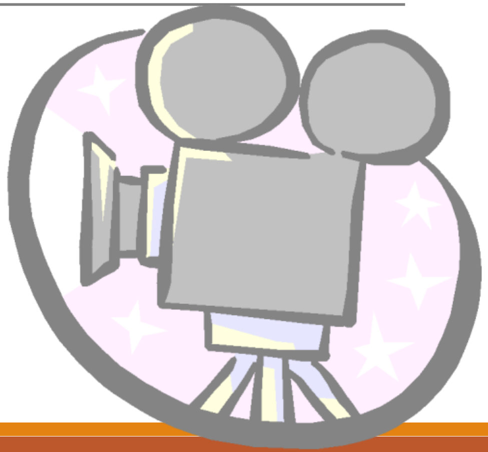
Image Processing

CS-317/CS-341



Outline

- ➢Array vs. Matrix Operations
- ➢Linear vs. Nonlinear operations
- ➢Arithmetic Operations
- ➢Logic Operations

Array vs. Matrix Operations

An array operation involving two or more images is carried out on a pixel by pixel basis. Consider the following 2 * 2 Images:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

The *array product* of these two images is

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$

Note: Assume array operation throughout any image processing operation,.
Ex- raising to an image to a power, means each individual pixel raised to that power.

On the other hand, the matrix operation is given by:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

Linear vs. Nonlinear Operations

One of the most important classification of an image processing method is whether it is linear or nonlinear. Consider a general operator, H that produces an output image, g(x, y), for a given input image, f(x, y):

$$H\left[f(x, y)\right] = g(x, y) \qquad \text{..............................(1)}$$

H is said to be linear operator if

$$\begin{aligned} H\left[a_i f_i(x, y) + a_j f_j(x, y)\right] &= a_i H\left[f_i(x, y)\right] + a_j H\left[f_j(x, y)\right] \\ &= a_i g_i(x, y) + a_j g_j(x, y) \end{aligned} \qquad \text{..............................(2)}$$

Where, a_i, a_j, $f_i(x, y)$ *and* $f_j(x,y)$ are arbitrary constants and images (of the same size), respectively.

Linear vs. Nonlinear Operations

As a simple example, suppose that H is the sum operator $\sum$, that is the function of this operator is simply to sum its inputs. To test the linearity, start with the l.h.s of eq. 2

$$\begin{aligned} \sum\left[a_i f_i(x,y) + a_j f_j(x,y)\right] &= \sum a_i f_i(x,y) + \sum a_j f_j(x,y) \\ &= a_i \sum f_i(x,y) + a_j \sum f_j(x,y) \\ &= a_i g_i(x,y) + a_j g_j(x,y) \end{aligned}$$

On the other hand, consider max operation, whose function is to find maximum value of the pixels in an image.

Note: The simplest way to prove that a given function is nonlinear, is to find an example that fails the test in eq. 2.

Consider the following two images:

$$f_1 = \begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} \quad \text{and} \quad f_2 = \begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}$$

And suppose a1=1 and a2=-1, to test linearity start with L. H.S. of eq. 2

$$\max\left\{(1)\begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} + (-1)\begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}\right\} = \max\left\{\begin{bmatrix} -6 & -3 \\ -2 & -4 \end{bmatrix}\right\}$$
$$= -2$$

Working next with the R. H.S

$$(1)\max\left\{\begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix}\right\} + (-1)\max\left\{\begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}\right\} = 3 + (-1)7$$
$$= -4$$

Arithmetic & Logic Operations

- Used extensively in Image Processing

- The arithmetic expressions between two pixels p and q are denoted as follows:

  - Addition: ***p + q***

  - Subtraction: ***p – q***

  - Multiplication: ***p*q*** (also ***pq*** and ***p x q***)

  - Division: ***p/q***

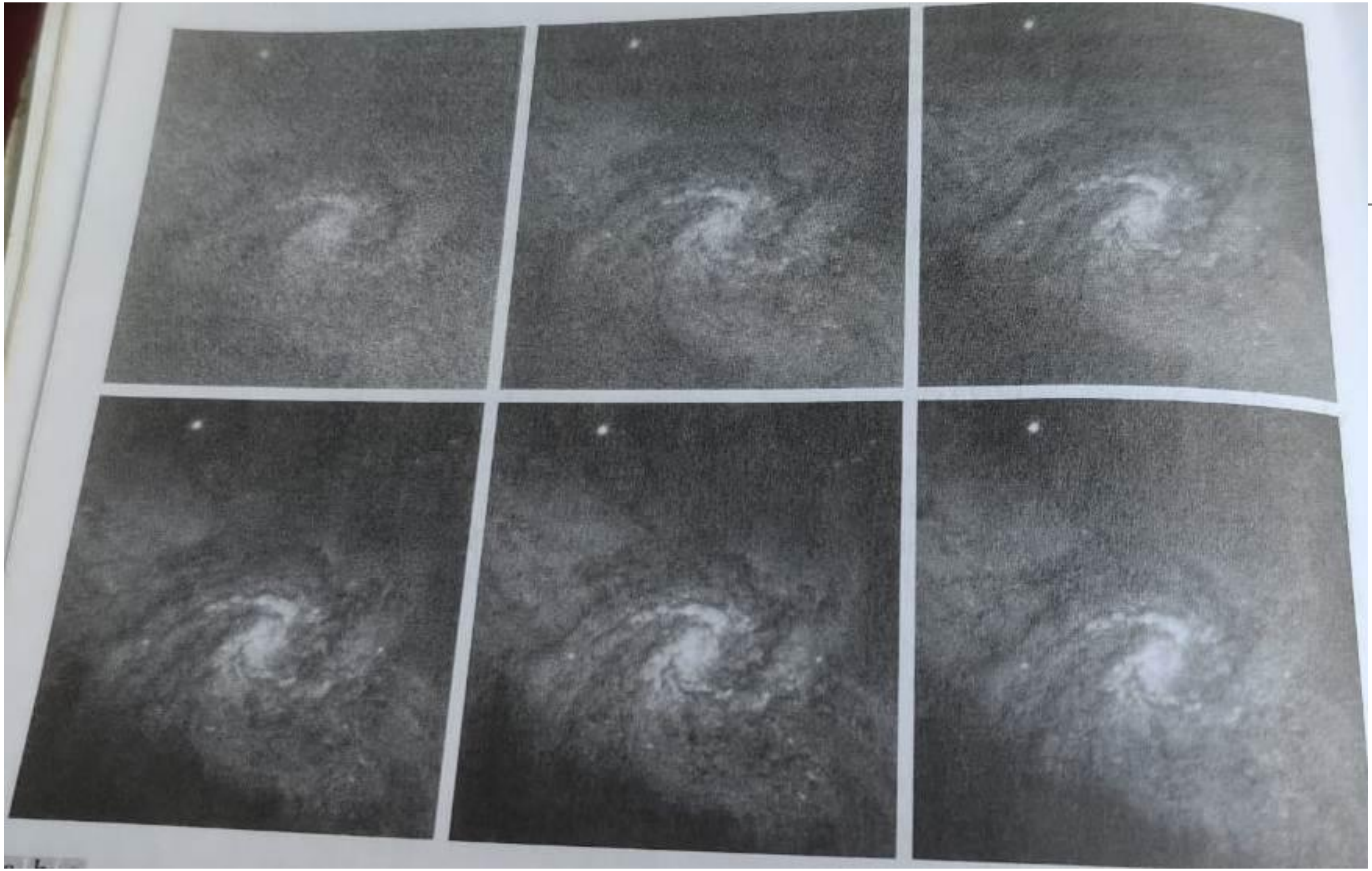
- Each operation on an entire image is carried out pixel by pixel.

- ***Image Addition*** : E.g. Image Averaging to reduce noise

- ***Image subtraction***: E.g. to remove static background information

- Image Multiplication/Division: correct gray level shading resulting from non-uniformities in illumination or in the sensor used to acquire the image.

Example

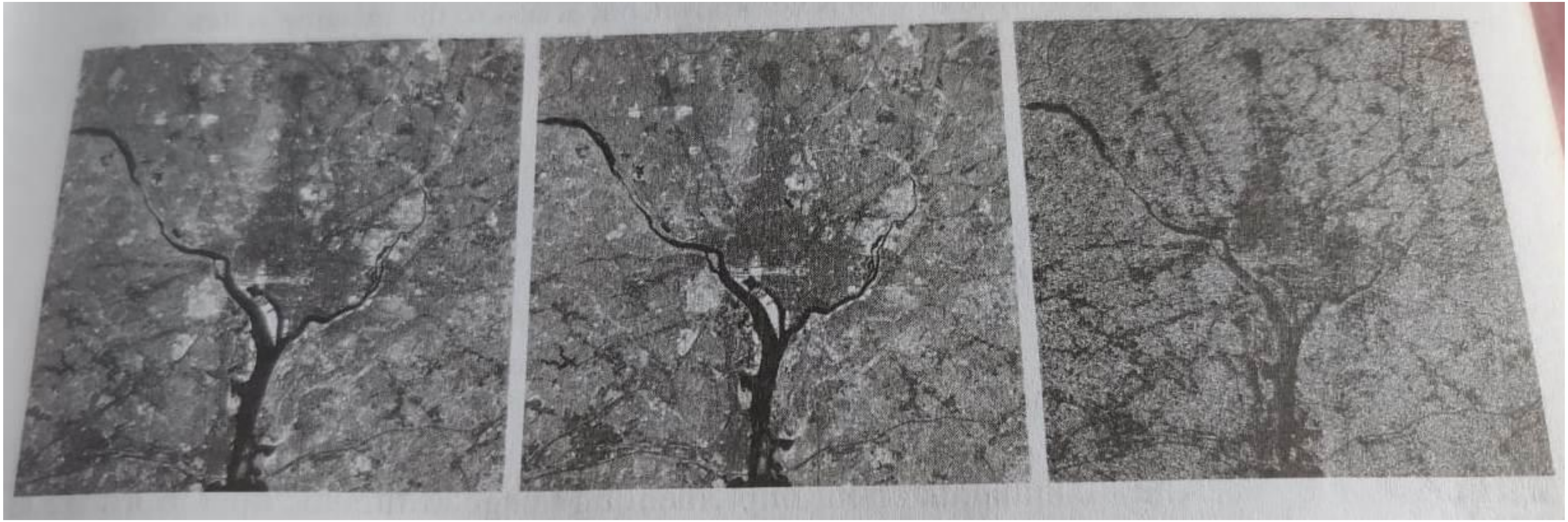


a Image of Galaxy pair NGC 3314
Corrupted by additive Gaussian noise

b-f Results of averaging 5, 10, 20, 50, 100 noisy images respectively

abc
def

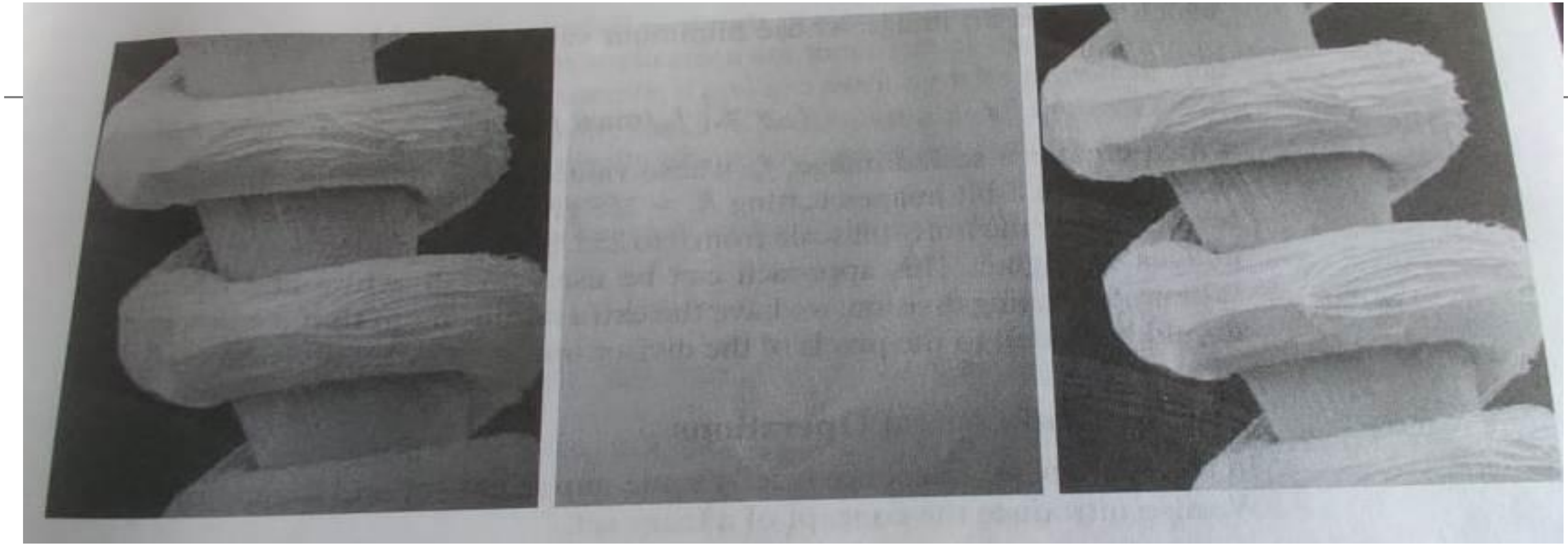
Example



(a) (b) (c)

a) Infrared image of Washington D. C. area b) image obtained by setting to zero the least significant bit of every pixel in (a)
(c) Difference of the two images, scaled to the range [0, 255 for clarity.

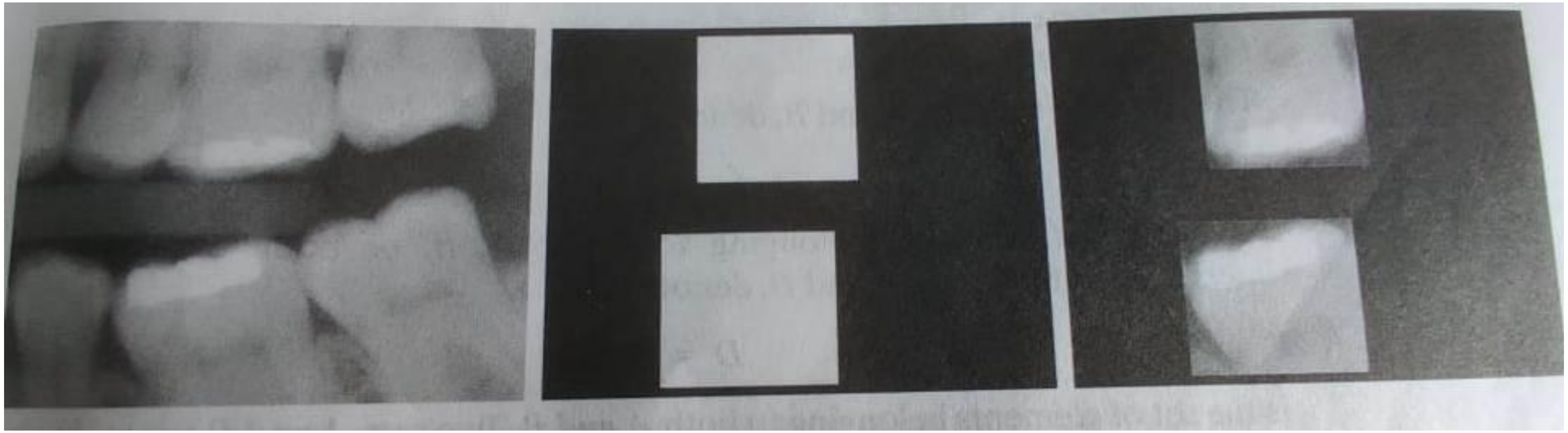
Example



(a) (b) (c)

Shading correction, (a) Shaded SEM image of a tungsten filament and support, magnified approximately 30 times
(b) The Shading pattern
(c) Product of (a) by the reciprocal of (b)

Example



(a) (b) (c)

(a) Digital Dental X-ray image.
(b) ROI mask for isolating teeth with fillings
(c) Product of (a) and (b)

Logic Operations

The principle logic operations used in Image Processing are ***AND, OR and COMPLEMENT***.

AND:	***p*** AND ***q*** (also ***p* . *q***)
OR:	***p*** OR ***q*** (also ***p* + *q***)
COMPLEMENT:	NOT ***q*** (also ***q***)

- Logic operations apply only to binary images, whereas arithmetic operations apply to multi-valued pixels.
- Logic operations are basic tools in binary image processing, where they are used for such tasks as masking, feature detection and shape analysis.
- Logic operations on entire images are done pixel by pixel.

Arithmetic & Logic Operations (A & L operations) – Neighborhood-oriented

- A& L operations are used ***neighbourhood oriented operations*** (in addition to pixel-by-pixel operations.

- Typically this takes the form of ***mask*** operations.
- The terms ***template, window*** and ***filter*** also are often used to denote a ***mask)***

Mask Operation: Let the value assigned to a pixel be a function of its gray level and gray level of its neighbours.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

3 x 3 Sub Image

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

Replacing the value of z_5 with the average value of the pixels in 3 x 3 region centered at the pixel with value z_5

A & L operations ...

The same operation above, in generic terms, would be to replace a pixel with a weighted average of pixels in the region where weights (coefficients) can be different for different pixels.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

$$\mathbf{z} = (\mathbf{w}_1\mathbf{z}_1 + \mathbf{w}_2\mathbf{z}_2 + \cdots + \mathbf{w}_9\mathbf{z}_9)$$

$$= \sum_{i=1}^{9} \mathbf{w}_i\mathbf{z}_i$$

- 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, such as noise reduction, region thinning, and edge detection.
- Computationally expensive. E.g applying a 3x3 mask to a 512x512 image requires nine multiplications and eight additions at each pixel location (total 2,359,296 multiplications and 2,097,152 additions)

Suggested Readings

- **Digital Image Processing by Rafel Gonzalez, Richard Woods, Pearson Education India, 2017.**

- **Fundamental of Digital image processing by A. K Jain, Pearson Education India, 2015.**

Thank you