CSE468: IMAGE PROCESSING

Digital Image Fundamentals

Lecturer: Dr. Hossam Hassan

Email : hossameldin.hassan@eng.asu.edu.eg

Computer and Systems Engineering

Basic Relationship b/w pixels

- Neighbors of a pixel
- Connectivity
- Labeling of Connected Components
- Relations, Equivalences, and Transitive Closure
- Distance Measures
- Arithmetic/Logic Operations

Neighbors of a pixel

- a pixel p at coordinate (x,y) has
  - $N_4(p)$: 4-neighbors of p
    (x+1, y), (x-1,y),(x,y+1), (x,y-1)

	x	
x	p	x
	x	

  - $N_D(p)$: 4-diagonal neighbors of p
    (x+1, y+1), (x+1,y-1),(x-1,y+1), (x-1,y-1)

x		x
	p	
x		x

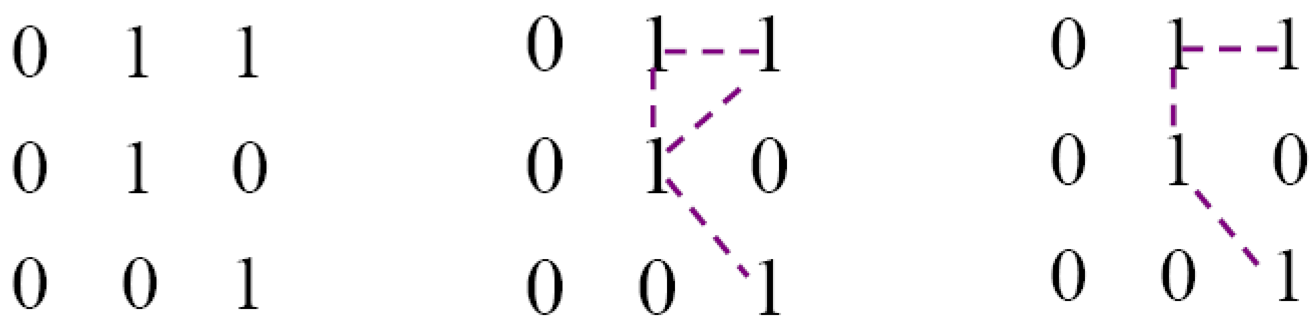
  - $N_8(p)$: 8-neighbors of p :
    a combination of $N_4(p)$ and $N_D(p)$

x	x	x
x	p	x
x	x	x

Connectivity

- Let V be the set of gray-level values used to defined connectivity
  - 4-connectivity :
    - 2 pixels p and q with values from V are 4-connected if q is in the set $N_4(p)$
  - 8-connectivity :
    - 2 pixels p and q with values from V are 8-connected if q is in the set $N_8(p)$
  - m-connectivity (mixed connectivity):
    - 2 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.
      - (the set of pixels that are 4-neighbors of both p and q whose values are from V)

Example



- m-connectivity eliminates the multiple path connections that arise in 8-connectivity.

Adjacent

- a pixel p is adjacent to a pixel q if they are connected.
- two image area subsets S1 and S2 are adjacent if some pixel in S1 is adjacent to some pixel S2.

Path

- a path from pixel p with coordinates (x,y) to pixel q with coordinates (s,t) is a sequence of distinct pixels with coordinates

  $$(x_0,y_0),(x_1,y_1),\ldots(x_n,y_n)$$

  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})
- n is the length of the path
- we can define 4-,8-, or m-paths depending on type of adjacency specified.

Distance Measures

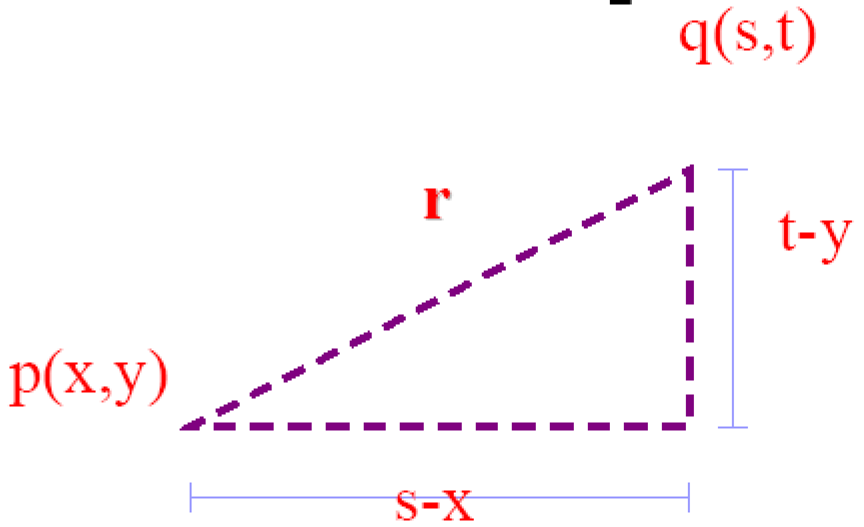
- for pixel p, q and z with coordinates (x,y), (s,t) and (u,v) respectively,

- D is a distance function or metric if

  - (a) $D(p,q) \geq 0$; $D(p,q) = 0$ iff $D=q$
  - (b) $D(p,q) = D(q,p)$
  - (c) $D(p,z) \leq D(p,q) + D(q,z)$

Euclidean Distance between p and q

$$D_e(p,q) = \left[(x-s)^2 + (y-t)^2\right]^{1/2}$$



radius (r) centered at (x,y)

City-block Distance: D_4

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

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

- diamond centered at (x,y)
- D_4 =1 are 4-neighbors of (x,y)

Chessboard distance: D_8

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

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 (x,y)

Arithmetic operators

- used extensively in most branches of image processing.
- Arithmetic operations b/w 2 pixels p and q :
  - Addition : p+q used in image average to reduce noise.
  - Subtraction : p-q basic tool in medical imaging.
  - Multiplication : pxq
    - to correct gray-level shading result from nonuniformities in illumination or in the sensor used to acquire the image.
  - Division : p÷q
- Arithmetic Operation entire images are carried out pixel by pixel.

Logic operators

- AND : p AND q (p•q)
- OR : p OR q (p+q)
- COMPLEMENT : NOT q ($\overline{q}$)
- logic operations apply only to binary images.
- arithmetic operations apply to multivalued pixels.
- logic operations used for tasks such as masking, feature detection, and shape analysis.
- logic operations perform pixel by pixel.

Mask operation

- Besides pixel-by-pixel processing on entire images, arithmetic and Logical operations are used in neighborhood oriented operations.

		⋮		
	z_1	z_2	z_3	
⋯	z_4	z_5	z_6	⋯
	z_7	z_8	z_9	
		⋮		

Mask Operation

- Let the value assigned to a pixel be a function of its gray level and the gray level of its neighbors.
- e.g., replace the gray value of pixel Z_5 with the average gray values of it's neighborhood within a 3x3 mask.

$$Z = \frac{1}{9}(Z_1 + Z_2 + Z_3 + \ldots + Z_9)$$

Mask Operator

- In general term:

$$Z = \frac{1}{9}Z_1 + \frac{1}{9}Z_2 + \frac{1}{9}Z_3 + \ldots + \frac{1}{9}Z_9$$

$$= w_1 Z_1 + w_2 Z_2 + w_3 Z_3 + \ldots w_9 Z_9$$

$$= \sum_{i=1}^{9} w_i Z_i$$

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Mask Coefficient

- Proper selection of the coefficients and application of the mask at each pixel position in an image makes possible a variety of useful image operations
  - noise reduction
  - region thinning
  - edge detection
- Applying a mask at each pixel location in an image is a computationally expensive task.

Image Geometry

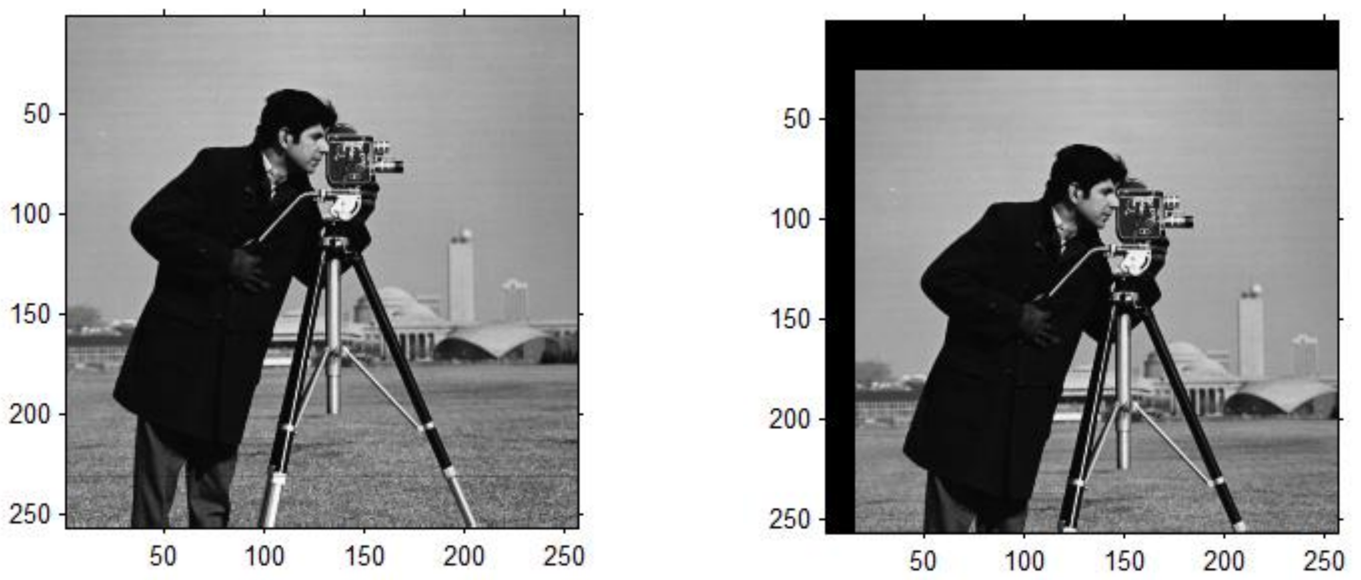
Basic image transformation (can be in 2D or 3D):-

- Scaling
- Rotation
- Translation

Transformation Matrix

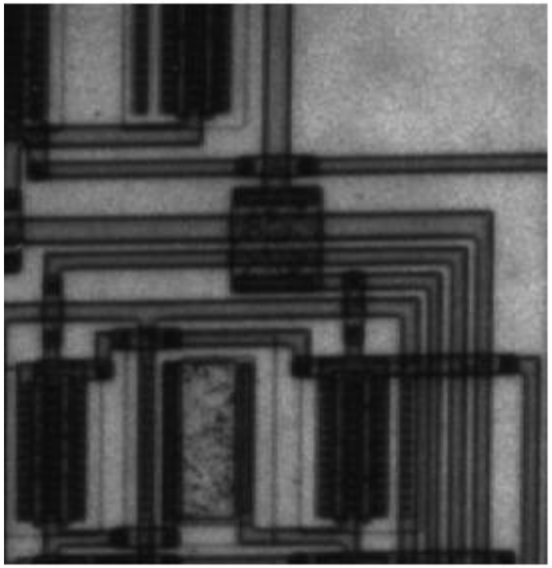
- Translation (Example in 2D)

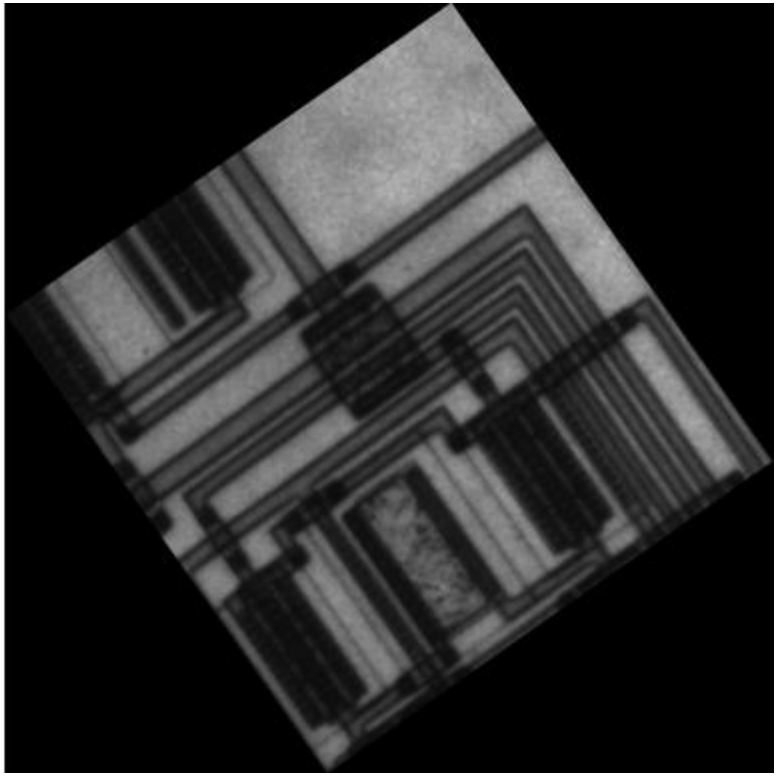
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$



Transformation Matrix

- Rotation Matrix (Example in 2D) $\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$





Transformation Matrix

- Scaling/Resizing

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{sx} & 0 & 0 \\ 0 & \mathbf{sy} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

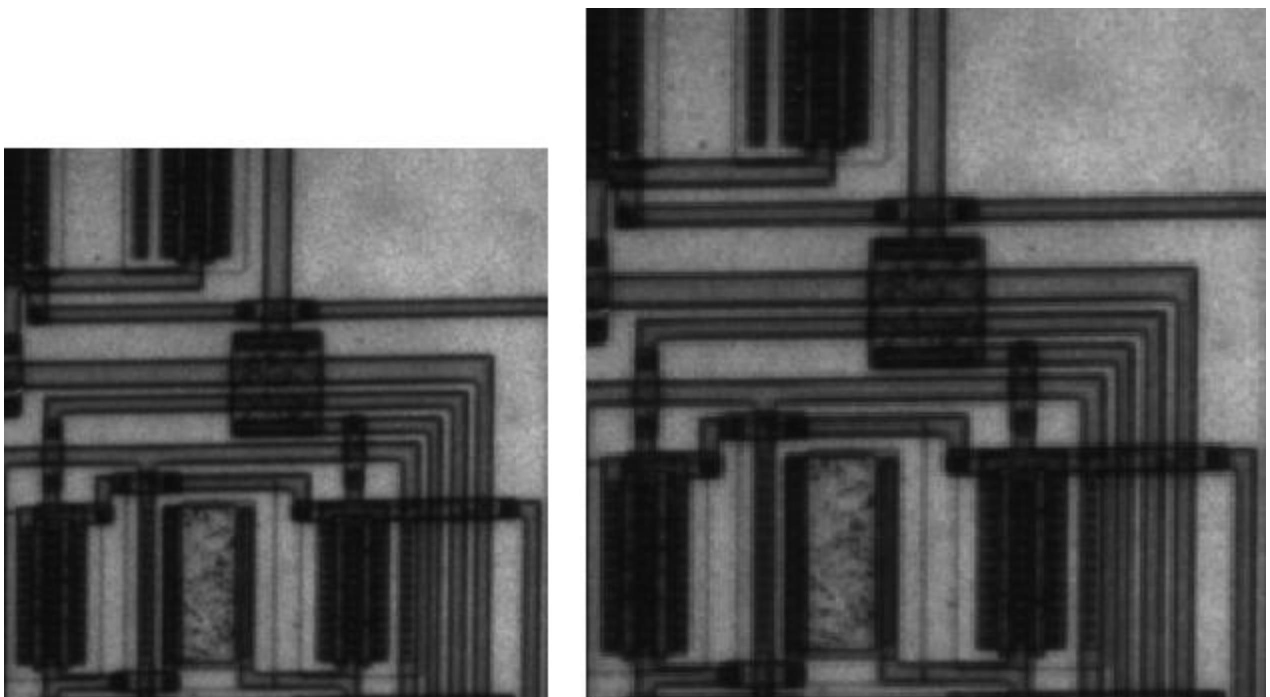
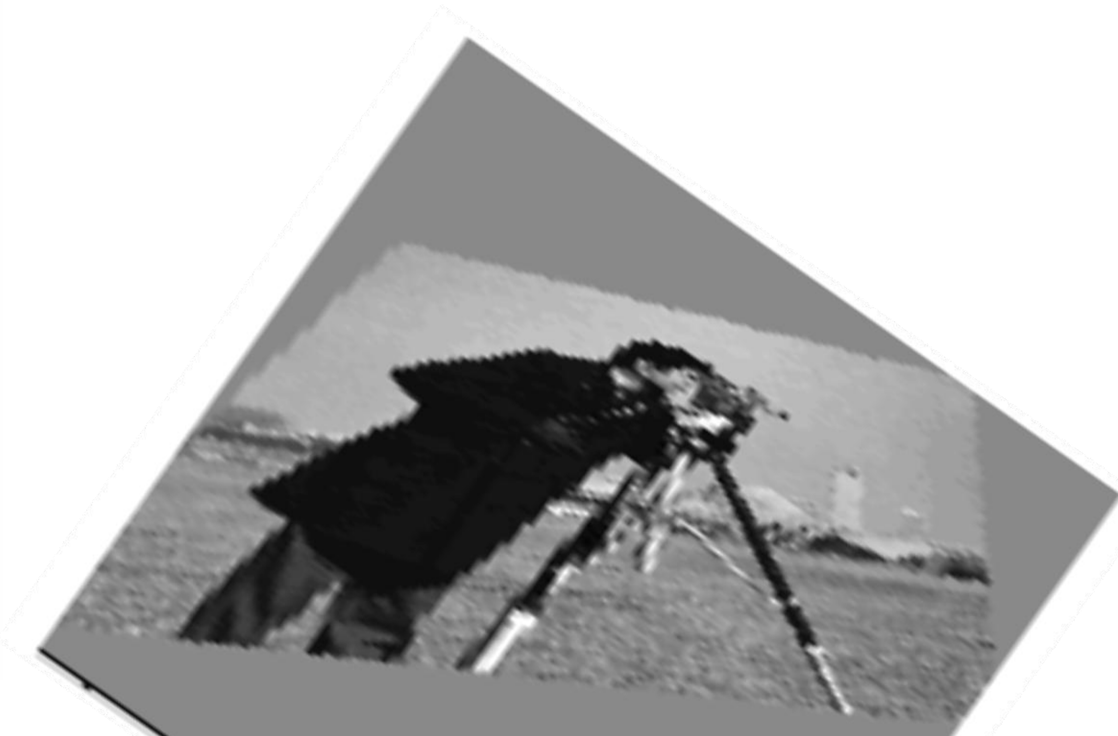


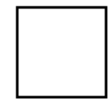
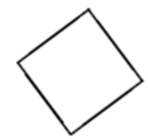
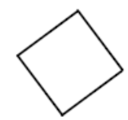
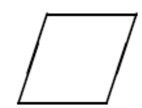
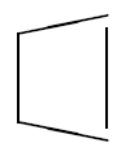
Image Courtesy of Steve Decker and Shujaat Nadeem

Projective Transformation Matrix

$$\begin{cases} \hat{x} = \dfrac{xh_1 + yh_2 + h_3}{xh_7 + yh_8 + h_9} \\ \hat{y} = \dfrac{xh_4 + yh_5 + h_6}{xh_7 + yh_8 + h_9} \end{cases} \qquad H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$$



Hierarchy of Coordinate Transformations

Transformation	Preserves	Icon
translation	orientation	
rigid (Euclidean)	lengths	
* similarity	angles	
affine	parallelism	
projective	straight lines	

***Homogeneous Scaling, rotation, and translation**