Review

What we discussed last time

- Image acquisition, formation and sensing
- Sampling, quantisation
- Image representation
- Spatial, Intensity and temporal resolution

Quiz

Q1. Consider an image with 100 lines and 1000 pixels per line. Each pixel can take 256 different values. The total amount of bits needed to store that image is ____ ?

Q2. A color video has frame rate of 30 frames per second. Considering that spatial resolution of each frame (images) is 1000×1000 pixels and every color component is stored using 8 bits then a minute of (uncompressed) video will occupy ?

Quiz

Q3. An imaging system and object surface has illumination and reflectance range as:

$$1 <= i(x,y) < = 100 \text{ and } 0.5 < = r(x,y) < = 1$$

The acquired image is stored as grayscale with 8 bits per pixel. An intensity value of say 25 at any location will get stored as ___?

Contents

This lecture will cover:

– Neighbourhood and Connectivity

– Connected component labeling

– Distance measures

– Basic Image Operations

Neighbors of a Pixel

A pixel *p* at coordinates (*x*,*y*) has four *horizontal* and *vertical* neighbors whose coordinates are given by:

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

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

This set of pixels, called the 4-*neighbors* of *p*, is denoted by $N_4(p)$.

Neighbors of a Pixel

The four *diagonal* neighbors of *p* have coordinates:

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

and are denoted by $N_D(p)$.

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

These points, together with the 4-neighbors, are called the 8-neighbors of p, denoted by $N_8(p)$.

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

Pixel Connectivity

Connectivity between pixels is important:

- Because it is used to determine components/ regions in an image, trace contours and establish boundaries of objects

Two pixels are connected if:

- They are “neighbors” in some sense (i.e. $N_4(p)$, $N_8(p)$, …)
- Their gray levels satisfy a specified criterion of similarity (e.g. equality, …)

Adjacency

Let V: a set of intensity values used to define adjacency.

In a binary image, $V = \{1\}$, if we are referring to adjacency of pixels with value 1.

In a gray-scale image, the idea is the same, but V typically contains more elements, for example, $V = \{180, 181, 182, \ldots, 200\}$

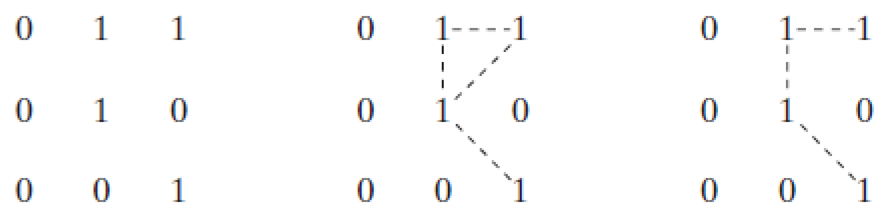
Types of Adjacency

1. **4-adjacency:** Two pixels *p* and *q* with values from *V* are 4-adjacent if *q* is in the set $N_4(p)$.
2. **8-adjacency:** Two pixels *p* and *q* with values from *V* are 8-adjacent if *q* is in the set $N_8(p)$.
3. **m-adjacency (mixed):** Two pixels *p* and *q* with values from *V* are m-adjacent if :
   - *q* is in $N_4(p)$ or
   - *q* is in $N_D(p)$ and the set $N_4(p) \cap N_4(q)$ has no pixel whose values are from V (no intersection)

Types of Adjacency

Mixed adjacency eliminates the ambiguities that often arise when 8-adjacency is used.

Eg.



a b c

FIGURE 2.26 (a) Arrangement of pixels; (b) pixels that are 8-adjacent (shown dashed) to the center pixel; (c) *m*-adjacency.

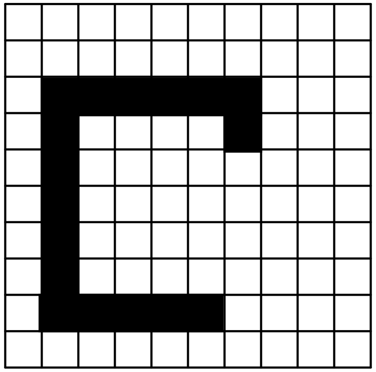
A Digital Path

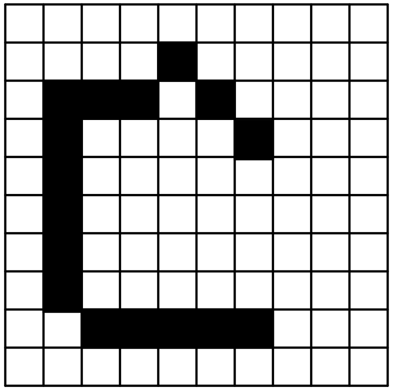
A digital path (or curve) from pixel p with coordinate (x,y) to pixel q with coordinate (s,t) is a sequence of distinct pixels:

(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

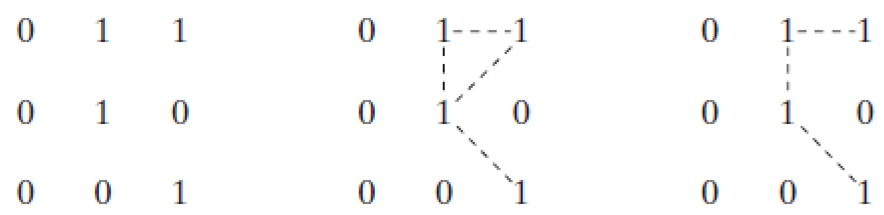
(x_i, y_i) and (x_{i-1}, y_{i-1}) are adjacent for $1 \le i \le n$





A Digital Path

Return to the previous example:



a b c

FIGURE 2.26 (a) Arrangement of pixels; (b) pixels that are 8-adjacent (shown dashed) to the center pixel; (c) *m*-adjacency.

In figure (b) the paths between the top right and bottom right pixels are 8-paths. And the path between the same 2 pixels in figure (c) is m-path

Connectivity

- Let R represent a subset of pixels in an image.
- A pixel pair *p* and *q* are said to be connected in *R* if there exists a path between them consisting entirely of pixels in *R*.
- A **connected component** is a maximal set of pixels in R that is connected.
- There can be more than one such set within a given R.
- Set R is a **connected set** if it has only one connected component.

Region and Boundary

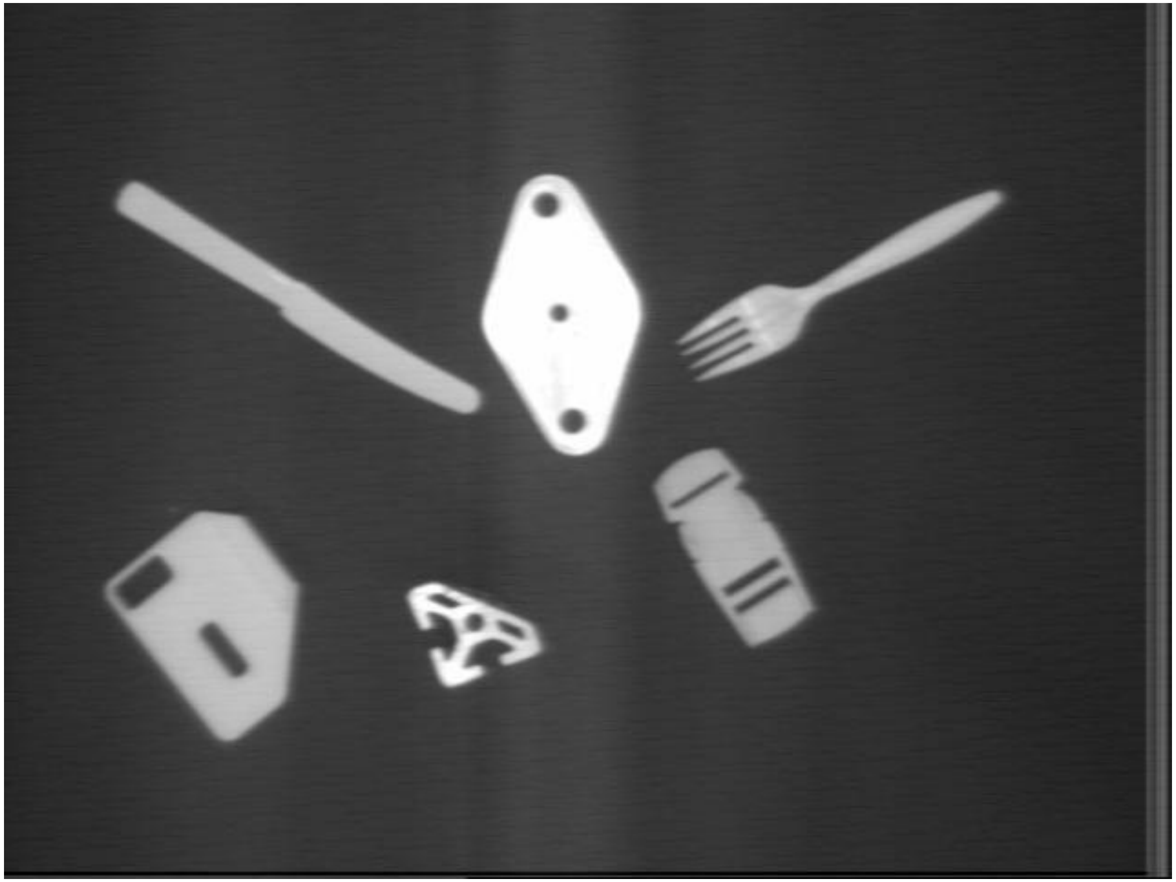
Region

Let *R* be a subset of pixels, we call *R* a region of the image if *R* is a connected set.

Boundary

The *boundary* (also called *border* or *contour*) of a region *R* is the set of pixels in the region that have one or more neighbors that are not in *R*.

Application



Need to SEGMENT image into separate COMPONENTS(regions)

Connected Components

1	1	1	0	0	0	0	0
1	1	1	0	1	1	0	0
1	1	1	0	1	1	0	0
1	1	1	0	0	0	1	0
1	1	1	0	0	0	1	0
1	1	1	0	0	0	1	0
1	1	1	0	0	1	0	0
1	1	1	0	0	0	0	0

1	1	1	0	0	0	0	0
1	1	1	0	1	1	0	0
1	1	1	0	1	1	0	0
1	1	1	0	0	0	1	0
1	1	1	0	0	0	1	0
1	1	1	0	0	0	1	0
1	1	1	0	0	1	0	0
1	1	1	0	0	0	0	0

1	1	1	0	0	0	0	0
1	1	1	0	2	2	0	0
1	1	1	0	2	2	0	0
1	1	1	0	0	0	4	0
1	1	1	0	0	0	4	0
1	1	1	0	0	0	4	0
1	1	1	0	0	3	0	0
1	1	1	0	0	0	0	0

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

a	b
c	d

FIGURE 9.19 Connected components (a) Four 4-connected components. (b) Two 8-connected components. (c) Label matrix obtained using 4-connectivity (d) Label matrix obtained using 8-connectivity.

Connected Components Labeling

First Pass

	r
t	p

- Examine pixels in scanline order
- When we visit point p,points r and t have been visited and labeled.
- If p=0: no action;
- If p=1: check r and t.

– both r and t = 0; assign new label to p;

	1					2	2				3	3	3	3	3	3	3
4	1	1			5	2	2	2			3						
	1			6	5			2	2		3		7	7	7	7	7
					5	5	5	2			3		7				7
8		9				5	5				3		7		10		7
8		9									3		7	7	10		7
8	8	9									3						7
	8										3	3	3	3	3	3	7

(a)

– only one of r and t is a 1. assign its label to p;

– both r and t are 1:

- same label => assign it to p;
- different label=> assign lowest no. to p and

Mark labels in neighbors as equivalent (they are the same.)

Connected Components Labeling

Resolve Equivalences

- **Union-Find** Structure and operations
- Obtain transitive closure using **Floyd-Warshall** (F-W) algorithm

$$4 \equiv 1$$

$$6 \equiv 5, 2$$

$$9 \equiv 8$$

$$10 \equiv 7, 3$$

Second Pass

- examine the pixels in scanline order
- Replace the label with equivalent component label

	1					2	2				3	3	3	3	3	3	3
1	1	1			2	2	2	2			3						
	1			2	2			2	2		3		3	3	3	3	3
					2	2	2	2			3		3				3
8		8				2	2				3		3		3		3
8		8									3		3	3	3		3
8	8	8									3						3
	8										3	3	3	3	3	3	3

(d)

Distance Measures

For pixels p, q and z, with coordinates (x,y), (s,t) and (v,w), respectively, D is a distance function if:

(a) $D(p,q) \geq 0$ ($D(p,q) = 0$ iff $p = q$),

(b) $D(p,q) = D(q, p)$, and

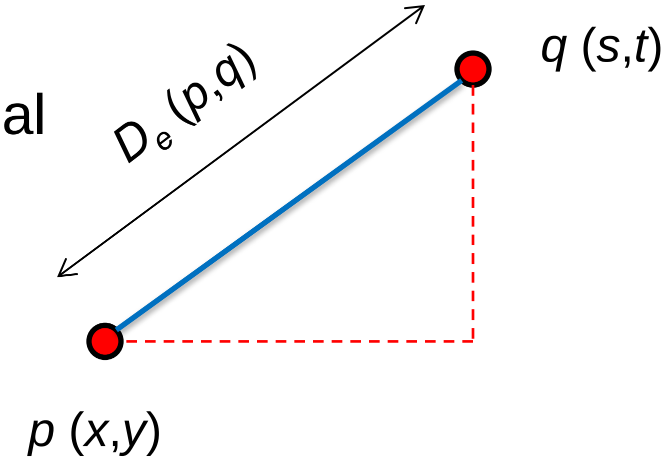
(c) $D(p,z) \leq D(p,q) + D(q,z)$.

Distance Measures

The ***Euclidean Distance*** between *p* and *q* is defined as:

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

Pixels having a 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)

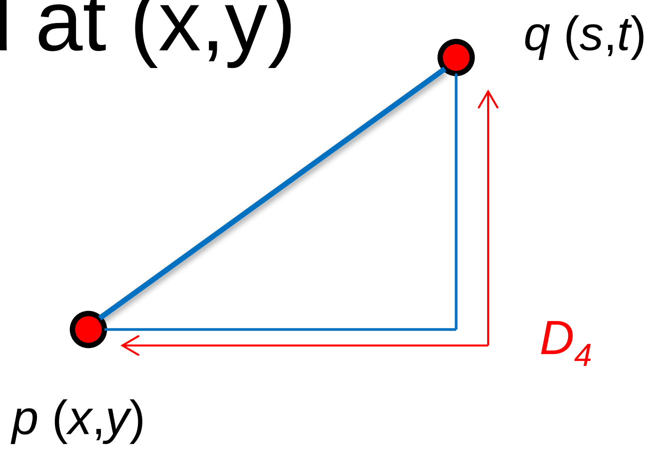


Distance Measures

The ***D_4 distance*** (also called ***city-block distance***) between *p* and *q* is defined as:

- $D_4(p,q) = | x - s | + | y - t |$
- forms a Diamond centered at (x,y)
- e.g. pixels with $D_4 \leq 2$ from p

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

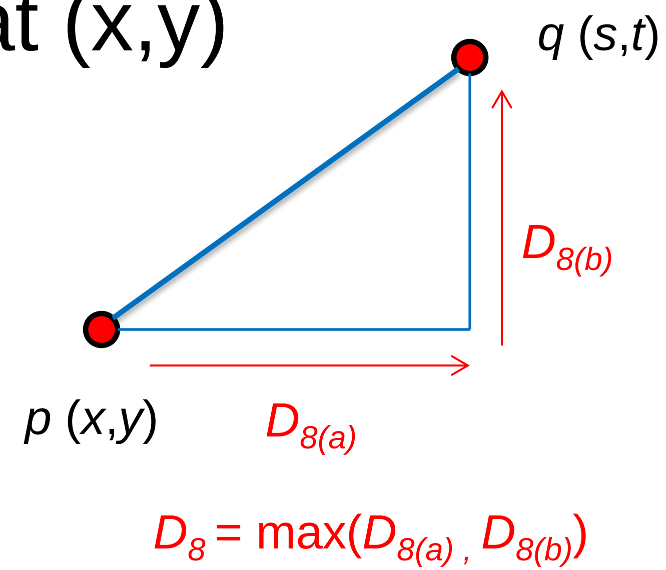


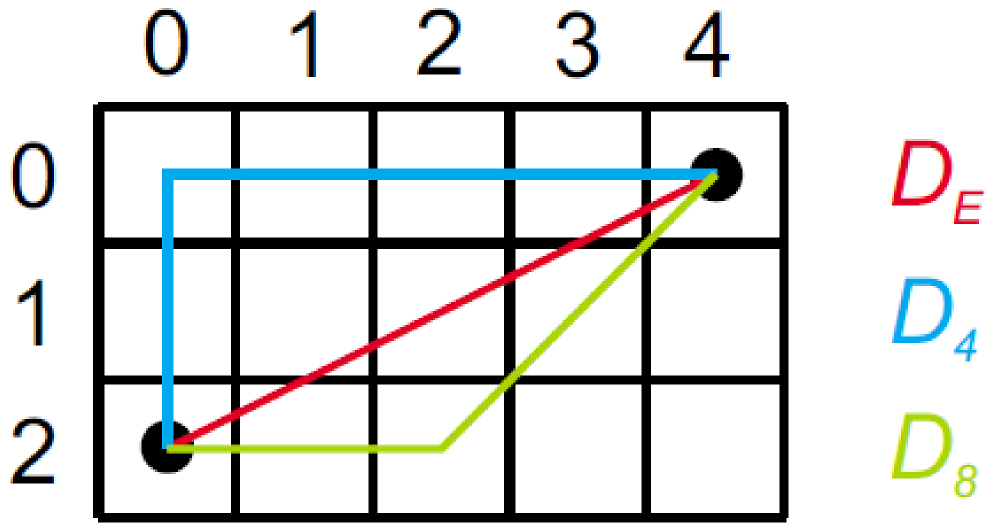
Distance Measures

The $\boldsymbol{D_8}$ ***distance*** (also called ***chessboard distance***) between *p* and *q* is defined as:

- $D_8(p,q) = \max(|x - s|, |y - t|)$
- forms a square centered at (x,y)
- e.g. pixels with $D_8 \leq 2$ from p

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




0 1 2 3 4
0
1
2
D_E
D_4
D_8

Distance Measures

D_4 and D_8 distances between p and q are independent of any paths that exist between the points

Dm distance: is defined as the shortest m-path between the points. In this case, the distance between two pixels will depend on the values of the pixels along the path, as well as the values of their neighbors.

e.g. assume $p, p_2, p_4 = 1$

p_1, p_3 = can have either 0 or 1

	p_3	p_4
p_1	p_2	
p		

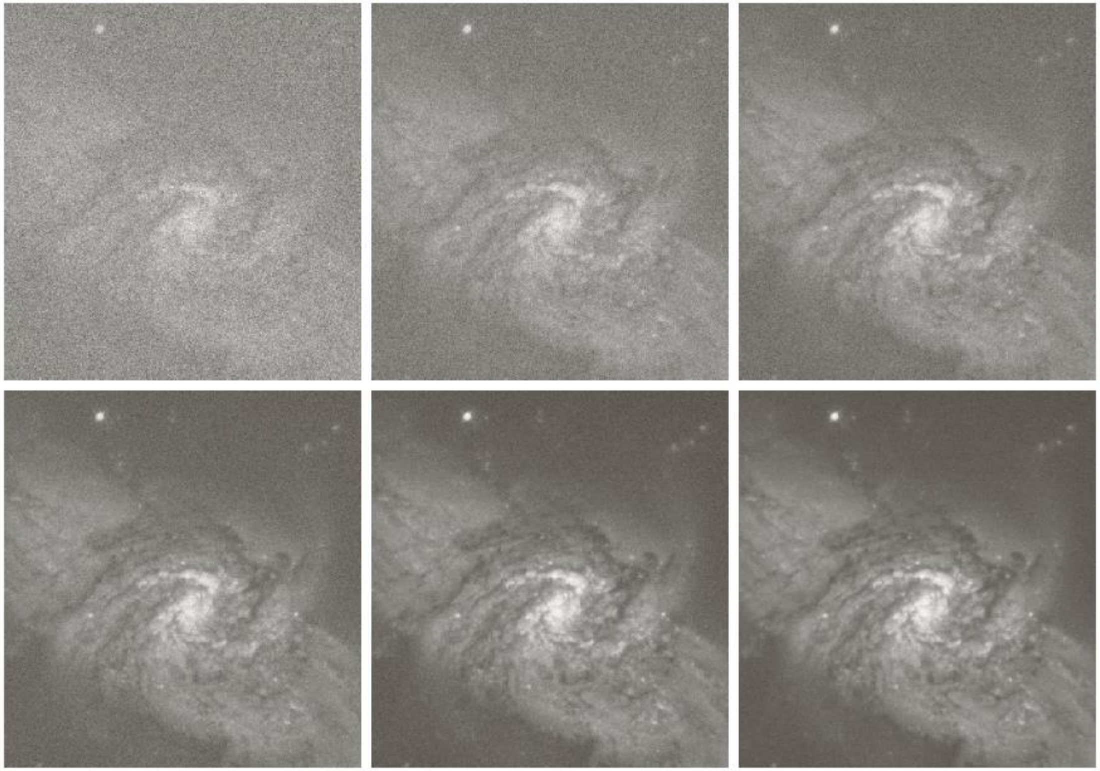
If only connectivity of pixels valued 1 is allowed, and p_1 and p_3 are 0, the m-distance between p and p_4 is 2.

If either p_1 or p_3 is 1, the distance is 3.

If both p_1 and p_3 are 1, the distance is 4 ($pp_1p_2p_3p_4$)

Basic Image Operations

Arithmetic Operations - Addition

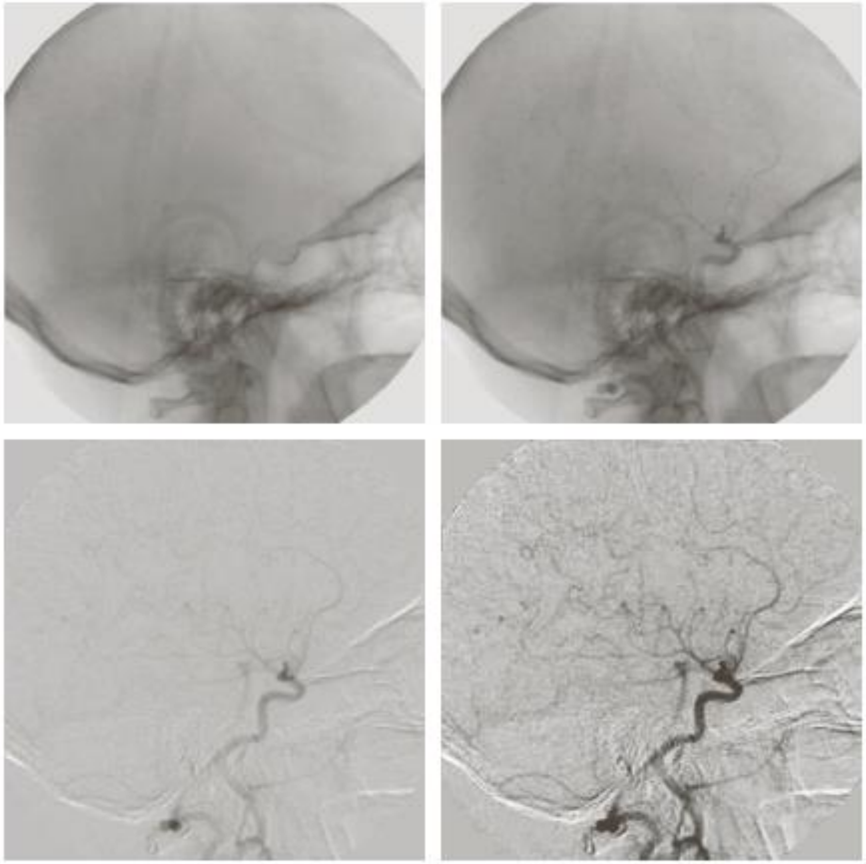


a b c
d e f

FIGURE 2.26 (a) Image of Galaxy Pair NGC 3314 corrupted by additive Gaussian noise. (b)–(f) Results of averaging 5, 10, 20, 50, and 100 noisy images, respectively. (Original image courtesy of NASA.)

Basic Image Operations

Arithmetic Operations - Subtraction



a b
c d

FIGURE 2.28 Digital subtraction angiography. (a) Mask image. (b) A live image. (c) Difference between (a) and (b). (d) Enhanced difference image. (Figures (a) and (b) courtesy of The Image Sciences Institute, University Medical Center, Utrecht, The Netherlands.)

Basic Image Operations

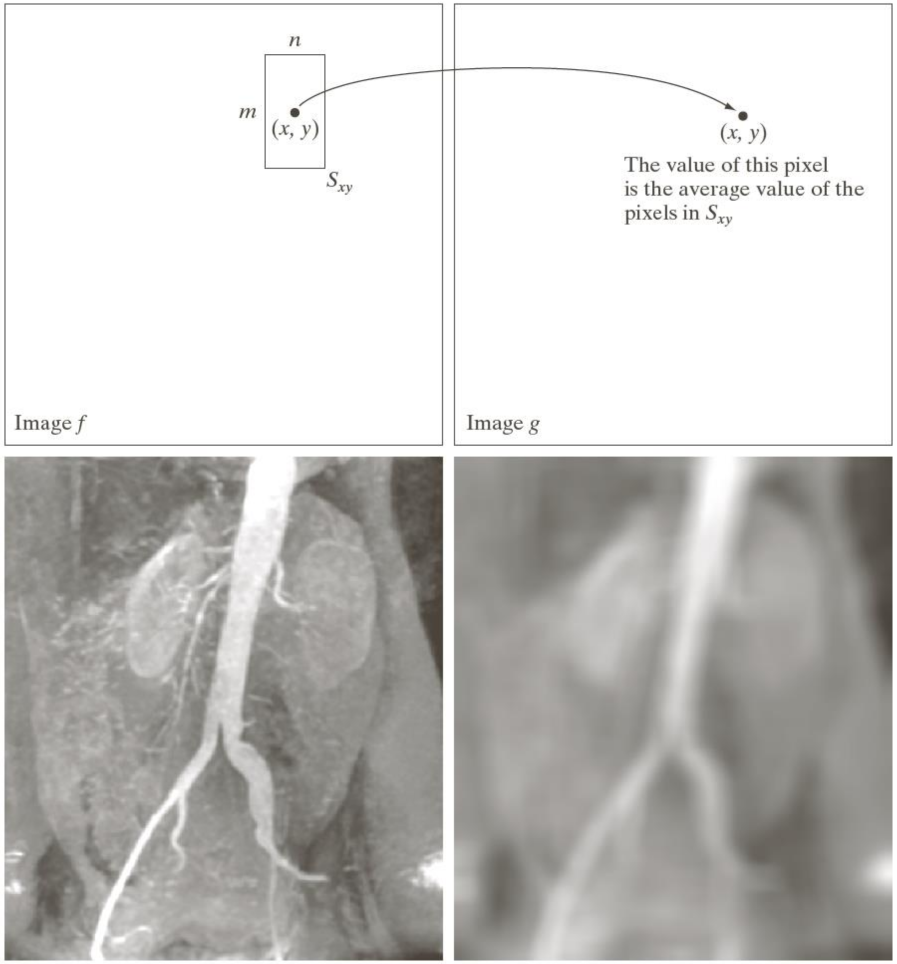
Arithmetic Operations - Multiplication

a b c

FIGURE 2.30 (a) Digital dental X-ray image. (b) ROI mask for isolating teeth with fillings (white corresponds to 1 and black corresponds to 0). (c) Product of (a) and (b).

Basic Image Operations

Spatial Operations – Neighborhood operations

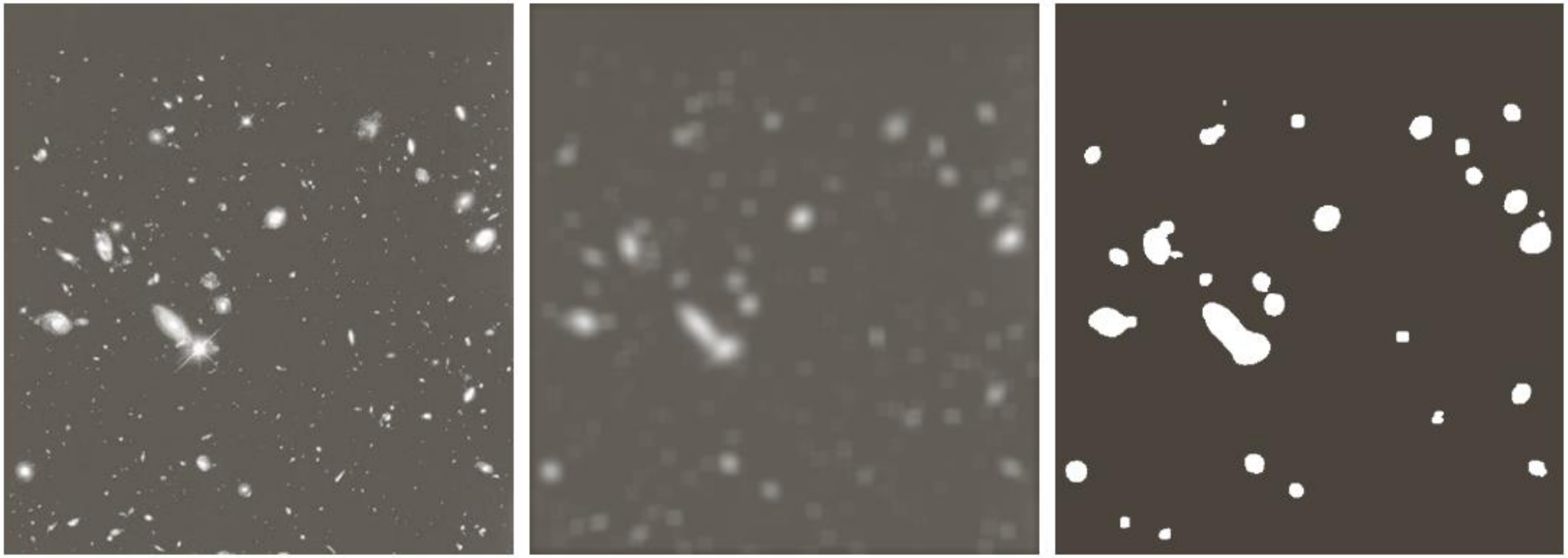


a	b
c	d

FIGURE 2.35 Local averaging using neighborhood processing. The procedure is illustrated in (a) and (b) for a rectangular neighborhood. (c) The aortic angiogram discussed in Section 1.3.2. (d) The result of using Eq. (2.6-21) with $m = n = 41$. The images are of size 790×686 pixels.

Basic Image Operations

Spatial Operations – Neighborhood operations



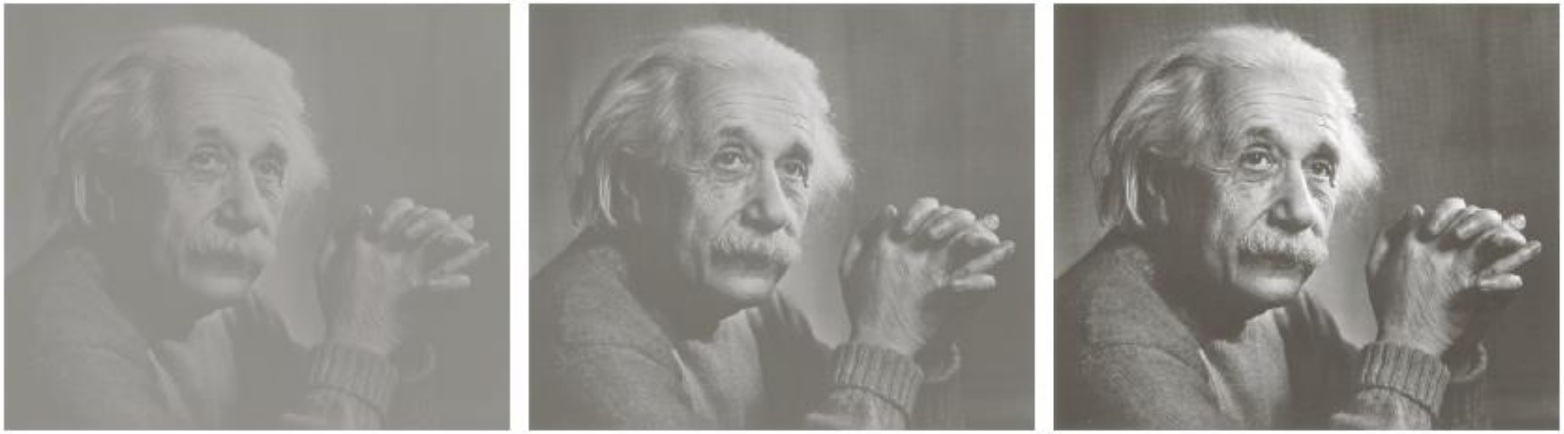
a b c

FIGURE 3.34 (a) Image of size 528 × 485 pixels from the Hubble Space Telescope. (b) Image filtered with a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

Basic Image Operations

Probabilistic methods – used in many ways.

Ex.- Treat intensity values as random variable, then the mean and variance can be useful measure of image characteristics.



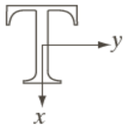
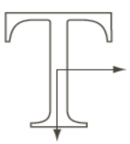
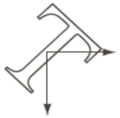
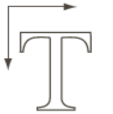
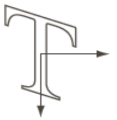
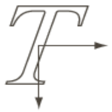
a b c

FIGURE 2.41 Images exhibiting (a) low contrast, (b) medium contrast, and (c) high contrast.

Basic Image Operations

Spatial Operations – Geometric transformations

TABLE 2.2
Affine transformations based on Eq. (2.6.–23).

Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v\cos\theta - w\sin\theta$ $y = v\cos\theta + w\sin\theta$	
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + s_v w$ $y = w$	
Shear (horizontal)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

Basic Image Operations

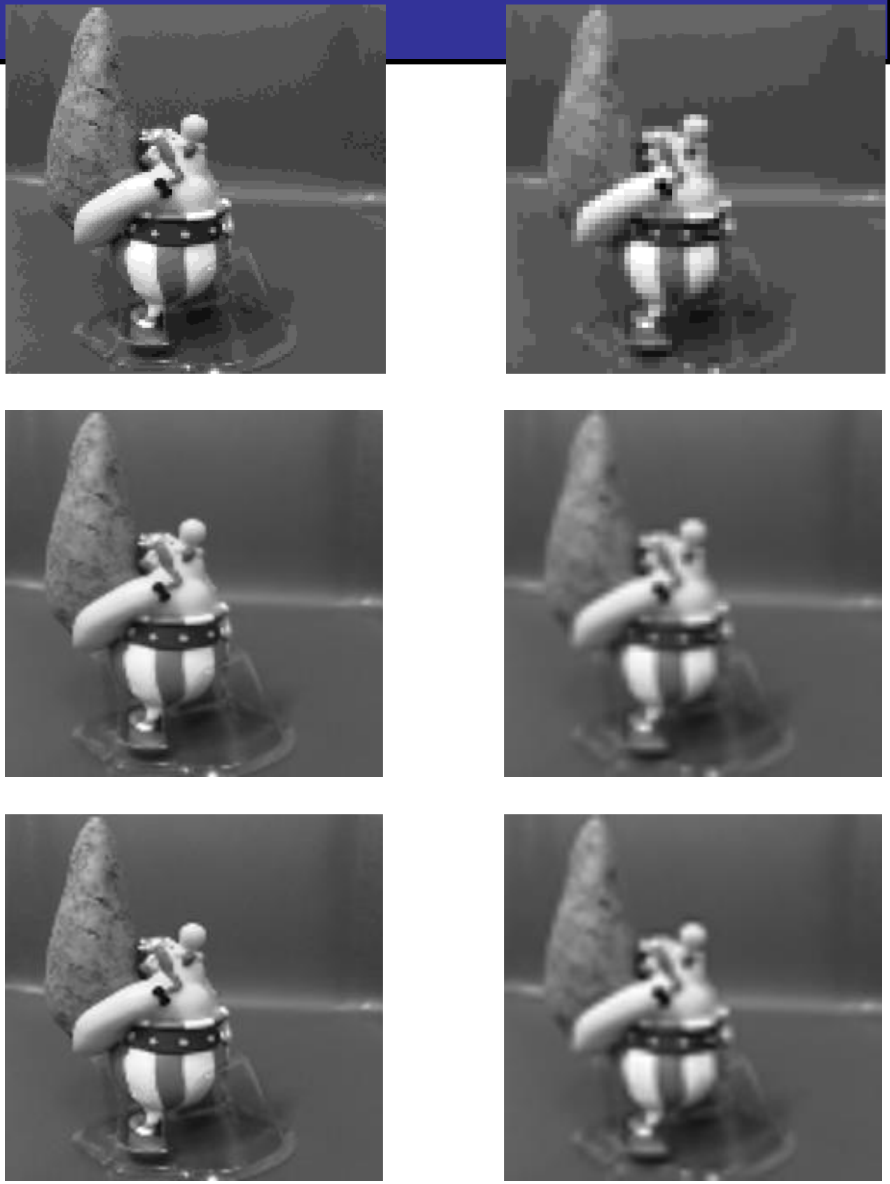
Image Resizing:

Down-sampling and

Up-sampling

Interpolation methods:

1. Nearest neighbour technique
2. Bilinear technique
3. Bicubic technique

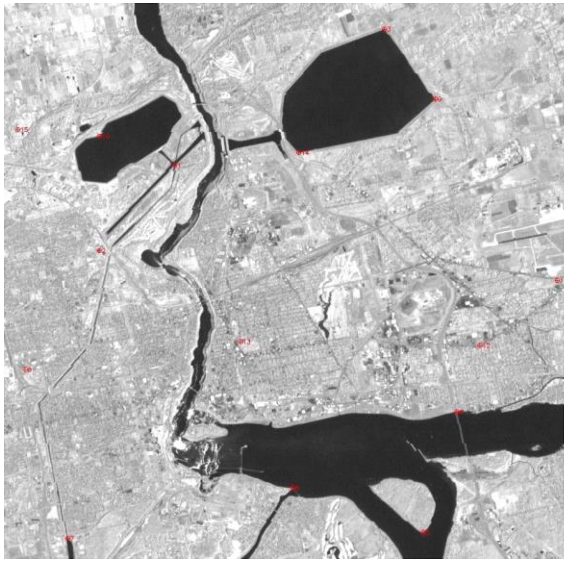


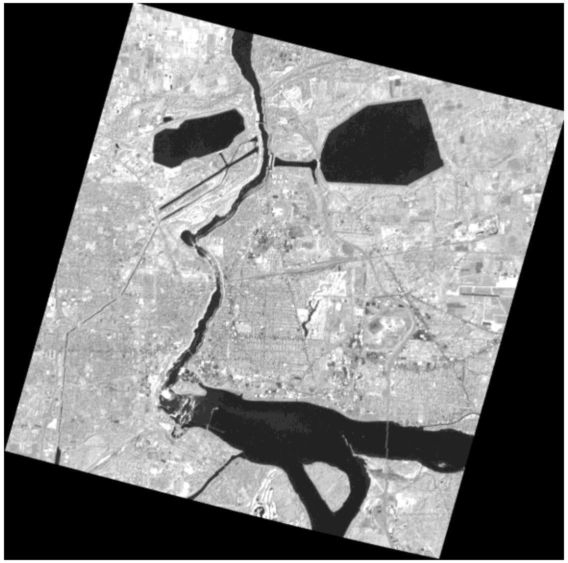
128 $\rightarrow$ 1024   64 $\rightarrow$ 1024

Basic Image Operations

Application – Image Registration

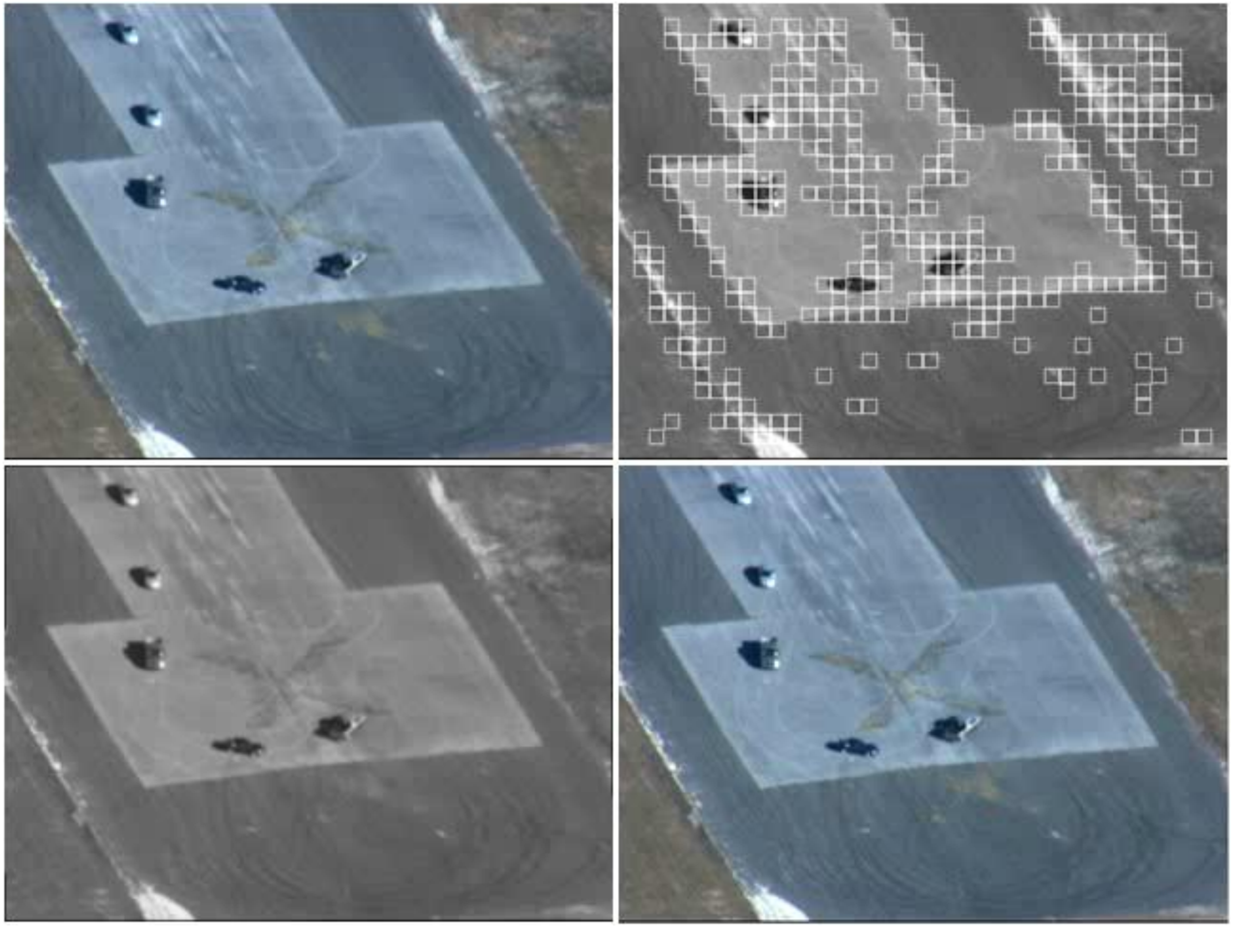
Image Registration is a process of estimating optimal transformation between two images which need to be aligned





Basic Image Operations

Application– Image Registration



Summary

We have looked at:

- Pixel neighbourhood and connectivity
- Distance measures
- Connected component labeling
- Basic Image Operations

Next time we start to look at techniques for image enhancement