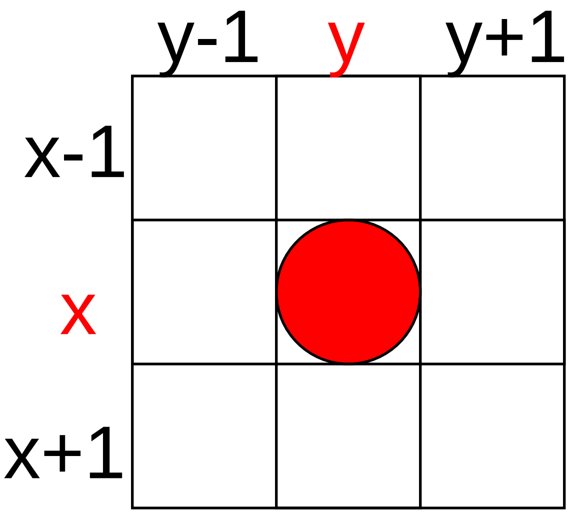
Digital Image Processing

Lecture # 3
Fundamentals & Spatial Enhancement

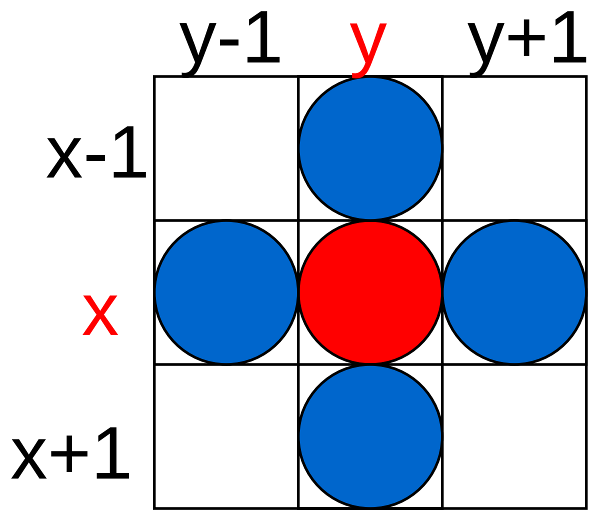
Relationships between pixels

- Neighbors of pixel are the pixels that are adjacent pixels of an identified pixel



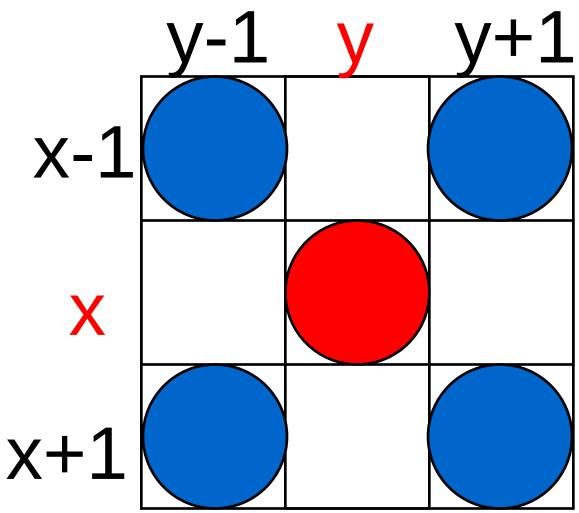
4- Neighbors of a Pixel –$N_4(p)$

What are the coordinates of each of the blue pixels



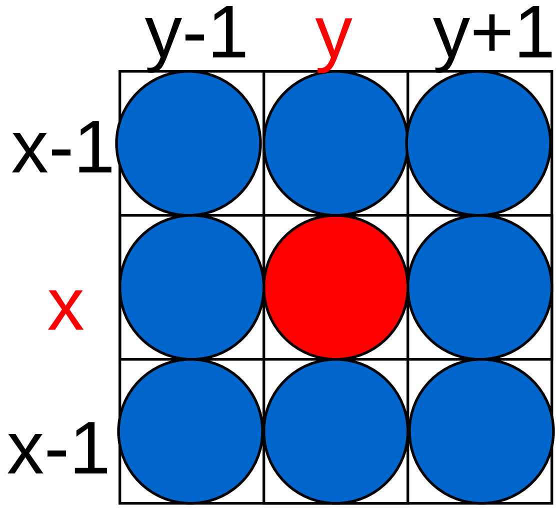
(x-1,y), (x+1,y), (x, y-1), (x, y+1)

Diagonal Neighbors of a Pixel – $N_D(p)$



(x-1,y-1), (x+1,y-1), (x-1, y+1), (x+1, y+1)

8- Neighbors of a Pixel –N_8(p)

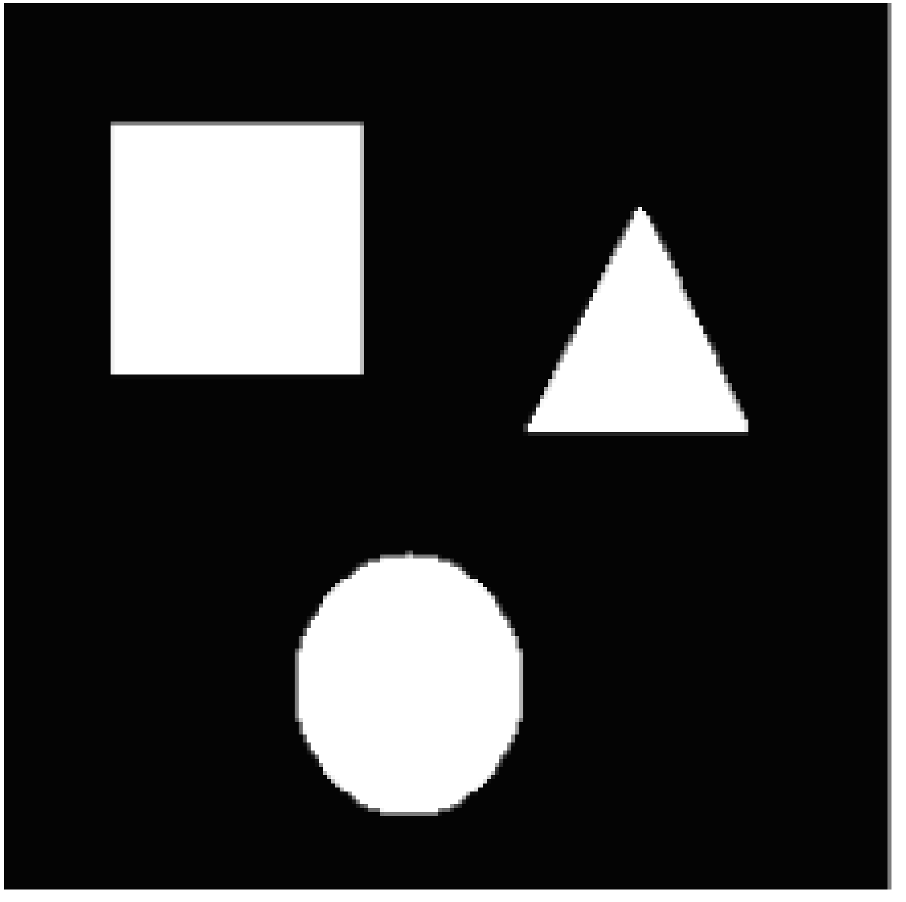


$$N_8(p) = N_4(p) \cup N_D(p)$$

(x-1,y), (x+1,y), (x, y-1), (x, y+1)

(x-1,y-1), (x+1,y-1), (x-1, y+1), (x+1, y+1)

Determine different regions in the image



Connectivity

- Establishing boundaries of objects and components in an image
- Group the same region by assumption that the pixels being the same color or equal intensity
- Two pixels p & q are connected if
  - *They are adjacent in some sense*
  - *If their gray levels satisfy a specified criterion of similarity*

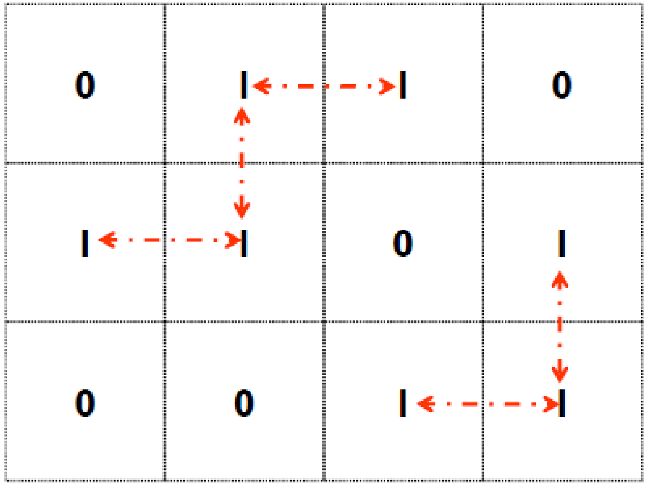
Connectivity

V: Set of gray levels used to define the criterion of similarity

4-connectivity

If gray level $(p, q) \in V, and\ q \in N_4(p)$

Set of gray levels V = {1}



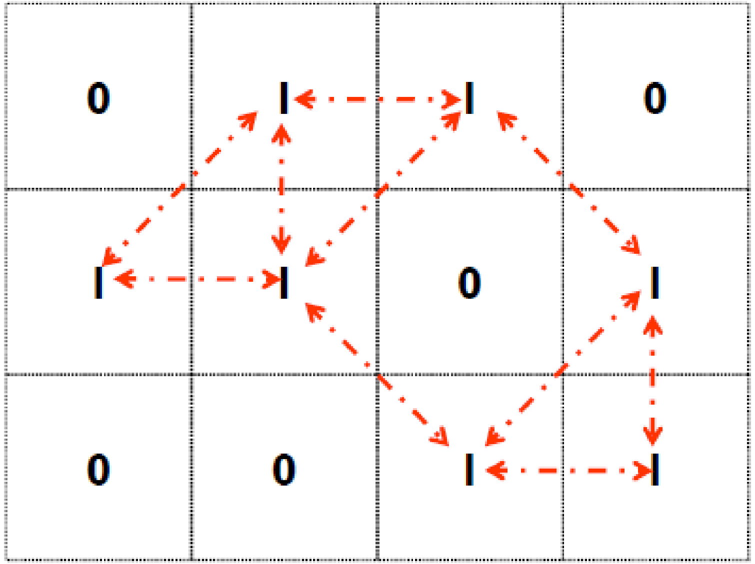
Connectivity

V: Set of gray levels used to define the criterion of similarity

8-connectivity

If gray level $(p,q) \in V, and\ q \in N_8(p)$

Set of gray levels V = {1}



Connectivity

V: Set of gray levels used to define the criterion of similarity

m-connectivity (Mixed Connectivity)

If gray level

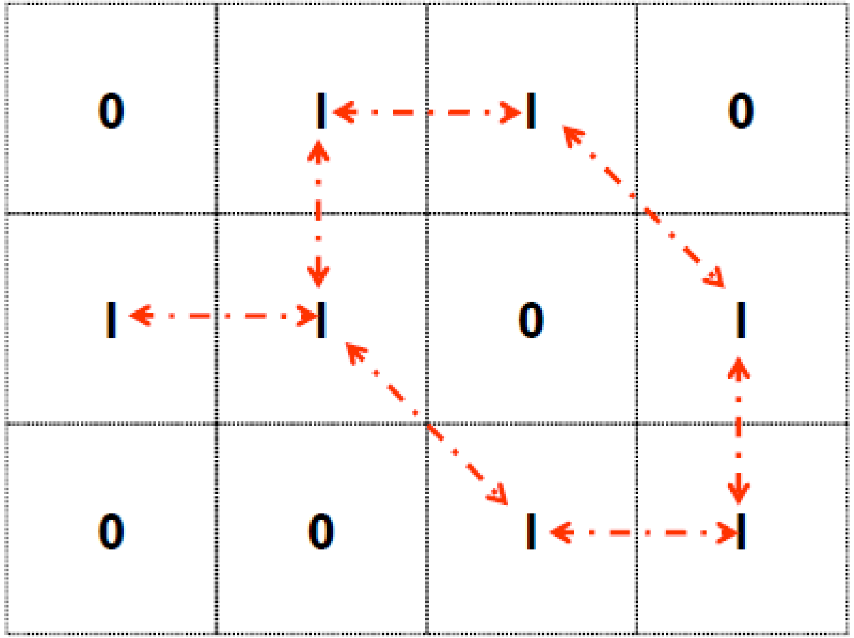
$(p, q) \in V$, $and\ q$ satisfies one of the following:

a. $q \in N_4(p)\ or$

b. $q \in N_D(p)\ And\ N_4(p) \cap N_4(q)$ has no pixels whose values are from V

Example: m – Connectivity

- Set of gray levels V = {1}



Note: Mixed connectivity can eliminate the multiple path connections that often occurs in 8-connectivity

Paths

- Path: Let coordinates of pixel p: (x, y), and of pixel q: (s, t)
- A *path* from p to q is a sequence of distinct pixels with coordinates: $(x_0, y_0), (x_1, y_1), \ldots\ldots, (x_n, y_n)$
  where $(x_0, y_0) = (x, y)$ & $(x_n, y_n) = (s, t)$, and (x_i, y_i) is adjacent to (x_{i-1}, y_{i-1}) $1 \le i \le n$

Test Yourself

Consider the image segment shown.

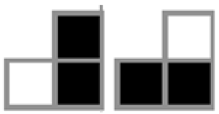
(a) Let $V = \{0, 1\}$ and compute the lengths of the shortest 4-, 8-, and m-path between p and q. If a particular path does not exist between these two points, explain why.

(b) Repeat for $V = \{1, 2\}$.

	3	1	2	1 (q)
	2	2	0	2
	1	2	1	1
(p)	1	0	1	2

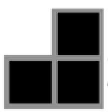
CC labeling – 4 Connectivity

- Process the image from left to right, top to bottom:

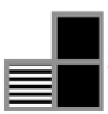


  1.) If the next pixel to process is 1

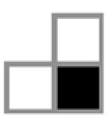
  i.) If only one of its neighbors (top or left) is 1, copy its label.



  ii.) If both are 1 and have the same label, copy it.

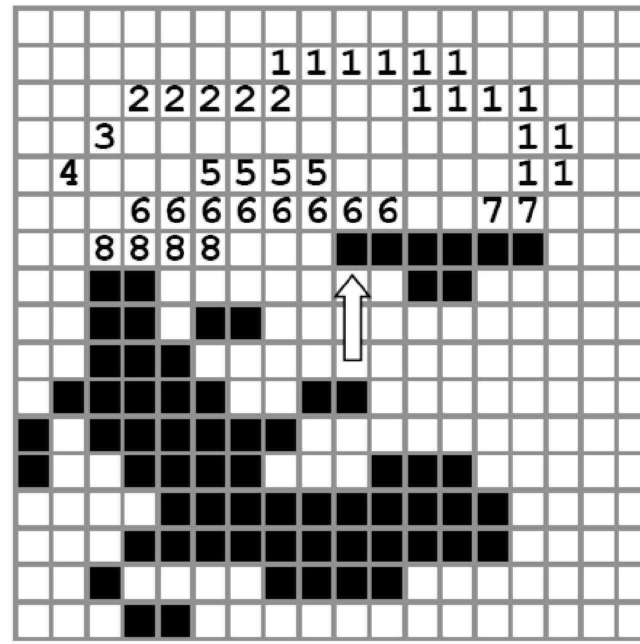


  iii.) If they have different labels
  - Copy the label from the left.
  - Update the equivalence table.



  iv.) Otherwise, assign a new label.

Pass 1

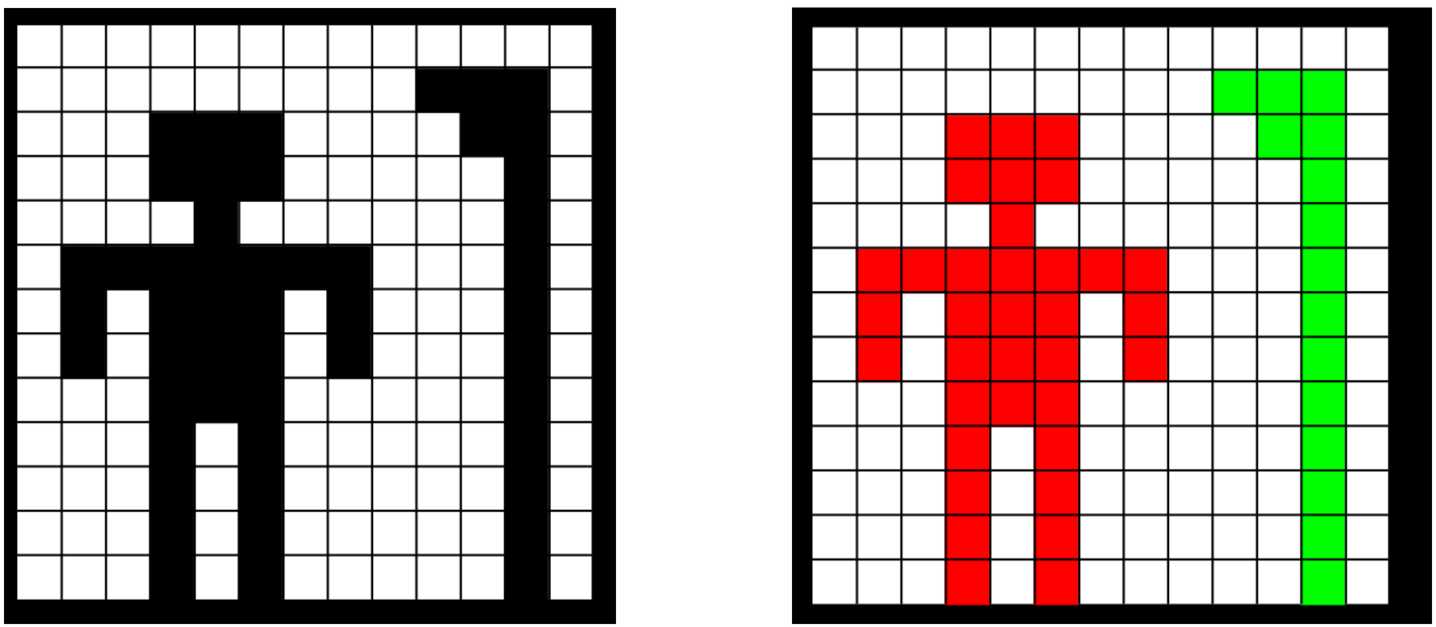



```
{1, 2, 7}
{3}
{4}
{5, 6, 8}
{}
```

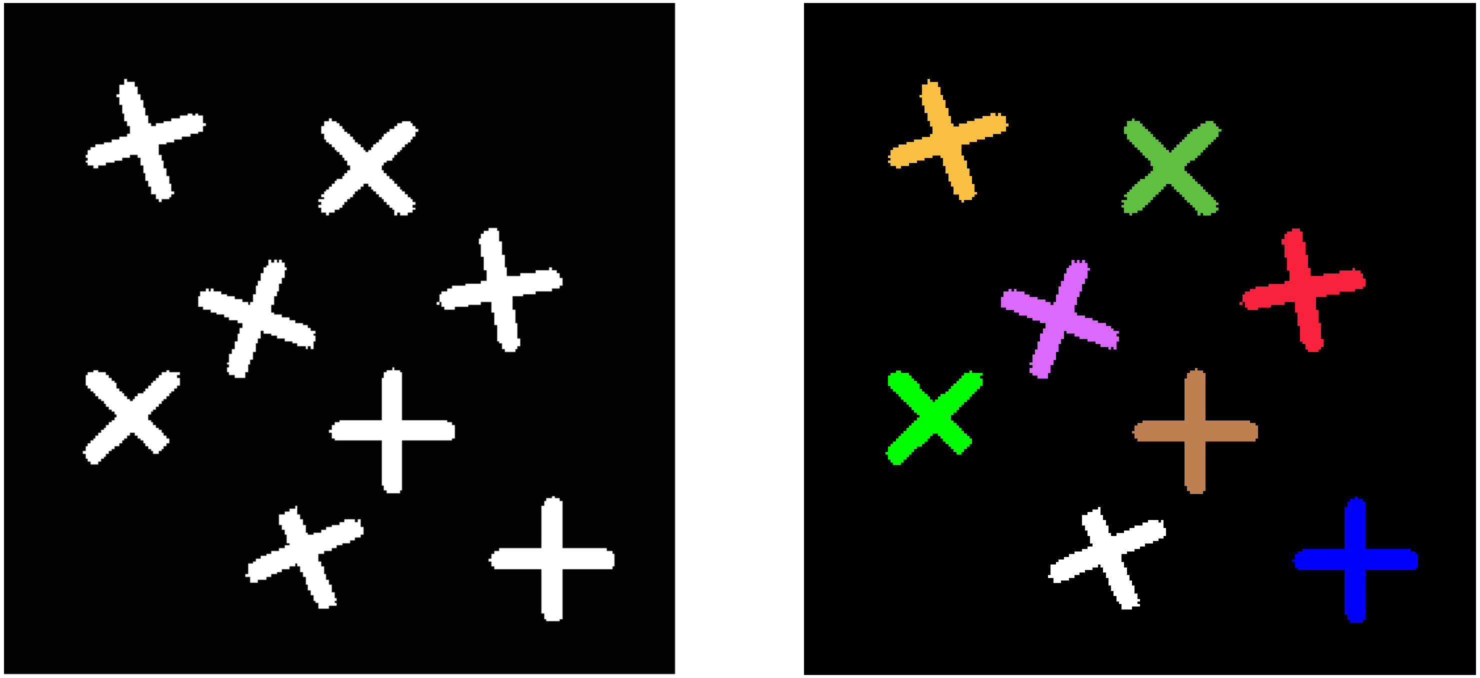
- Re-label with the smallest of equivalent labels

Pass 2

CC labeling – 4 Connectivity



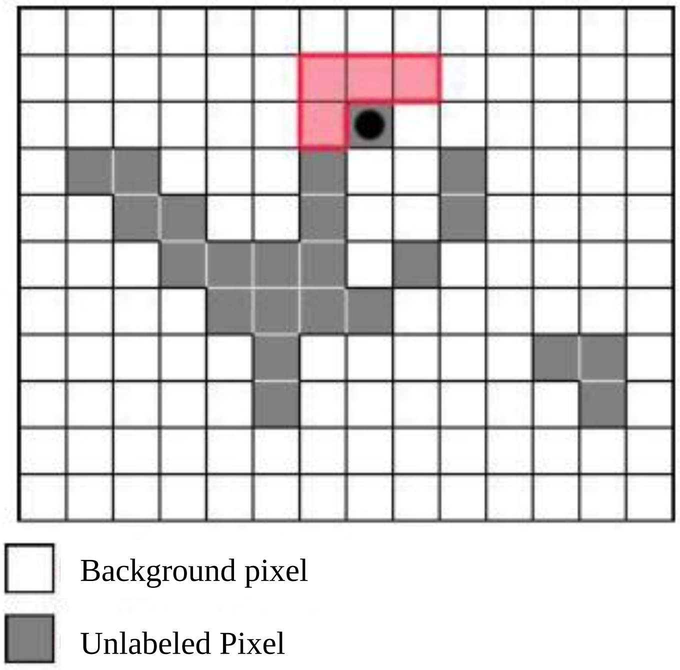
CC labeling – 4 Connectivity

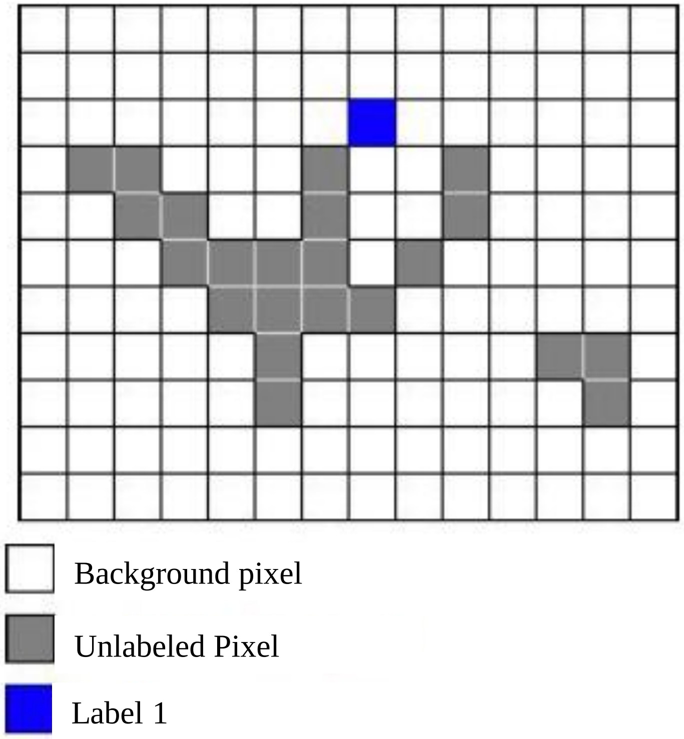


CC labeling – 8 Connectivity

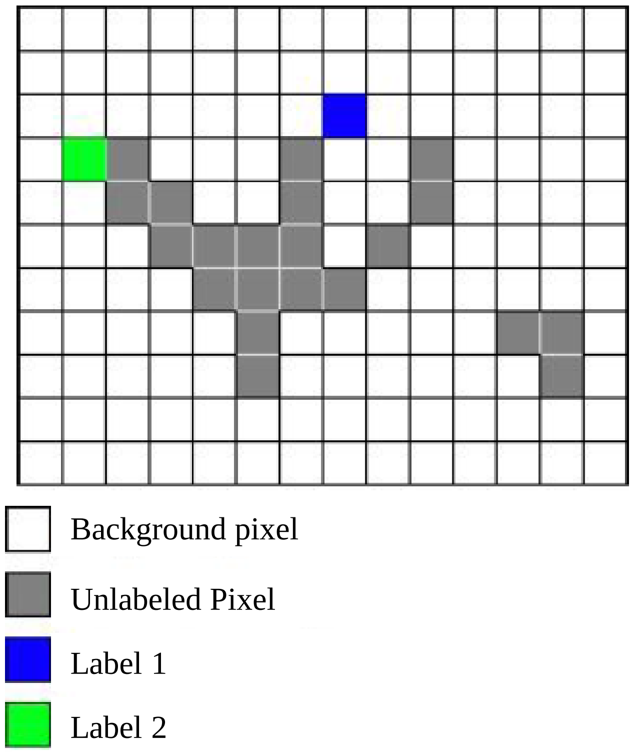
Same algorithm but examine also the upper diagonal neighbors of *p*

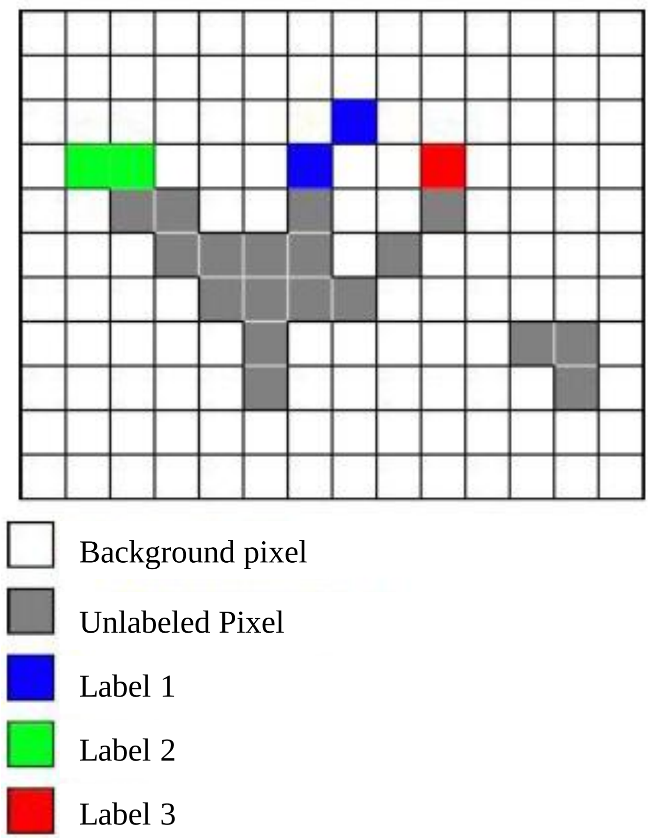
CC labeling – 8 Connectivity



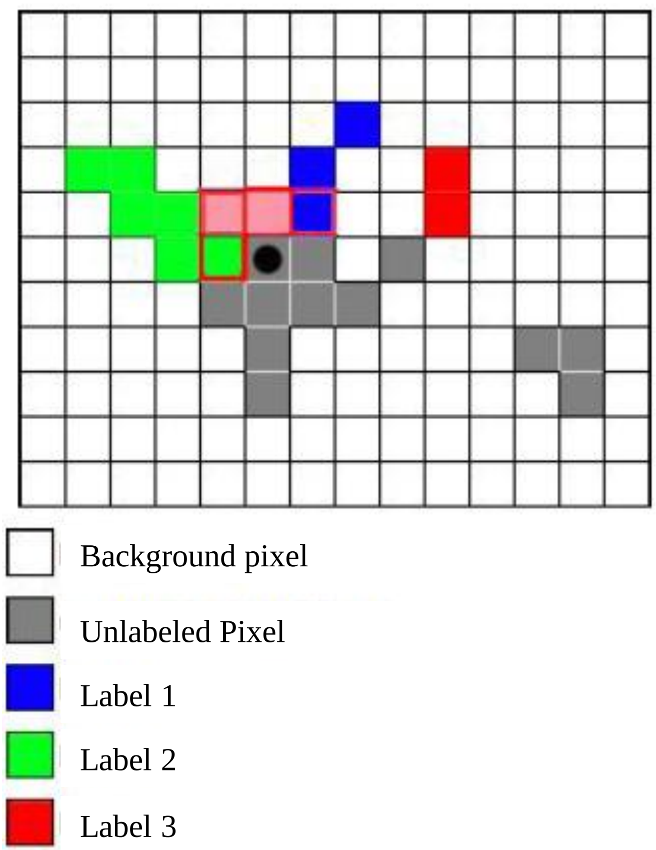


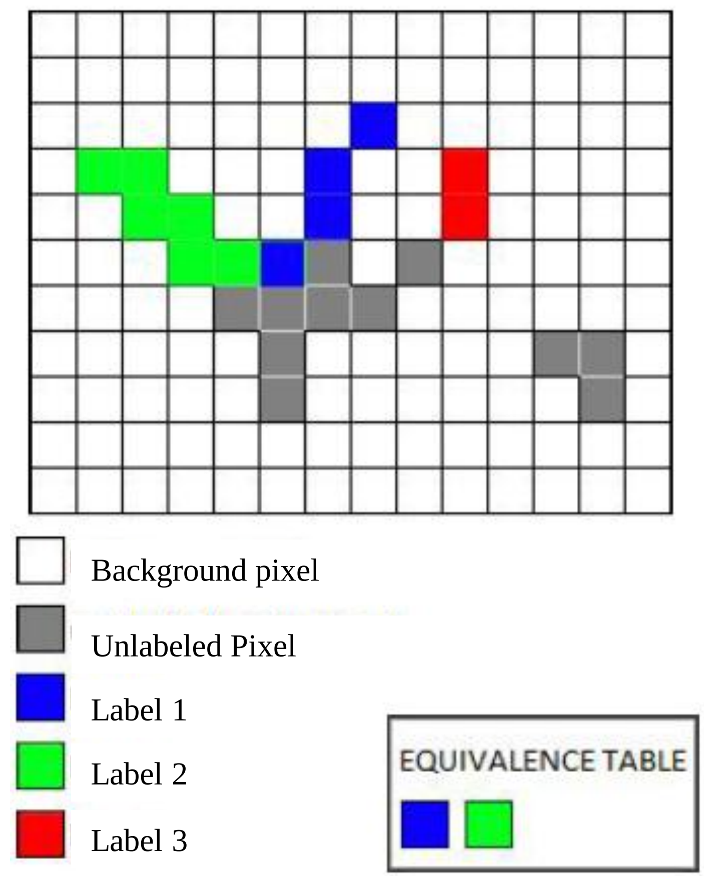
CC labeling – 8 Connectivity



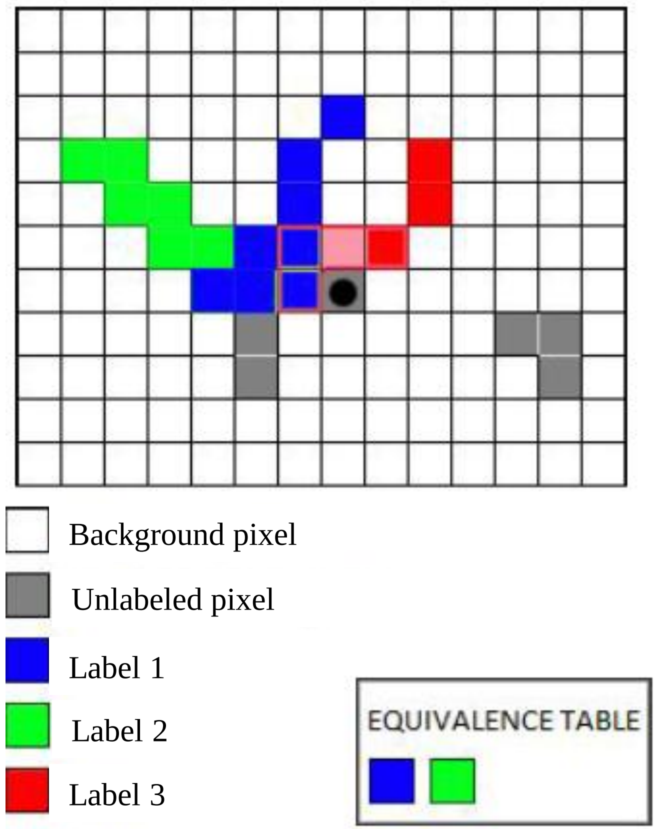


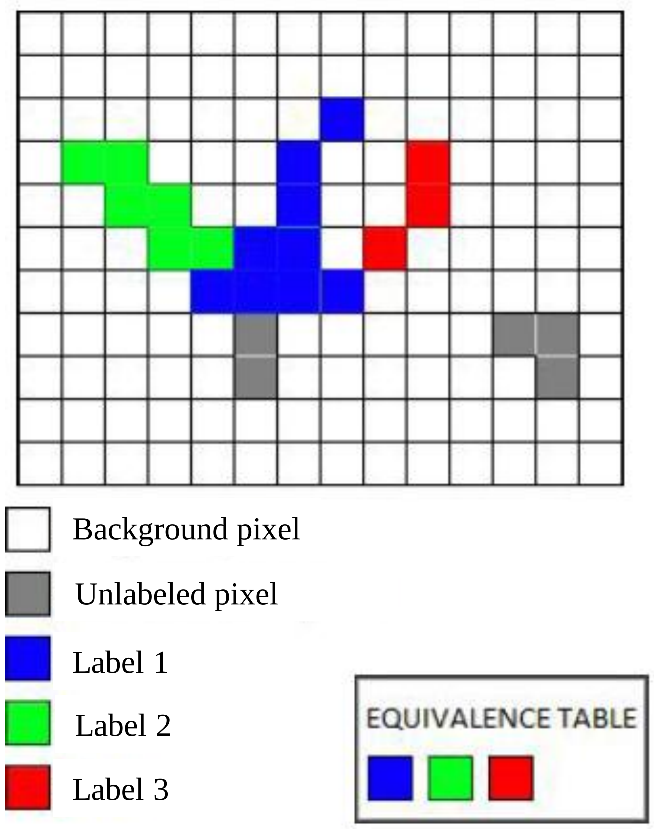
CC labeling – 8 Connectivity



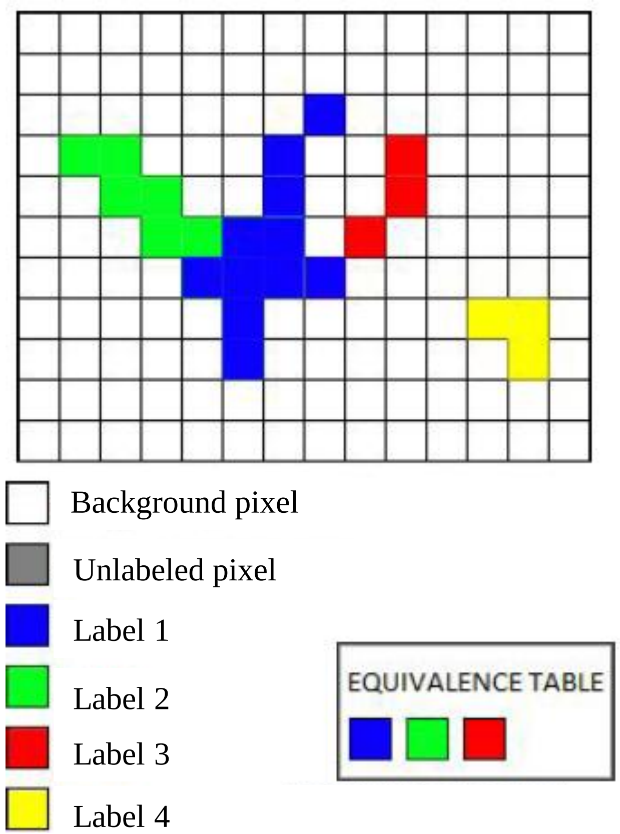


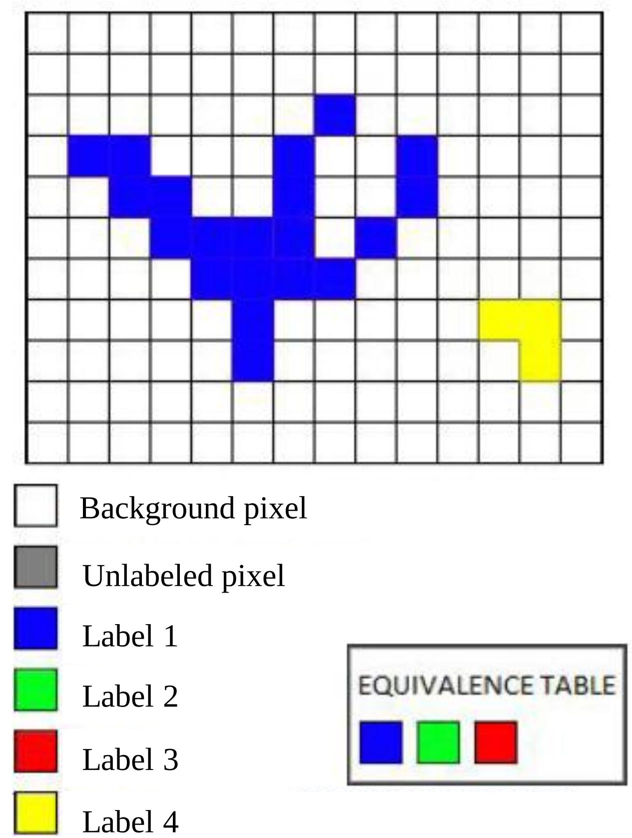
CC labeling – 8 Connectivity





CC labeling – 8 Connectivity





Distance Metrics

- Let pixels *p*, *q* and *z* have coordinates *(x,y)*, *(s,t)* and *(u,v)* respectively.

- D is a distance function or metric if
  - $D(p,q) \geq 0$ and
  - $D(p,q) = 0$ iff $p = q$ and
  - $D(p,q) = D(q,p)$ and
  - $D(p,z) \leq D(p,q) + D(q,z)$

City block distance (D_4 distance)

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

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

- Diamond with center at *(x,y)*
- D_4 = 1 are the 4 neighbors of pixel *p(x,y)*

Chessboard distance (D_8 distance)

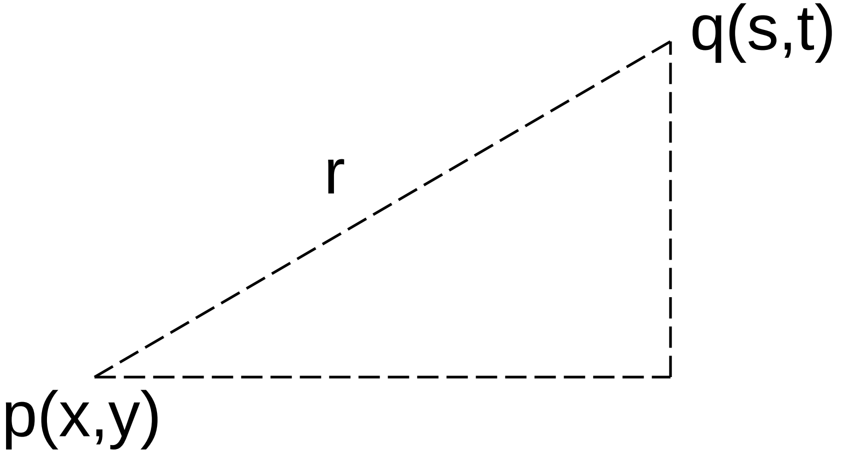
$$D_8(p,q) = \max(|x-s|,|y-t|)$$

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

- Square centered at *p(x,y)*
- D_8 = 1 are the 8 neighbors of pixel *p(x,y)*

Euclidean Distance

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



A circle with radius *r* centered at *(x,y)*

Arithmetic Operations

- Carried out between corresponding pixel pairs

$$s(x,y) = f(x,y) + g(x,y)$$

$$d(x,y) = f(x,y) - g(x,y)$$

$$p(x,y) = f(x,y) \times g(x,y)$$

$$d(x,y) = f(x,y) \div g(x,y)$$

Arithmetic Operations

- Conversion to range 0 – 255
- Difference of two 8-bit images: -255 to 255
- Sum of two 8-bit images: 0 to 510
- Solution?

Set all values < 0 to 0

Set all values > 255 to 255

Full range of arithmetic operation not captured

Arithmetic Operations

- First perform the operation

$$f_m = f - \min(f)$$

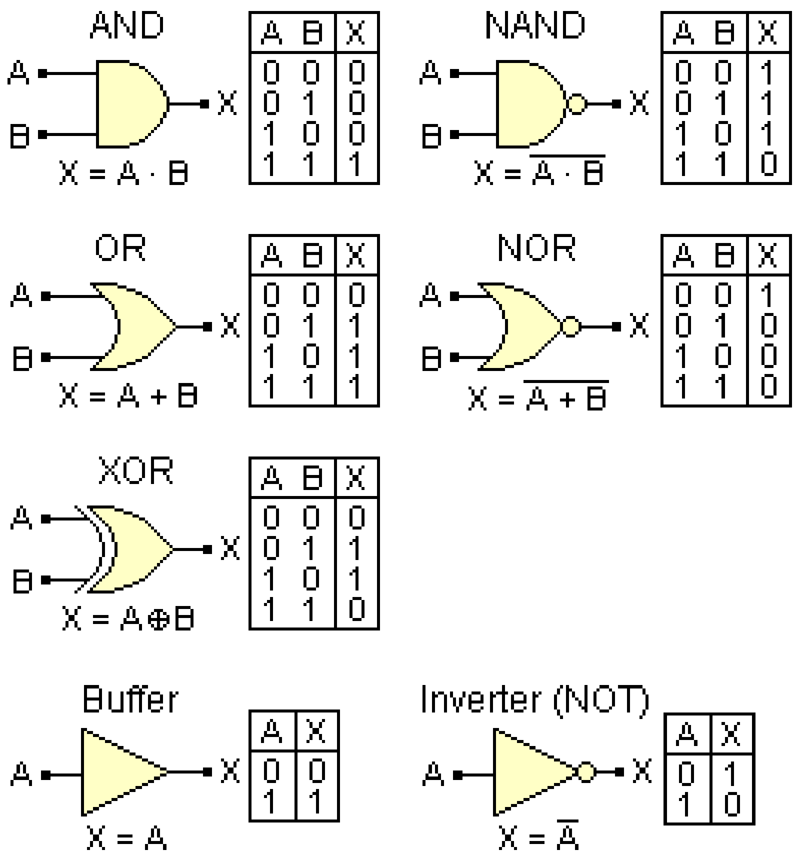
Creates an image whose minimum value is 0

- Then perform

$$f_s = K\left[f_m / \max(f_m)\right]$$

Creates a scaled image f_s with values in the range [0 K]

Logical Operations (Binary Images)



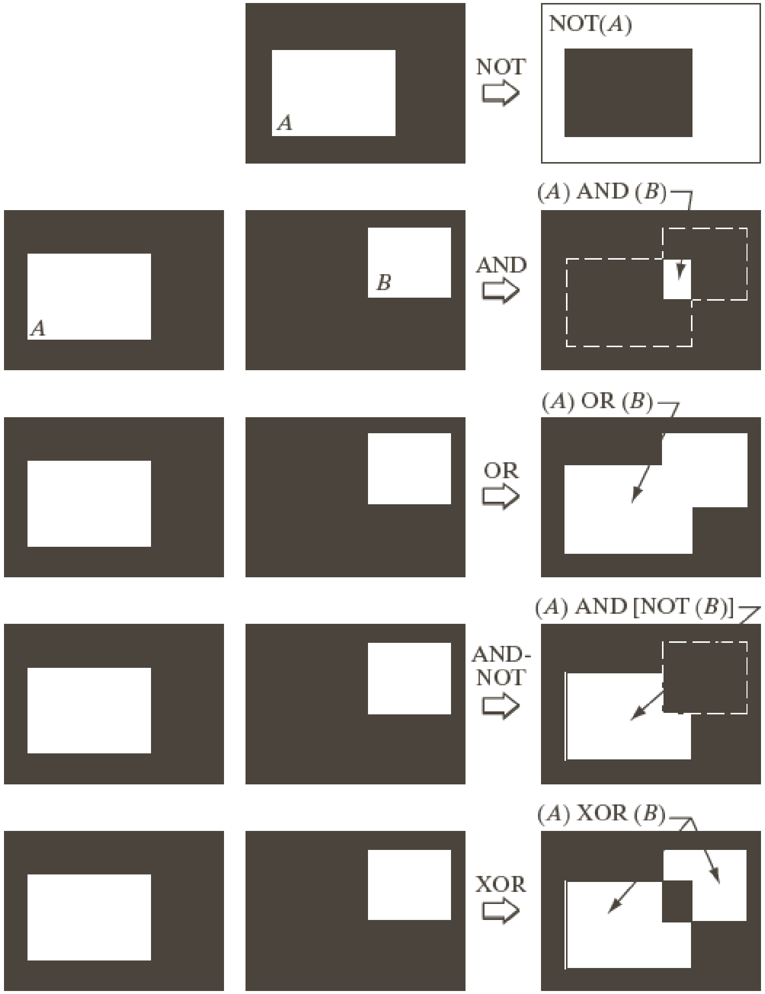
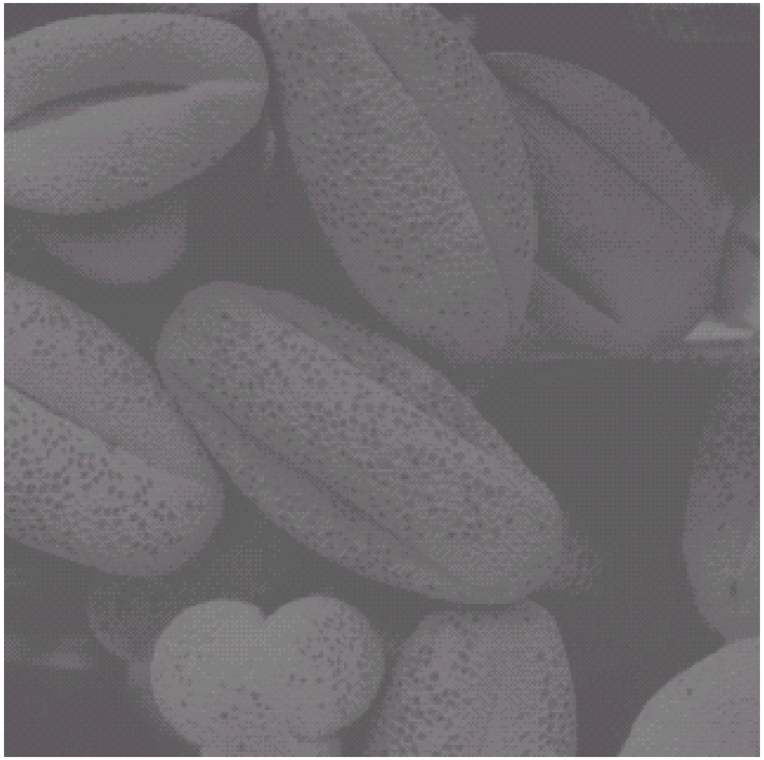


Image Enhancement

Image Enhancement

Image Enhancement



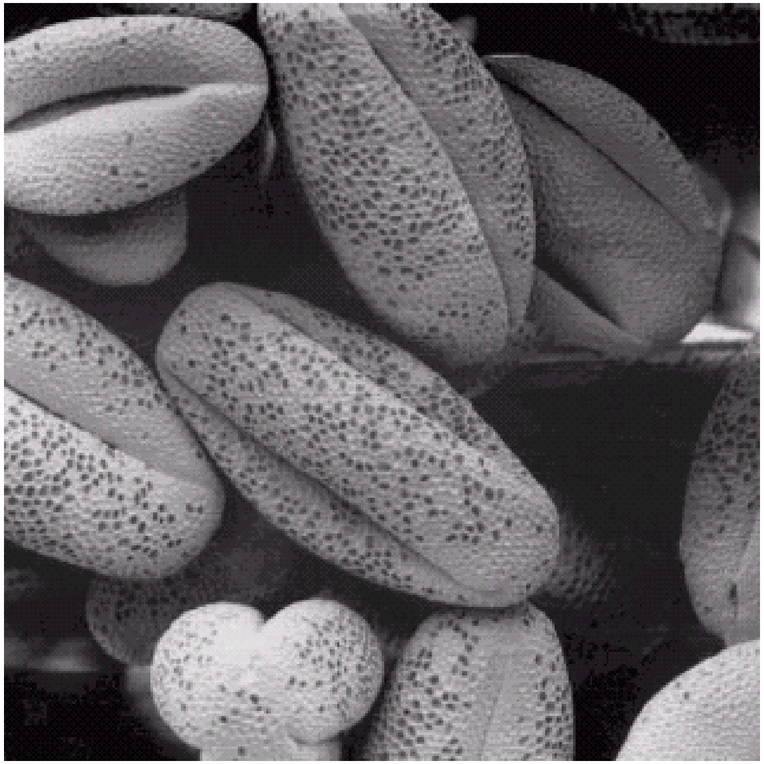


Image Enhancement

Process an image so that the result is more suitable than the original image for a specific application

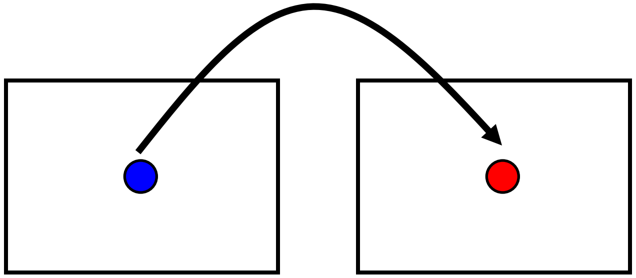
- Image Enhancement Methods
  - Spatial Domain: Direct manipulation of pixels in an image
  - Frequency Domain: Process the image by modifying the Fourier transform of an image

This Chapter – Spatial Domain

Types of image enhancement operations

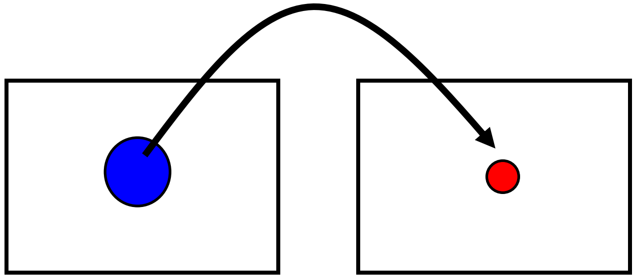
- Point/Pixel operations

  Output value at specific coordinates *(x,y)* is dependent only on the input value at *(x,y)*



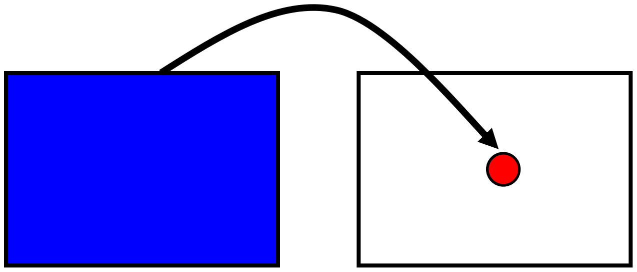
- Local operations

  The output value at *(x,y)* is dependent on the input values in the neighborhood of *(x,y)*



- Global operations

  The output value at *(x,y)* is dependent on all the values in the input image



Basic Concepts

- Most spatial domain enhancement operations can be generalized as:

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

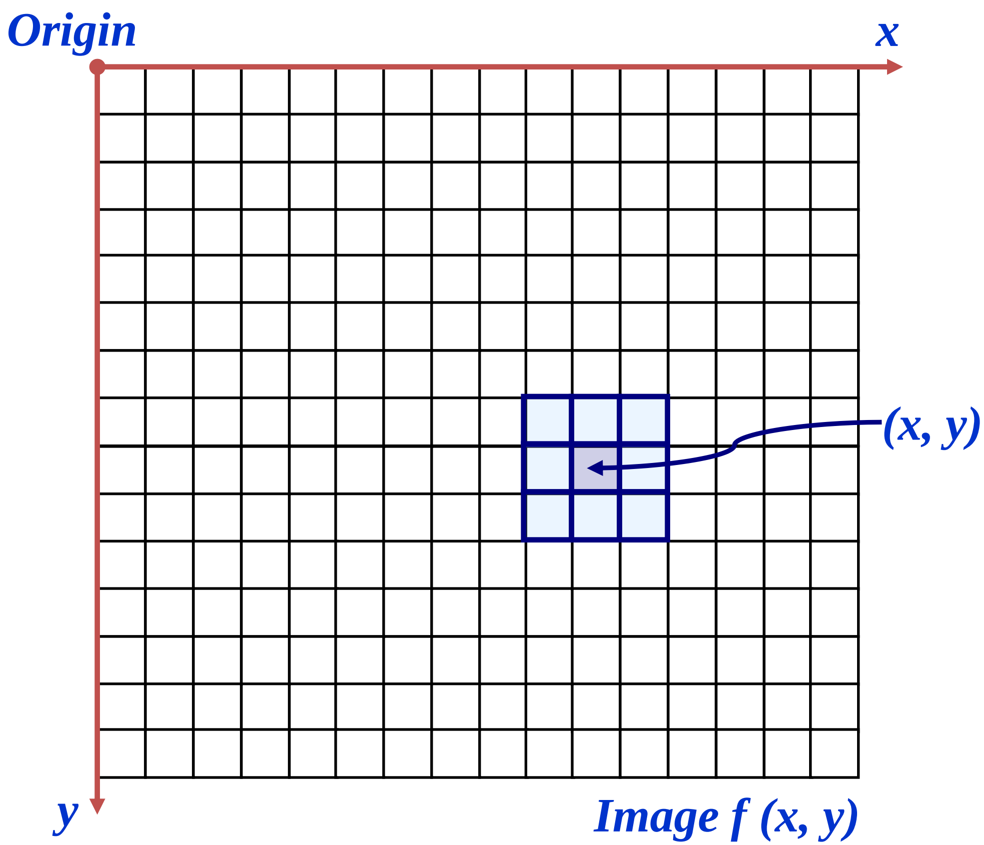
f (x, y) = the input image

g (x, y) = the processed/output image

T = some operator defined over some neighbourhood of *(x, y)*

Basic Concepts

A square or rectangular sub-image area centered at (x,y)



Point Processing

- In a digital image, point = pixel
- Point processing transforms a pixel's value as function of its value alone;
- It does not depend on the values of the pixel's neighbors.

Point Processing

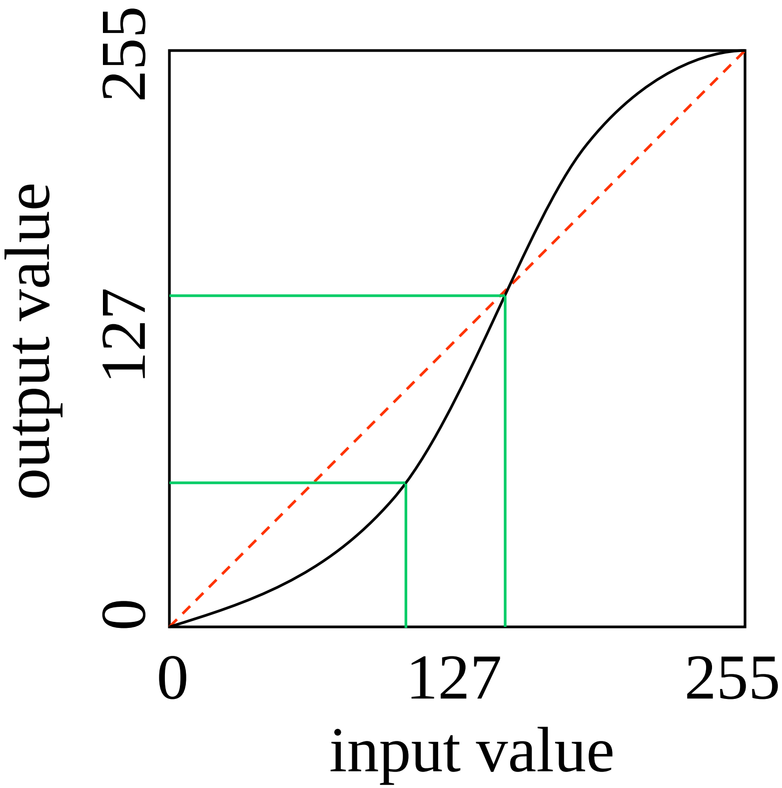
- Neighborhood of size 1x1:
- *g* depends only on *f* at *(x,y)*
- *T*: Gray-level/intensity transformation/ mapping function

$$s = T(r)$$

- *r* = gray level of *f* at *(x,y)*
- *s* = gray level of *g* at *(x,y)*

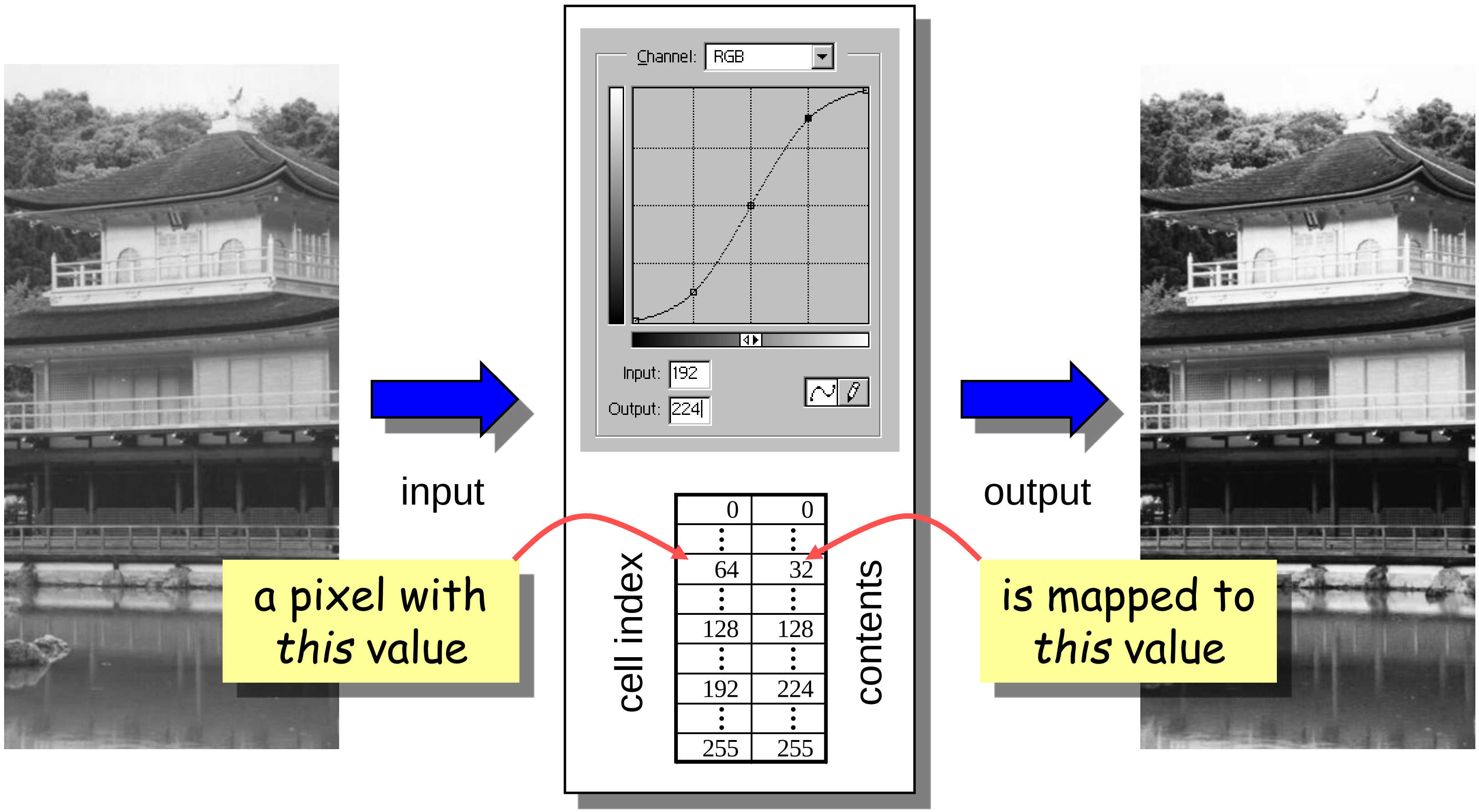
Point Processing using Look-up Tables

A look-up table (LUT) implements a functional mapping.

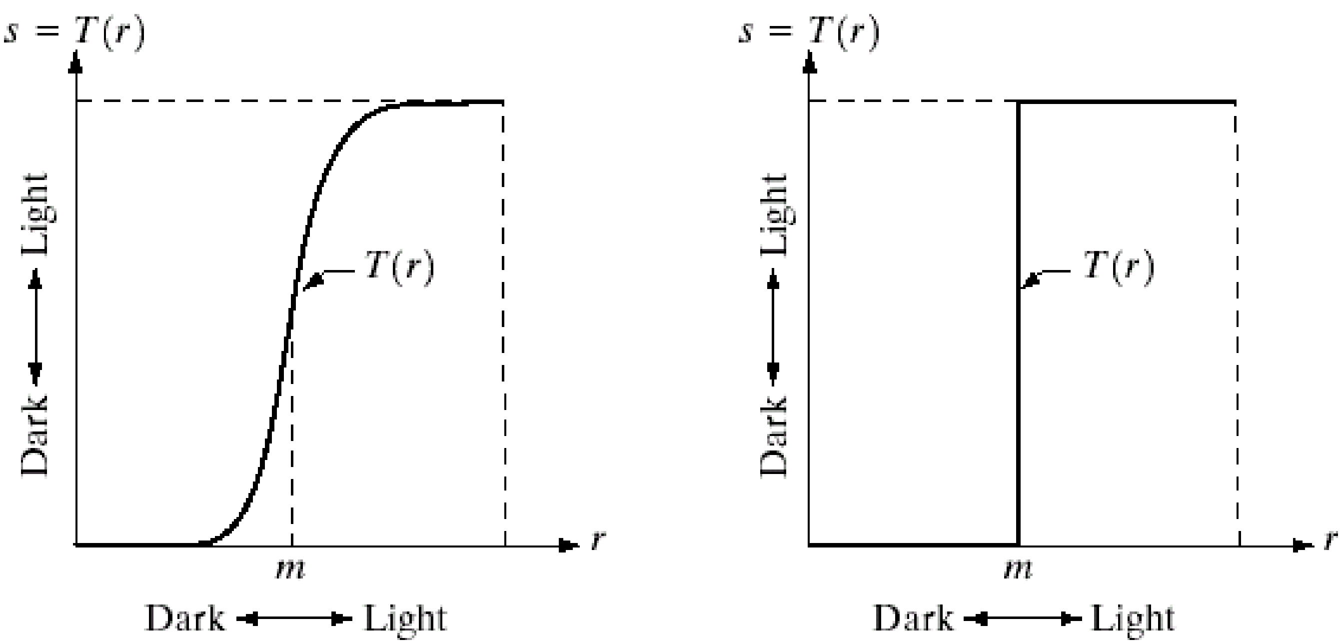


E.g.:	index	value
	...	...
	101	64
	102	68
	103	69
	104	70
	105	70
	106	71
	...	...
	input	output

Point Processing using Look-up Tables



POINT PROCESSING



a b

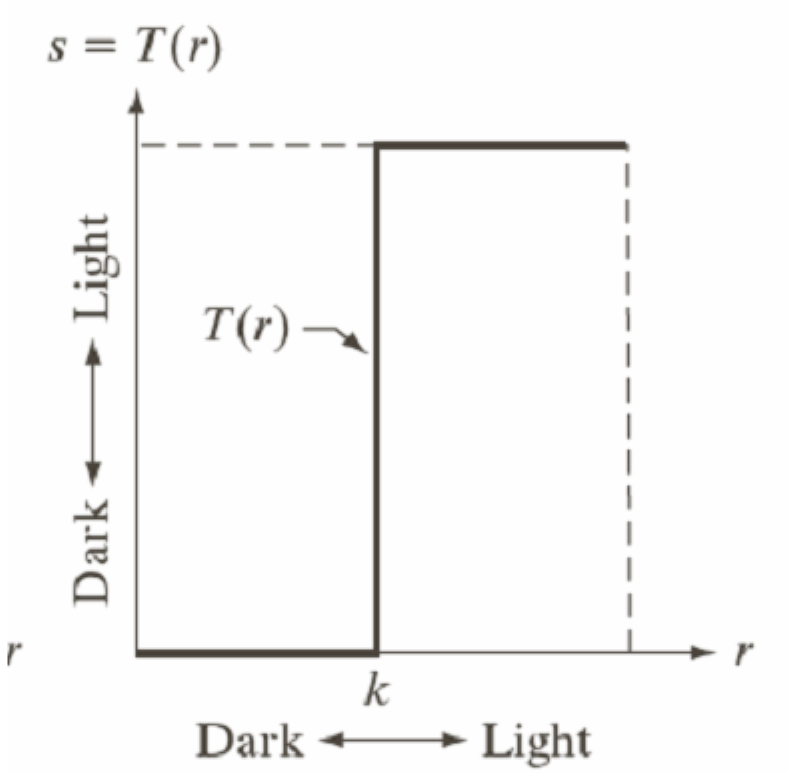
FIGURE 3.2 Gray-level transformation functions for contrast enhancement.

Contrast Stretching

Thresholding

Point Processing Example: Thresholding

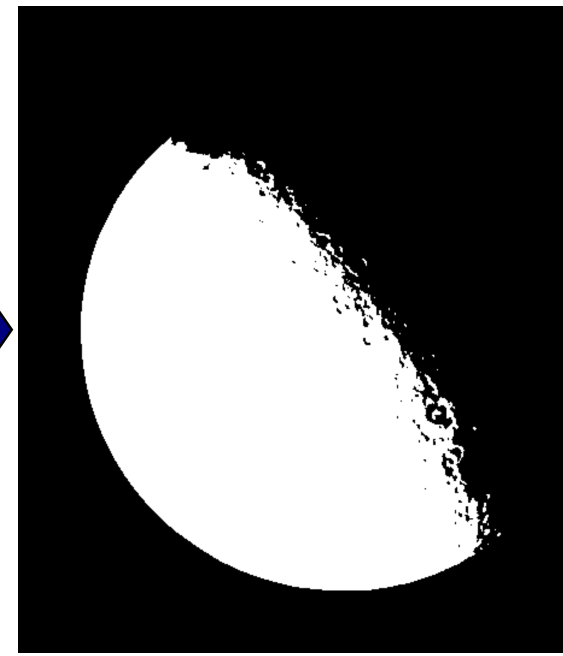
$$s = \begin{cases} 1.0 & r > threshold \\ 0.0 & r <= threshold \end{cases}$$



Point Processing Example: Thresholding

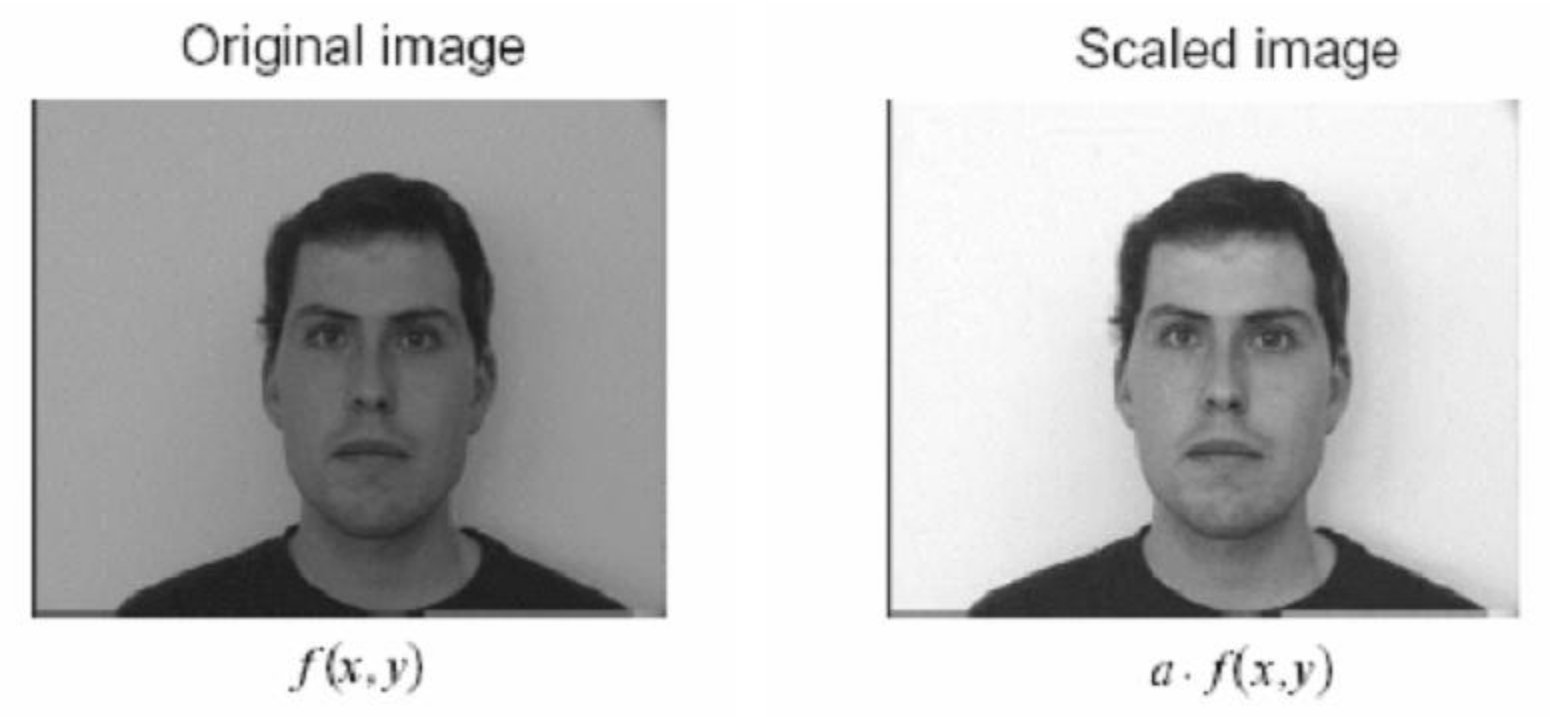
- Segmentation of an object of interest from a background

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



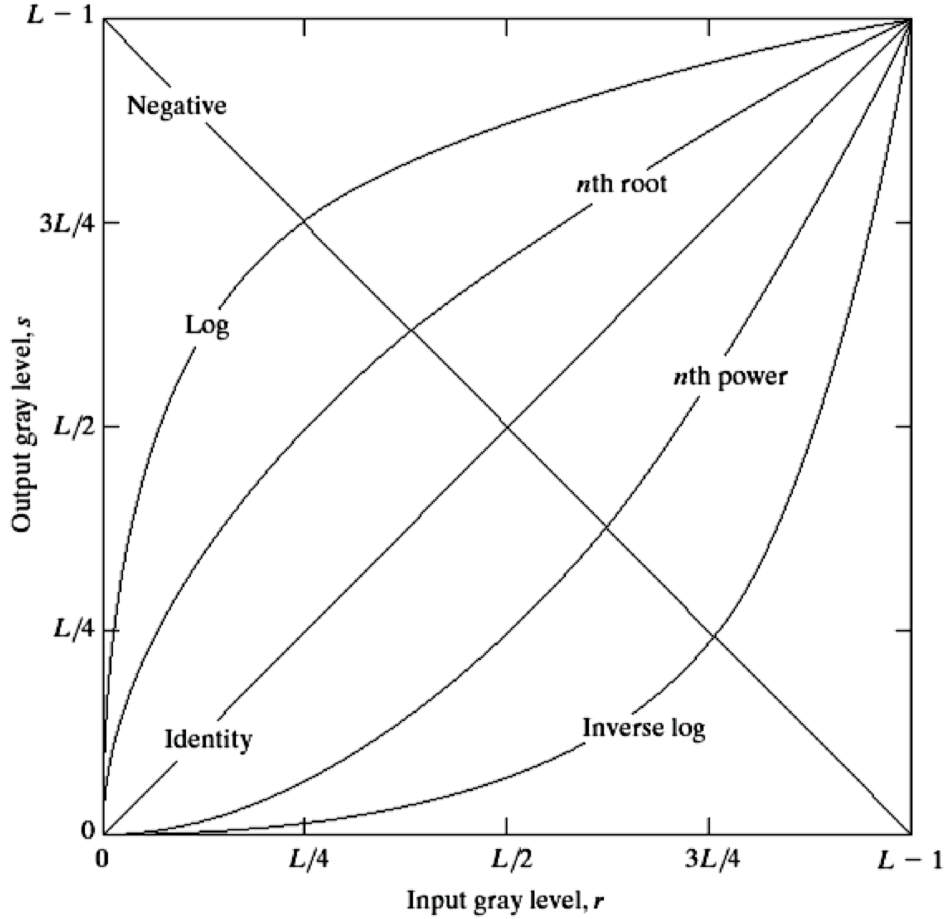
Point Processing Example: Intensity Scaling

$$s = T(r) = a.r$$



Point Processing 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



Point Processing Example: Negative Images

- Reverses the gray level order
- For L gray levels, the transformation has the form:

$$s = (L-1) - r$$

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

Point Processing Example: Negative Images

Input image (X-ray image)

Output image (negative)

Logarithmic Transformations

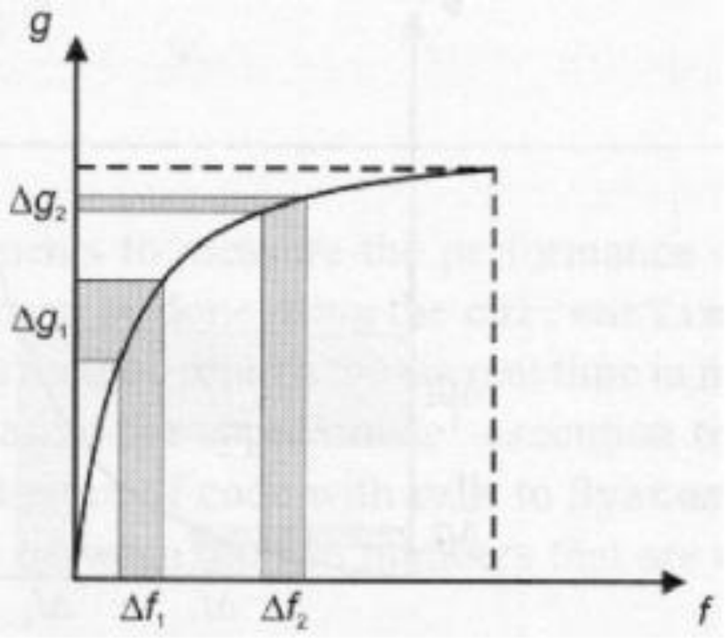
- The general form of the log transformation is

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

- 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

Logarithmic Transformations

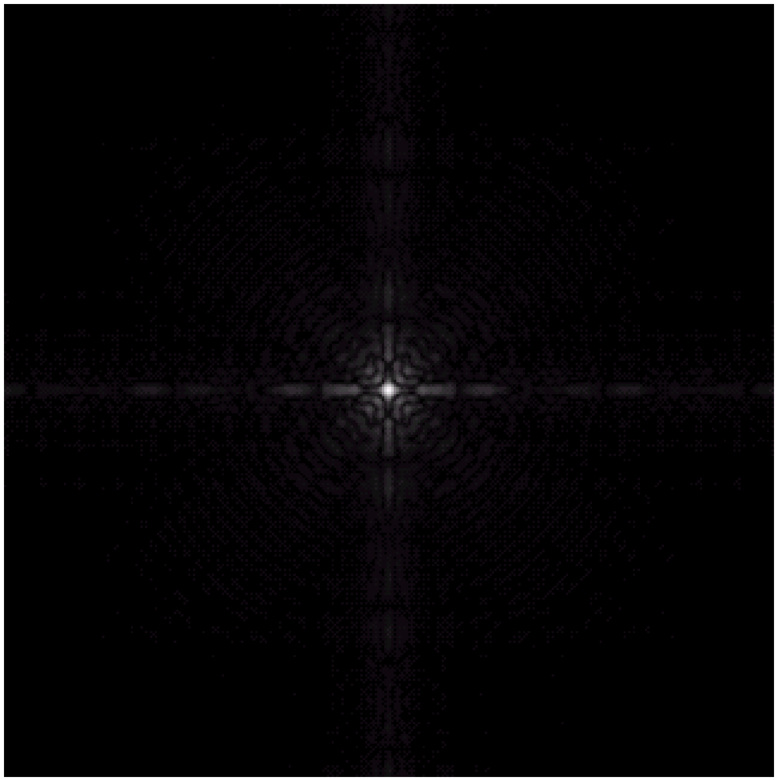
- **Properties**
  - For lower amplitudes of input image the range of gray levels is expanded
  - For higher amplitudes of input image the range of gray levels is compressed



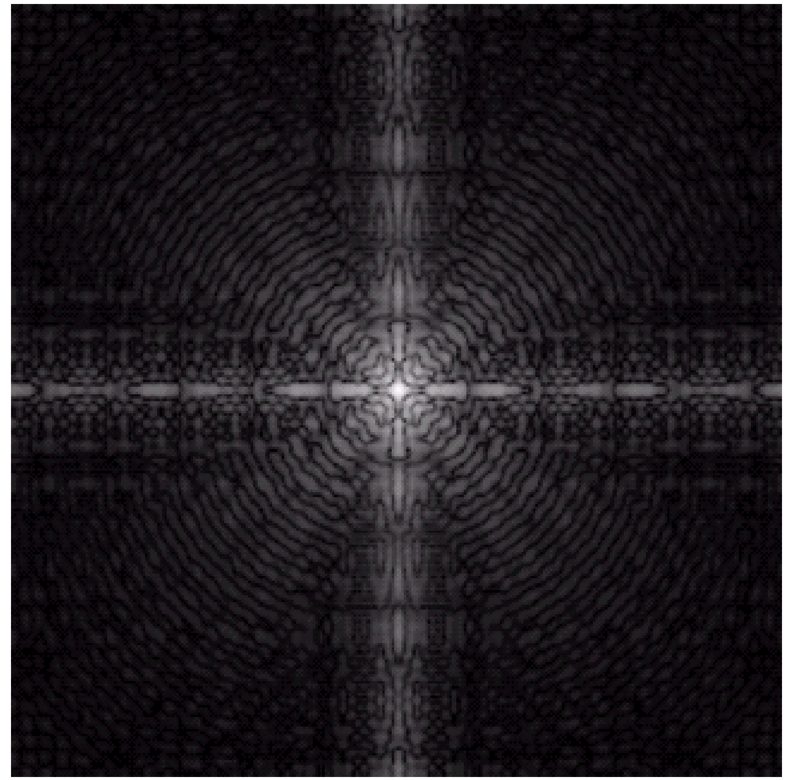
Logarithmic Transformations

- **Application**
  - This transformation is suitable for the case when the dynamic range of a processed image far exceeds the capability of the display device (e.g. display of the Fourier spectrum of an image)
  - Also called "dynamic-range compression / expansion"

Logarithmic Transformations



Fourier spectrum: image values ranging from 0 to 1.5×10^6



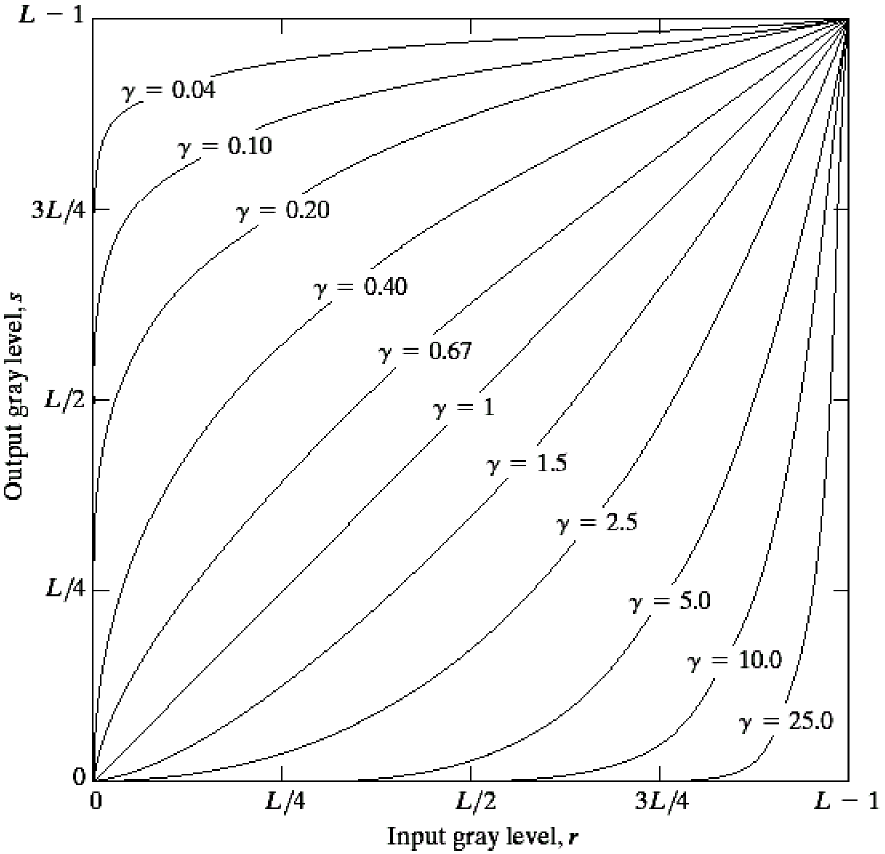
The result of log transformation with c = 1

Power Law Transformations

- Power law transformations have the following form

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

- 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

- For $\gamma < 1$: Expands values of dark pixels, compress values of brighter pixels
- For $\gamma > 1$: Compresses values of dark pixels, expand values of brighter pixels
- If $\gamma=1$ & $c=1$: Identity transformation ($s = r$)

- A variety of devices (image capture, printing, display) respond according to a power law and need to be corrected

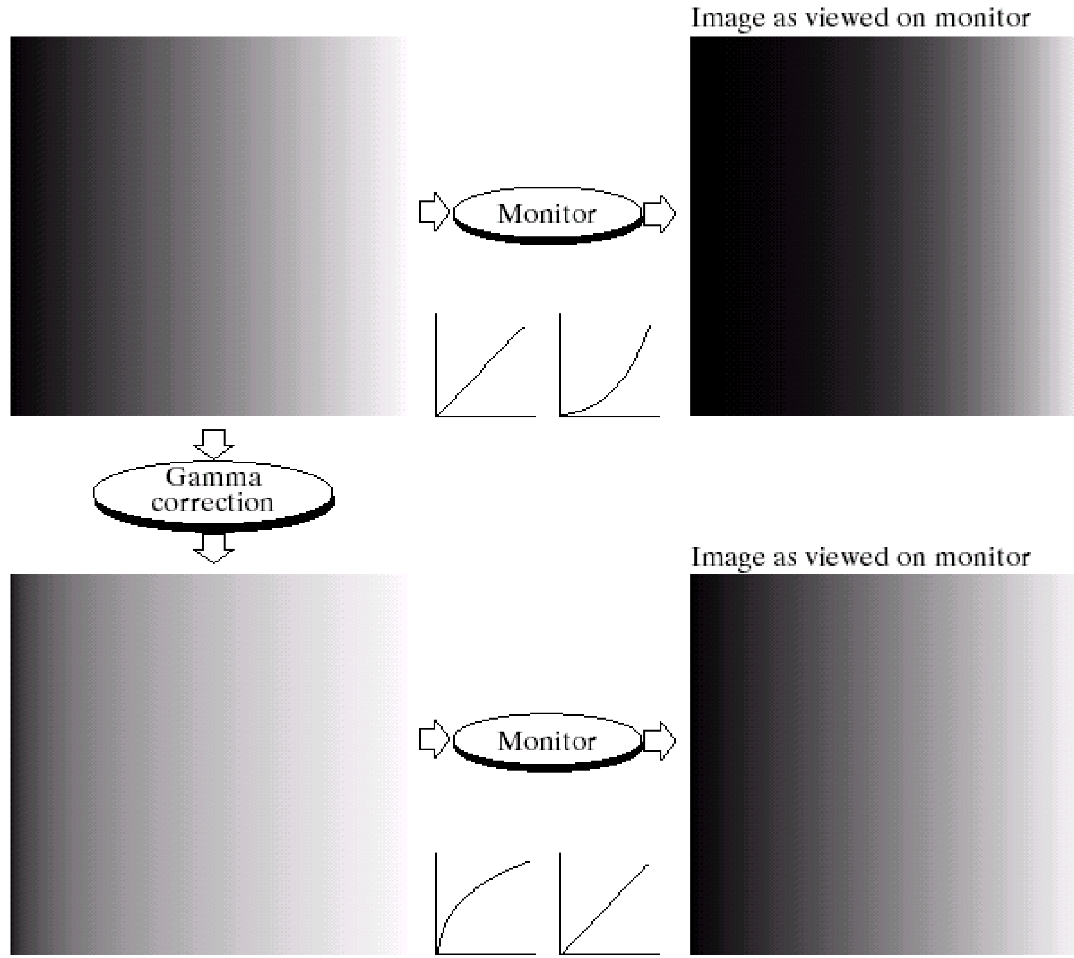
- **Gamma (γ) correction**

  The process used to correct the power-law response phenomena

Power Law Transformations: Gamma Correction

a b
c d

FIGURE 3.7
(a) Linear-wedge gray-scale image.
(b) Response of monitor to linear wedge.
(c) Gamma-corrected wedge.
(d) Output of monitor.



Power Law Transformations
Contrast Enhancement

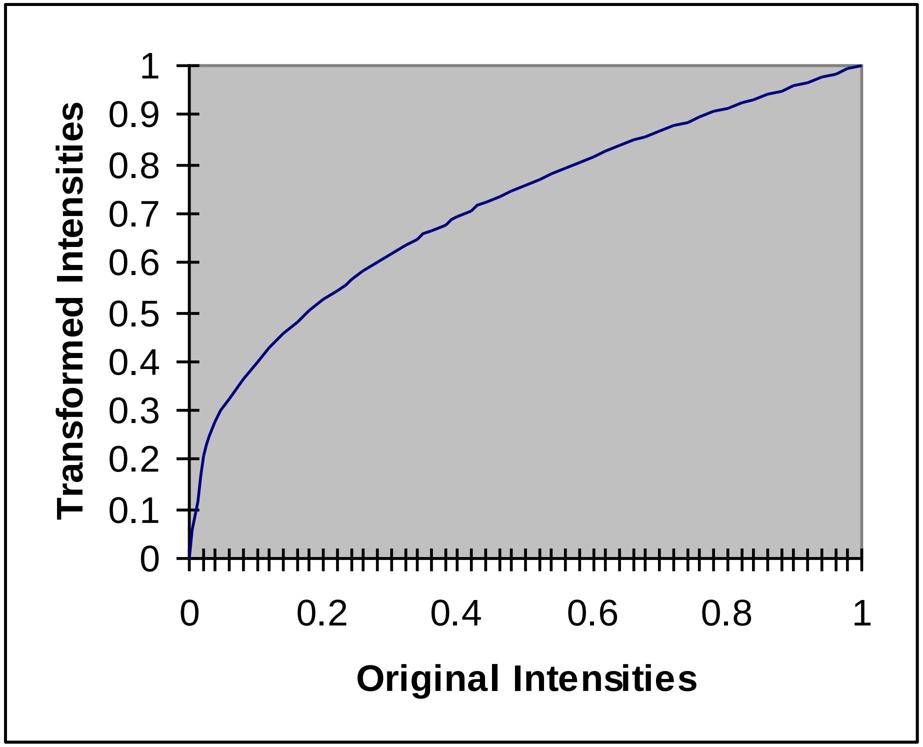
The images to the right show a magnetic resonance (MR) image of a fractured human spine

Power Law Transformations
Contrast Enhancement

$\gamma = 0.6$

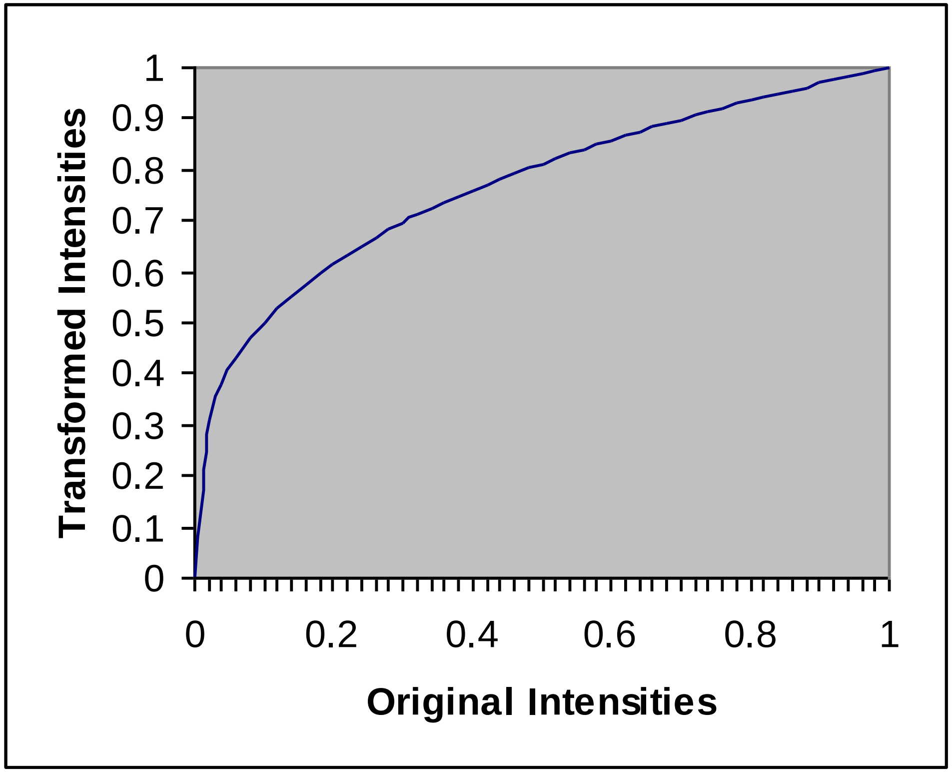
Power Law Transformations
Contrast Enhancement

$\gamma = 0.4$



Power Law Transformations
Contrast Enhancement

$\gamma = 0.3$



Power Law Transformations
Contrast Enhancement

MR image of

fractured human spine

Result after

Power law transformation

c = 1, $\gamma = 0.6$

Result after

Power law transformation

c = 1, $\gamma = 0.4$

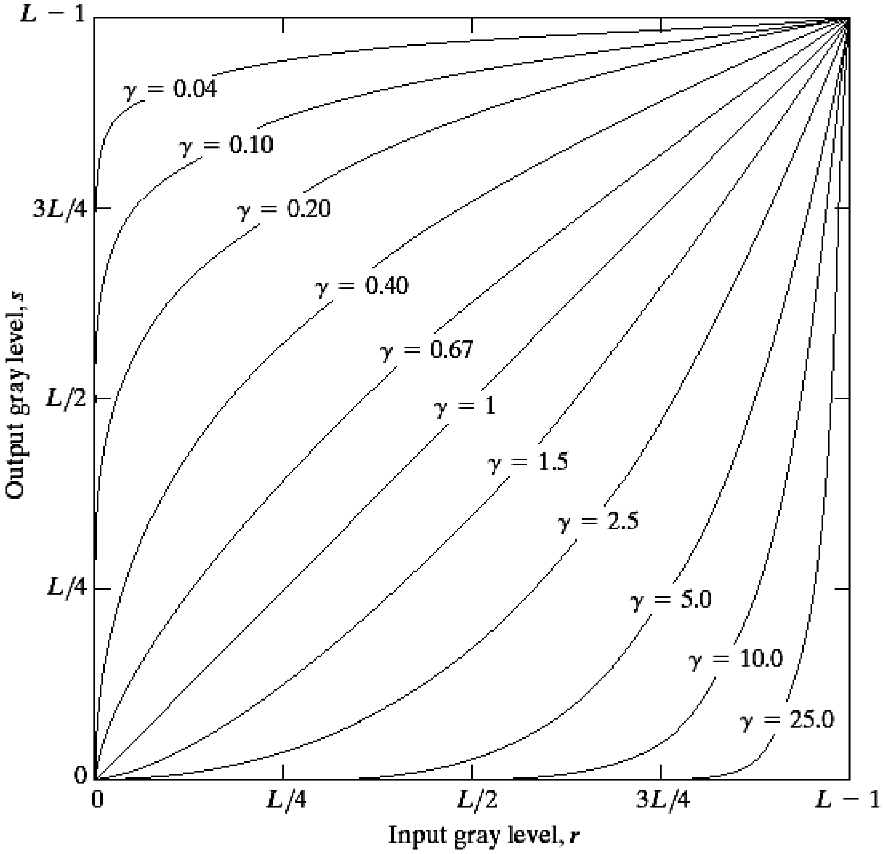
Result after

Power law transformation

c = 1, $\gamma = 0.3$

Power Law Transformations
Contrast Enhancement

When the γ is reduced too much, the image begins to reduce contrast to the point where the image started to have very slight "wash-out" look.



Power Law Transformations
Contrast Enhancement

Image has a washed-out appearance – needs $\gamma > 1$

Image Enhancement

Aerial Image

Result of Power law transformation c = 1, $\gamma = 3.0$ (suitable)

Result of Power law transformation c = 1, $\gamma = 4.0$ (suitable)

Result of Power law transformation c = 1, $\gamma = 5.0$ (high contrast, some regions are too dark)

Readings from Book (3rd Edn.)

- 2.5 Basic Relationships between Pixels
- 2.6 (Reading Assignment)
- 3.1 Background
- 3.2 Some basic intensity transformation functions

Acknowledgements

Material in these slides has been taken from, the following resources

- Digital Image Processing", Rafael C. Gonzalez & Richard E. Woods, Addison-Wesley, 2009
- Machine Vision: Automated Visual Inspection and Robot Vision", David Vernon, Prentice Hall, 1991