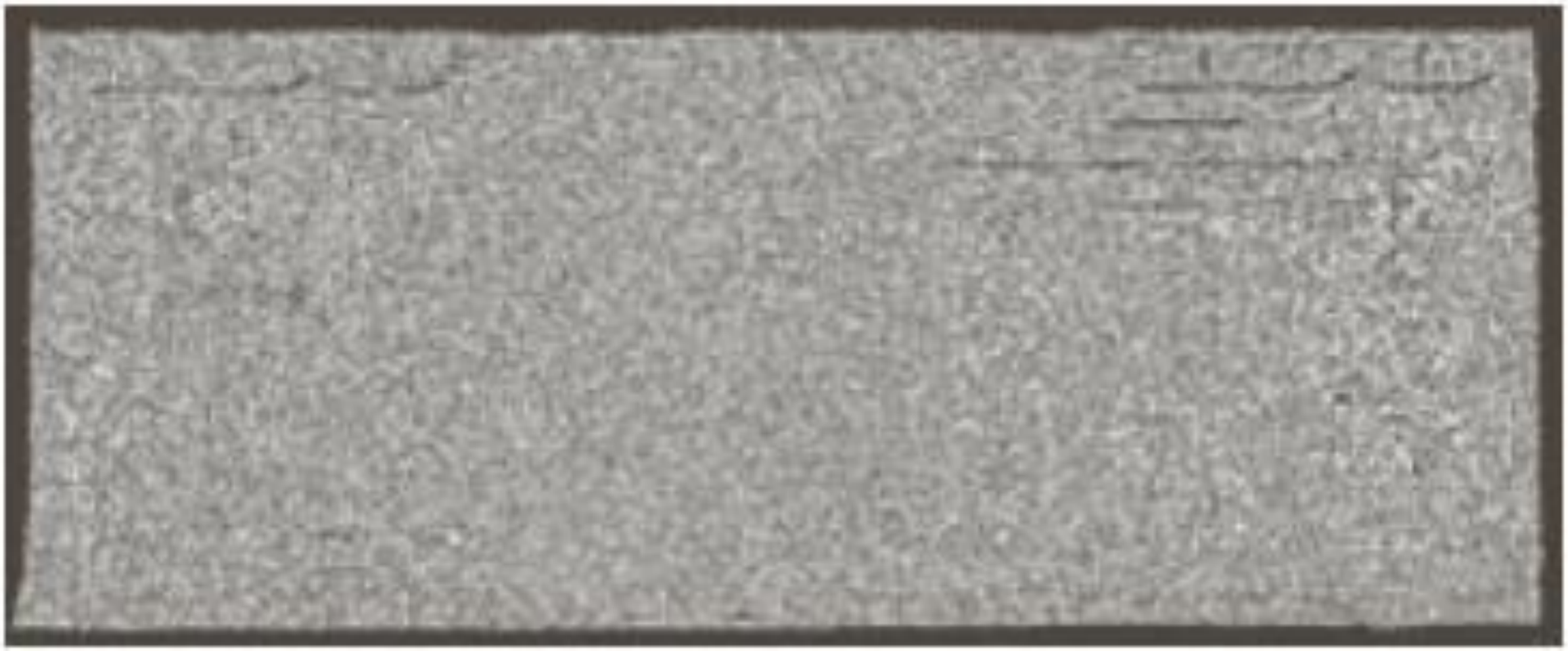
Bit Plane Slicing (cont…)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)

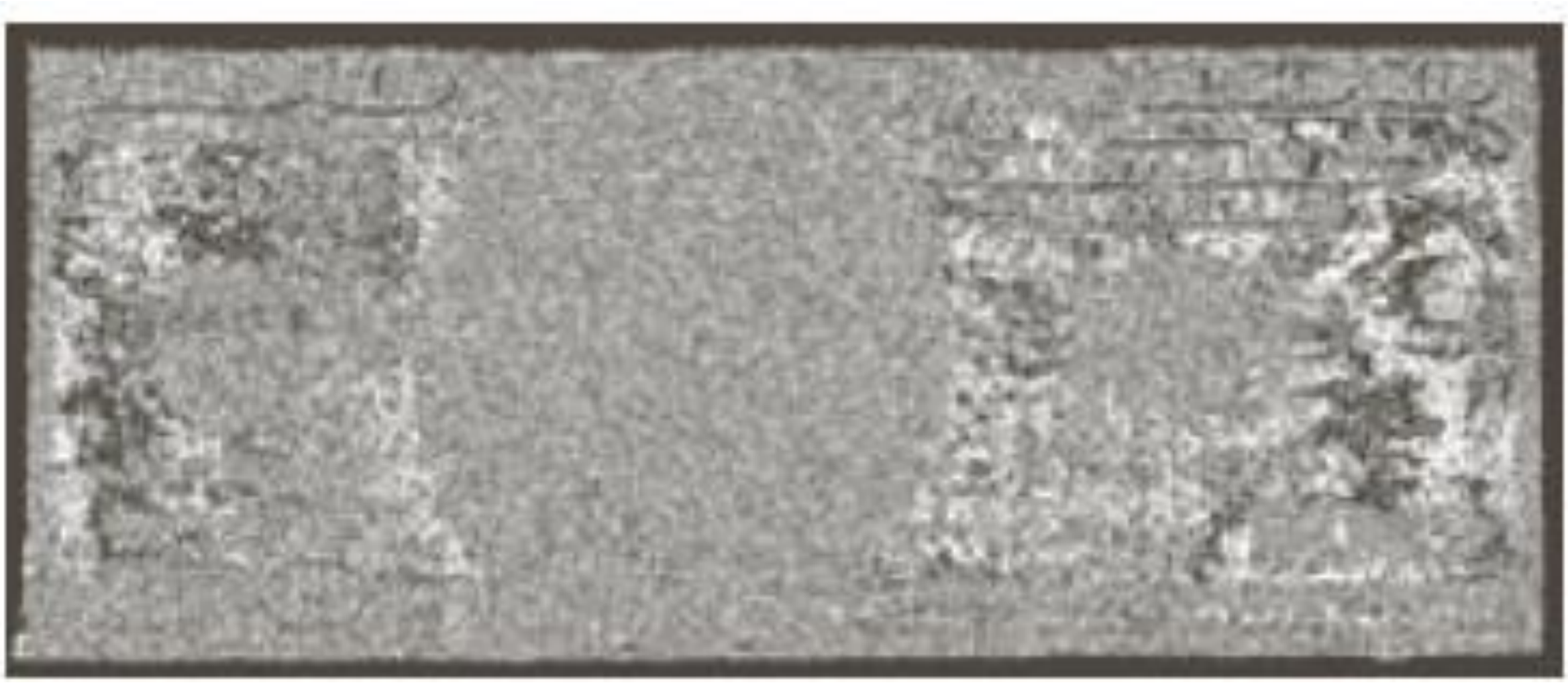
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

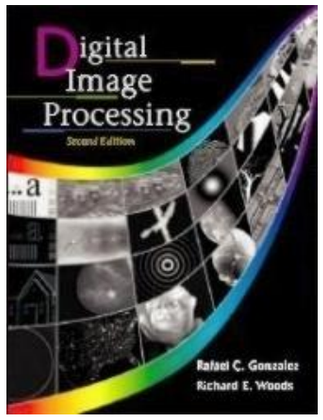
Bit Plane Slicing (cont…)

Reconstructed image using only bit planes 8 and 7

Reconstructed image using only bit planes 8, 7 and 6

Reconstructed image using only bit planes 7, 6 and 5

References



"Digital Image Processing", Rafael C. Gonzalez & Richard E. Woods, Addison-Wesley, 2002

–Much of the material that follows is taken from this book

–Image Processing and Pattern Recognition Slides of Dr. Sanjeeb Prasad Panday

Thank you !!!