Image Processing and Pattern Recognition

1

Lecture – 03

Harimohan Khatri

2

In this lecture…

i. Gray level transformations
   i. Point Operations
   ii. Contrast stretching
   iii. Thresholding
   iv. Digital Negative
   v. Intensity level slicing
   vi. Bit Plane slicing

3

Gray Level Transformations

- The spatial domain processes can be expressed as;

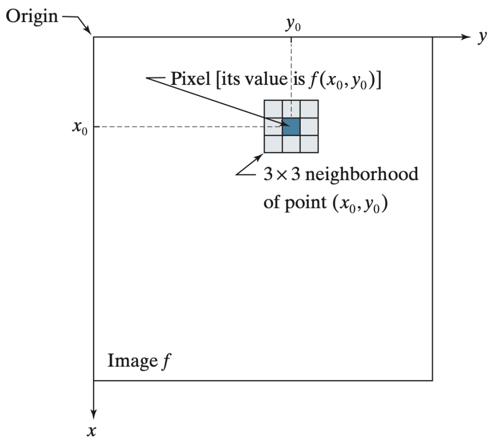
$$g(x,y) = T[f(x,y)]$$

where

f(x,y) is an input image,

g(x,y) is the output image, and

T is an operator on f defined over a neighborhood of point (x,y).



4

Point Operations

- The smallest possible neighborhood is of size 1×1 . In this case, ***g*** depends only on the value of ***f*** at a single point (x,y) and T becomes an intensity transformation function of the form

$$s = T(r)$$

Where **r** is the input gray level, **T** is the transformation function, and **s** is the output gray level.

- Approaches whose results depend only on the intensity at a point are called point processing techniques.

5

Negatives

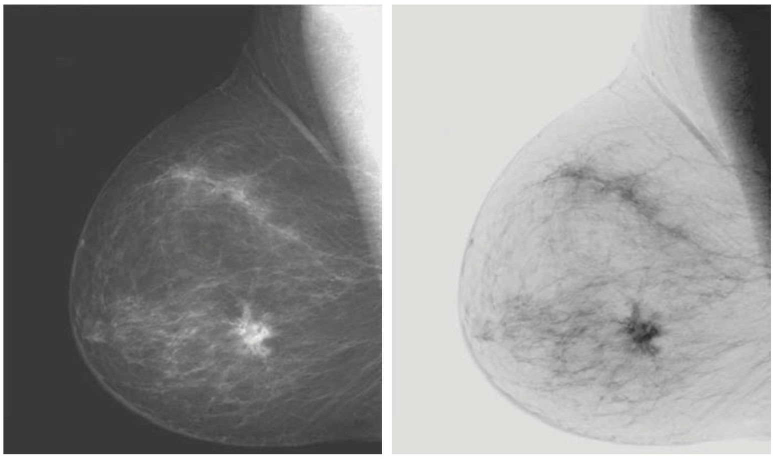
- The negative of an image with intensity levels in the range [0,L−1] is obtained by using the negative transformation function which has the form:

$$s = L - 1 - r$$

- This type of processing is used, for example, in enhancing white or gray detail embedded in dark regions of an image, especially when the black areas are dominant in size.

6

Negatives



(a) A Digital mammogram. (b) Negative image

7

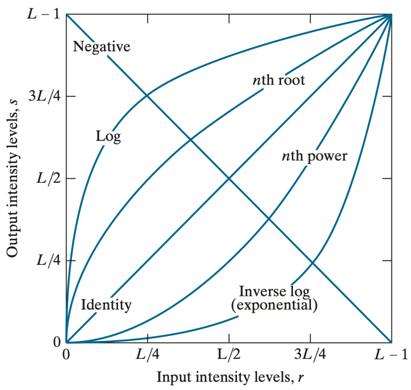
Log Transformations

- The general form of the log transformation is

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

where c is a constant and it is assumed that $r \geq 0$. We usually set c to 1, Grey levels must be in the range [0.0, 1.0]

- This transformations maps a narrow range of low intensity values in the input into a wider range of output levels. For example, how input levels in the range [0,L/4] map to output levels to the range [0, 3L/4].
- Conversely, higher values of input levels are mapped to a narrower range in the output.

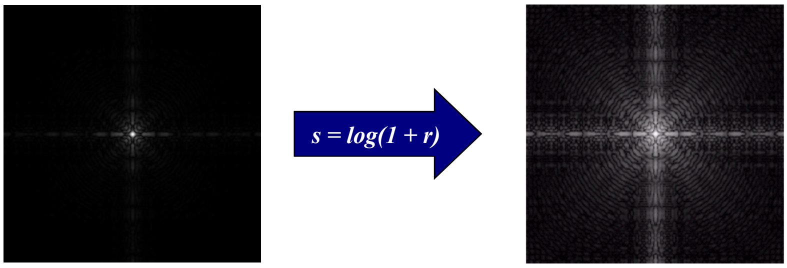


Some basic intensity transformation functions.

8

Log Transformations

- 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



9

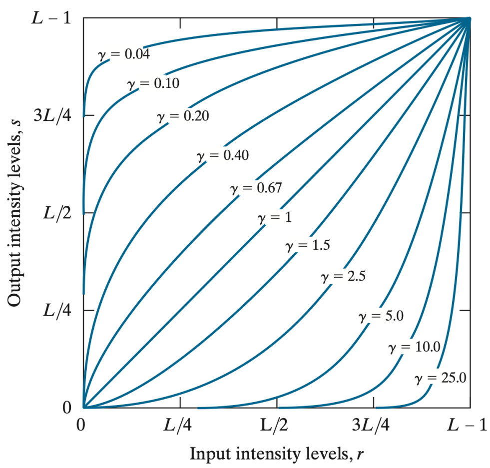
Power Law (Gamma) Transformations

- Power-law transformations have the form

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

where c and γ are positive constants.

- The response of many devices used for image capture, printing, and display obey a power law.
- By convention, the exponent in a power-law equation is referred to as.
- The process used to correct these power-law response phenomena is called gamma correction.



10

Power Law (Gamma) Transformations

Original Image

$s = r^{0.3}$

$s = r^{0.4}$

$s = r^{0.6}$

11

Power Law (Gamma) Transformations

Original Image

$s = r^{3.0}$

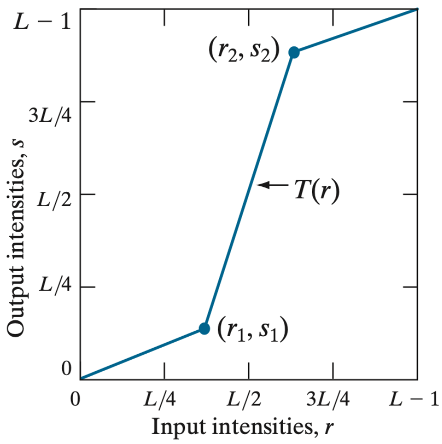
$s = r^{4.0}$

$s = r^{5.0}$

12

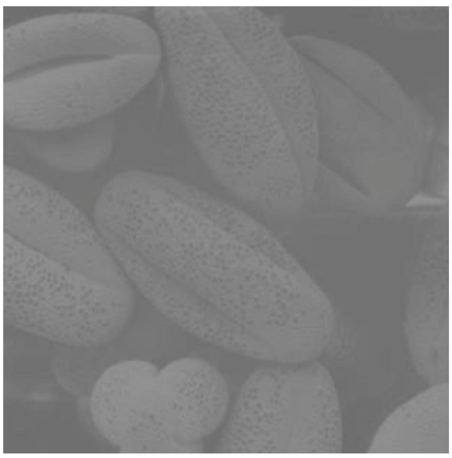
Contrast Stretching

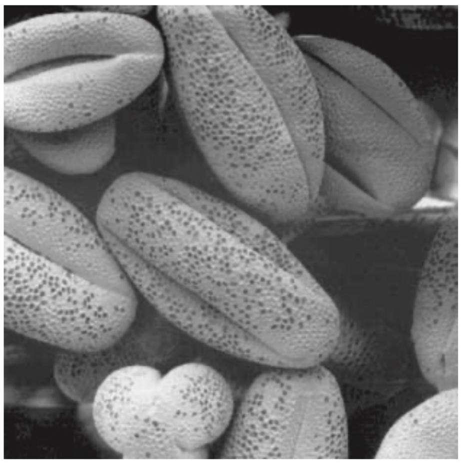
- Contrast stretching expands the range of intensity levels in an image so that it spans the ideal full intensity range.
- The locations of points (r_1,s_1) and (r_2,s_2) control the shape of the transformation function.
- Contrast stretching can be obtained by setting $(r_1,s_1) = (r_{min},0)$ and $(r_2,s_2)=(r_{max}, L-1)$ where r_{min} and r_{max} denote the minimum and maximum intensity levels in the input image respectively.
- The transformation stretches the intensity levels linearly to the full intensity range, [0,L-1].



13

Contrast Stretching



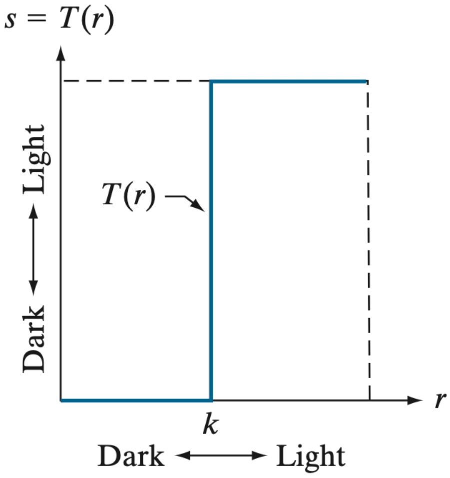


(a) A low-contrast electron microscope image of pollen, magnified 700 times. (b) Result of contrast stretching

14

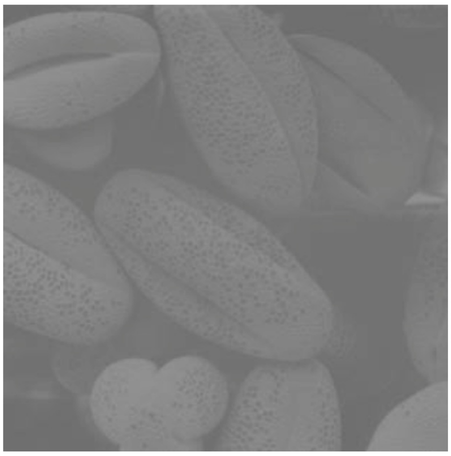
Thresholding

- If r1 = r2, s1 = 0, and s2 = L−1 the transformation becomes a thresholding function that creates a binary image.
- Choose a threshold value T, and for each pixel in the grayscale image:
  - If the pixel intensity is greater than T, set it to 255 (white).
  - If it is less than or equal to T, set it to 0 (black).



15

Thresholding

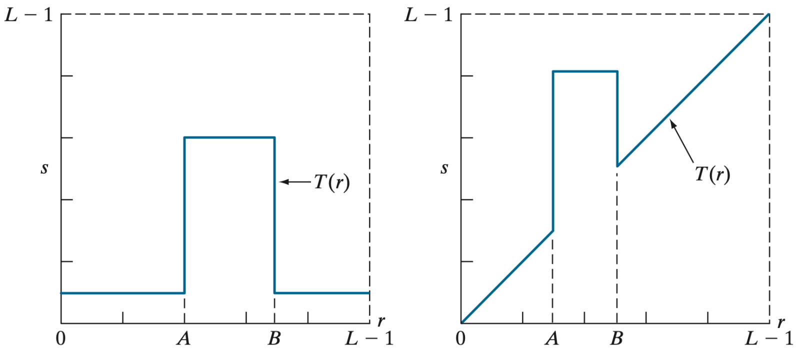


Result of thresholding.

16

Intensity Level Slicing

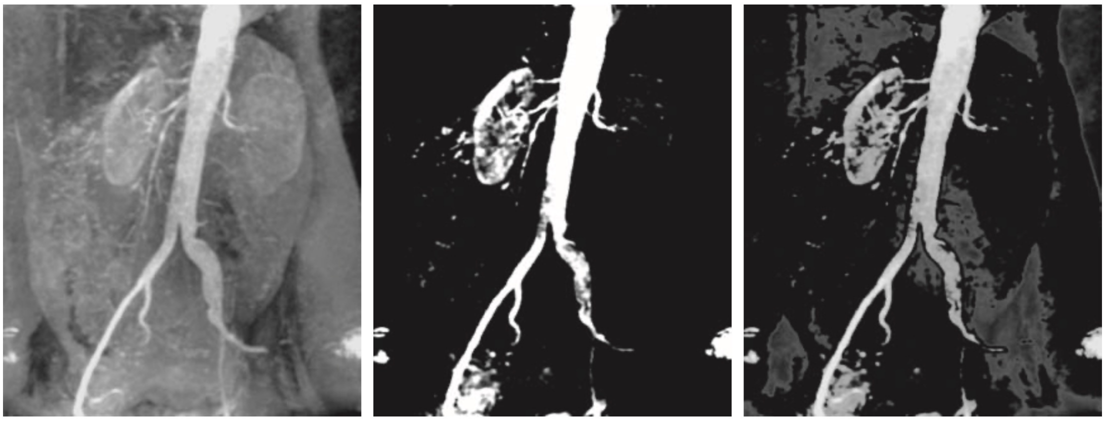
- There are applications in which it is of interest to highlight a specific range of intensities in an image.



(a) This transformation Function highlights range [A,B] and reduces all other intensities to a lower level.
(b) This function highlights range [A,B] and leaves other intensities unchanged.

17

Intensity Level Slicing



(a) Aortic angiogram. (b) Result of slicing transformation with the range of intensities of interest selected in the upper end of the gray scale. (c) Result of transformation with the selected range set near black, so that the grays in the area of the blood vessels and kidneys were preserved.

18

Bit Plane Slicing

- A grayscale image typically has 8 bits per pixel, meaning each pixel value ranges from 0 to 255 and can be represented in binary as 8 bits (e.g., 11001101).
- Each of these 8 bits forms a bit plane:

  Bit plane 0: least significant bit (LSB)

  ...

  Bit plane 7: most significant bit (MSB)
- **Higher bit planes (e.g., 6 or 7):** Carry most of the visual information.
- **Lower bit planes (e.g., 0 or 1):** Contain finer details or noise; sometimes useful for watermarking or steganography.

19

Bit Plane Slicing

20

Bit Plane Slicing

Bit plane - 7

21

Bit Plane Slicing

Bit plane - 6

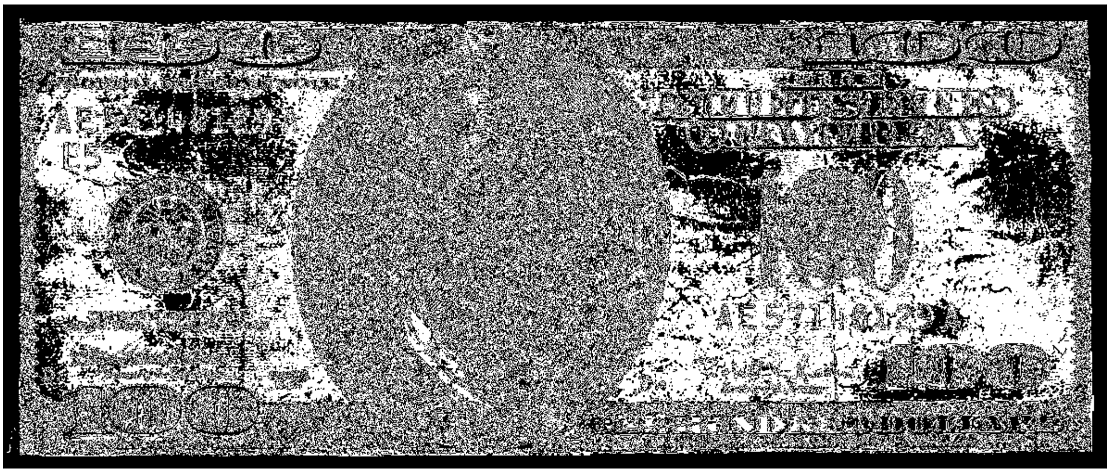
22

Bit Plane Slicing

Bit plane - 5

23

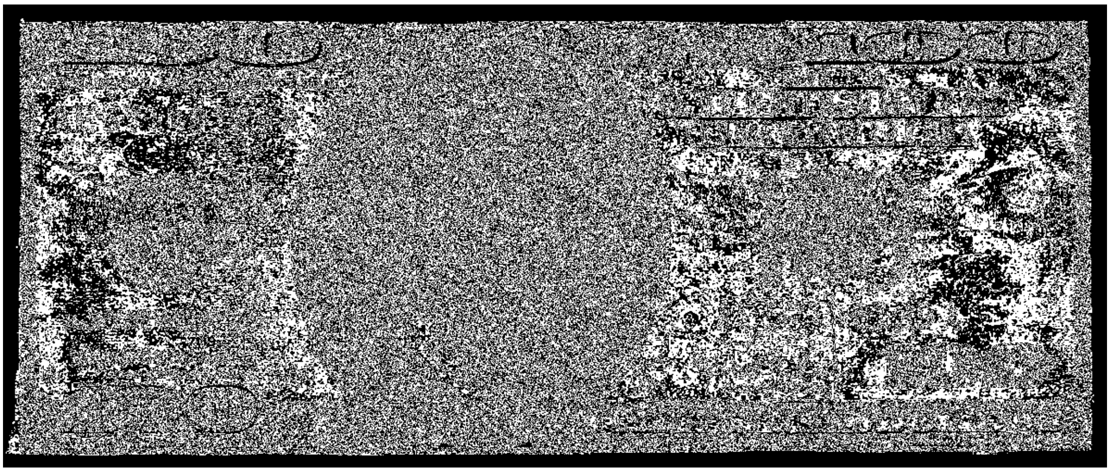
Bit Plane Slicing



Bit plane - 4

24

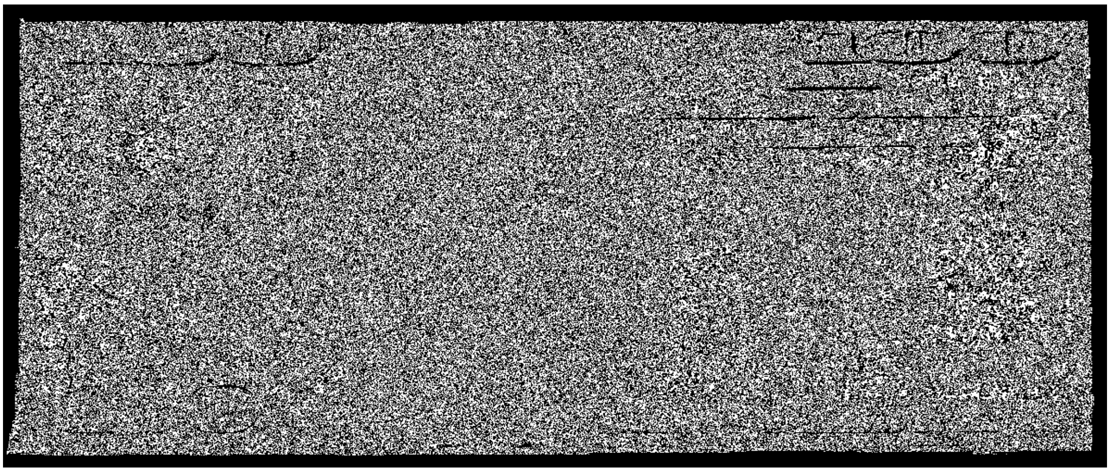
Bit Plane Slicing



Bit plane - 3

25

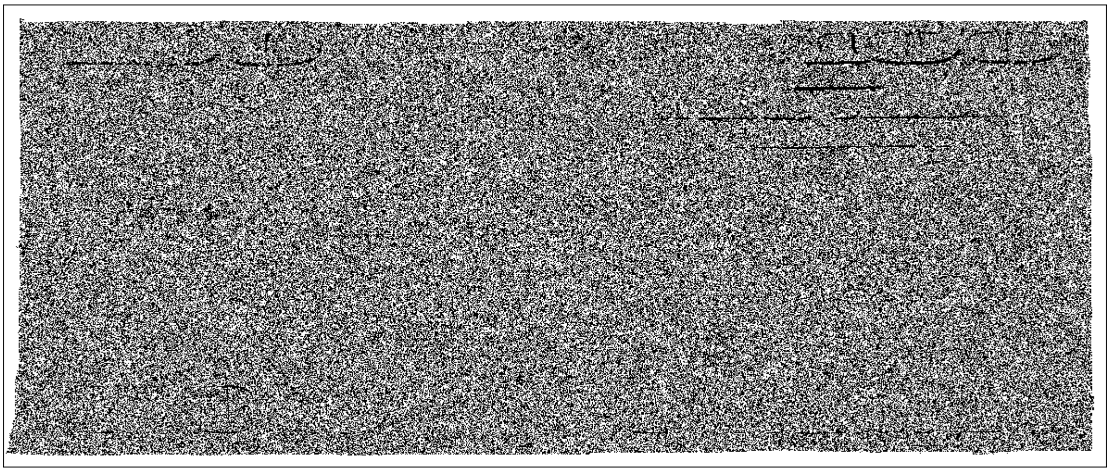
Bit Plane Slicing



Bit plane - 2

26

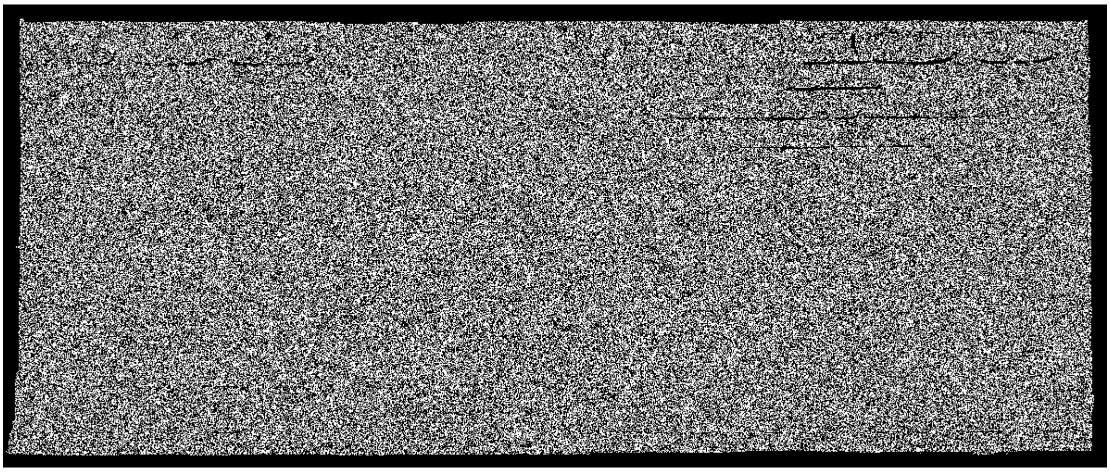
Bit Plane Slicing



Bit plane - 1

27

Bit Plane Slicing



Bit plane - 0

28

Bit Plane Slicing

- Bit plane slicing can be used in the field of image reconstruction and steganography.

Image reconstructed from bit planes: (a) 8 and 7; (b) 8, 7, and 6; (c) 8, 7, 6, and 5

29

Lecture – 04

Harimohan Khatri

30

In this lecture…

i. Histogram plotting of an image
ii. Histogram equalization
iii. Histogram Matching

31

Histogram of an Image

- Let r_k, for k = 0,1,2,…,L-1, denote the intensities of a L-level digital image, f(x,y). The unnormalized histogram of f is defined as

$$h(r_k) = n_k \qquad \text{for } k = 0,1,2,\ldots,L-1$$

where n_k is the number of pixels in *f* with intensity r_k, and the subdivisions of the intensity scale are called histogram bins.

- Similarly, the normalized histogram of f is defined as

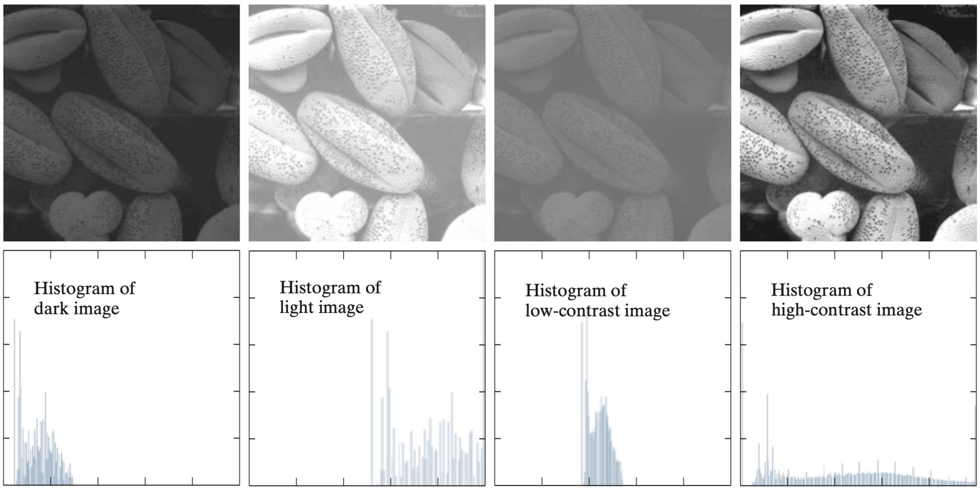
$$p(r_k) = \frac{h(rk)}{MN} = \frac{n_k}{MN}$$

where M and N are the number of image rows and columns respectively.

- Histogram shape is related to image appearance.

32

Histogram of an Image



33

Histogram Equalization

- Histogram equalization is a process for increasing the contrast in an image by spreading the histogram out to be approximately uniformly distributed
- The gray levels of an image that has been subjected to histogram equalization are spread out and always reach white
  - The increase of dynamic range produces an increase in contrast
- For images with low contrast, histogram equalization has the adverse effect of increasing visual graininess

34

Histogram Equalization

- The intensity transformation function we are constructing is of the form

  $s=T(r) \qquad 0 \le r \le L-1$
- An output intensity level s is produced for every pixel in the input image having intensity r
- We assume T(r) is monotonically increasing in the interval $0 \le r \le L-1$

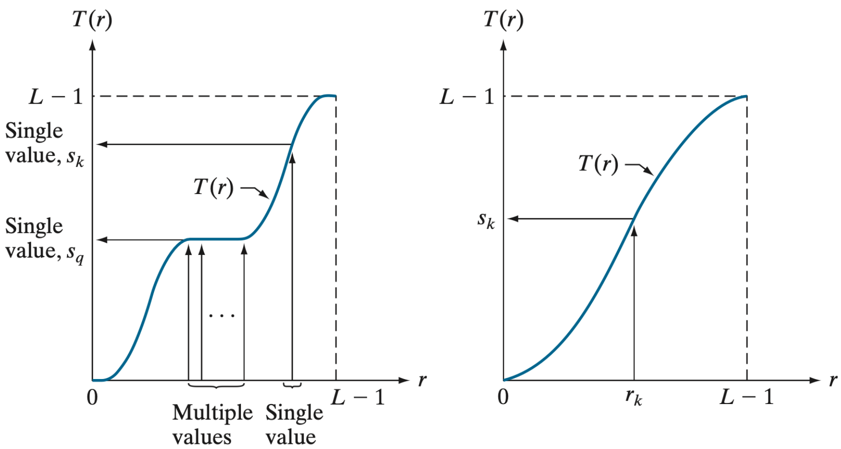
  $0 \le T(r) \le L-1 \qquad \text{for } 0 \le r \le L-1$
- If we define the inverse

  $r=T^{-1}(s) \qquad 0 \le s \le L-1$

  Then T(r) should be strictly monotonically increasing

35

Histogram Equalization



(a) Monotonic increasing function, (multiple values can map to a single value.)
(b) Strictly monotonic increasing function (one-to-one mapping, both ways)

36

Histogram Equalization

- Histogram equalization requires construction of a transformation function s_k

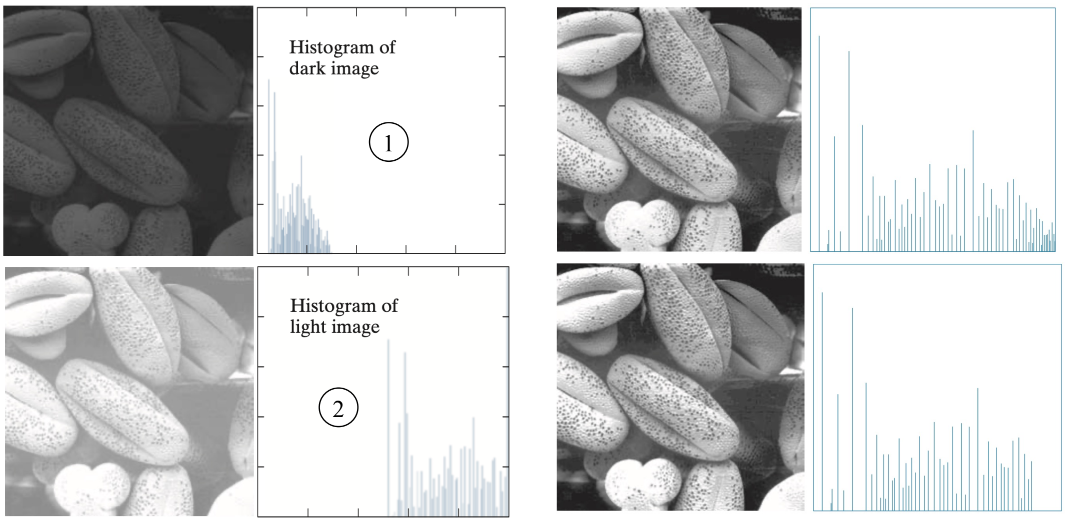
$$s_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{M \times N} \qquad s_k = T(r_k) = \frac{(L-1)}{M \times N}\sum_{j=0}^{k} n_j$$

where r_k is the kth gray level, n_k is the number of pixels with that gray level, MxN is the number of pixels in the image, and k=0,1,...,L-1

- This yields an s with as many elements as the original image's histogram (normally 256 for our test images)
- The values of s will be in the range [0,1]. For constructing a new image, s would be scaled to the range [1,256]

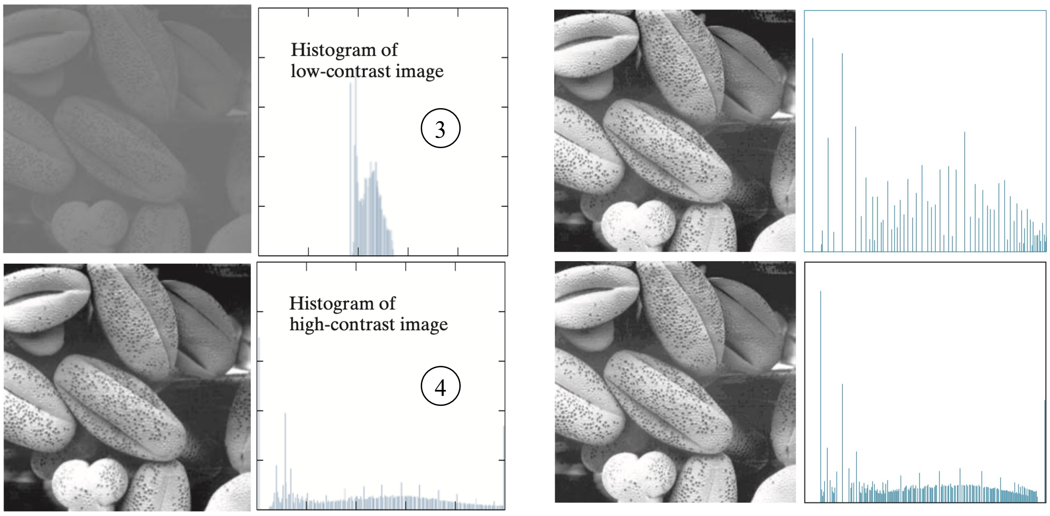
37

Histogram Equalization



38

Histogram Equalization



39

Histogram Equalization

- Spreading out the frequencies in an image (or equalizing the image) is a simple way to improve dark or washed out images
- The formula for histogram equalization is given as

$$\boxed{\begin{aligned} s_k &= T(r_k) \\ &= \sum_{j=1}^{k} p_r(r_j) \\ &= \sum_{j=1}^{k} \frac{n_j}{n} \end{aligned}}$$

where

r_k: input intensity

s_k: processed intensity

k: the intensity range (e.g 0.0 – 1.0)

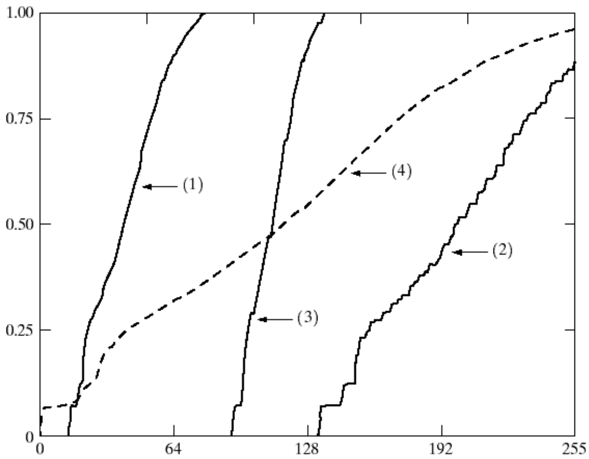
n_j: the frequency of intensity j

n: the sum of all frequencies

40

Histogram Equalization

- The functions used to equalize the images in the previous example



41

Exercise – 01

- Perform the histogram equalization for the following image.

Gray Levels	0	1	2	3	4	5	6	7
No of Pixels	11	8	12	10	6	2	4	1

42

Exercise – 02

- Perform the histogram equalization for the following image.

Gray Levels	0	1	2	3	4	5	6	7
No of Pixels	0	4	11	15	8	2	0	0

Exercise – 03

- Perform the histogram equalization for the following image.

2	1	1	2	2
2	5	0	0	0
2	2	1	1	0
1	2	1	2	2
5	5	2	1	1

6	4	4	6	6
6	7	1	1	1
6	6	4	4	1
4	6	4	6	6
7	7	6	4	4

Exercise – 04

- Perform the histogram equalization for the following image.

2	1	1	2	2	1	1	2
2	13	13	0	0	0	2	3
2	12	12	13	0	1	9	4
1	12	11	12	12	10	10	9
1	5	12	10	10	11	11	9
1	5	5	5	6	9	10	9
2	4	5	6	7	7	9	6
2	4	4	5	5	6	6	5

45

Histogram Matching (Specification)

- Method that generates images by changing the histogram of input image by specified histogram is called histogram matching or histogram specification.
- Consider r and z denote the intensity levels of the input and output images.
- We can estimate p(r) from the given input image, and p(z) is the specified PDF that we wish the output image to have
- In discrete form the histogram equalization function is given as:

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) \qquad k = 0,1,2,\ldots,L-1$$

46

Histogram Matching (Specification)

- Similarly, given a specific value of s_k , the discrete formulation for computing the transformation function

$$G(z_q) = (L-1)\sum_{i=0}^{q} p_z(z_i)$$

- for a value of q in histogram matching,

$$G(z_q) = s_k$$

where $p_z(z_i)$ is the ith value of the specified histogram.

- Finally, we obtain the desired value z_q from the inverse transformation:

$$z_q = G^{-1}(s_k)$$

47

Process for Histogram Specialization

1. Compute the histogram, $p_r(r)$, of the input image, and use

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) \qquad k = 0,1,2,\ldots,L-1$$

to map the intensities in the input image to the intensities in the histogram-equalized image. Round the resulting values, s_k , to the integer range [0,L-1].

2. Compute all values of function $G(Z_q)$ using the Eq.

$$G(z_q) = (L-1)\sum_{i=0}^{q} p_z(z_i)$$

for q= 0,1,2, …,L-1 where $p_z(z_i)$ are the values of the specified histogram. Round the values of G to integers in the range [0,L-1]. Store the rounded values of G in a lookup table.

48

Process for Histogram Specialization

3. For every value of s_k, k= 0,1,2,…,L-1 use the stored values of G from Step 2 to find the corresponding value of z_q so that $G(z_q)$ is closest to s_k. Store these mappings from s to z. When more than one value of z_q gives the same match (i.e., the mapping is not unique), choose the smallest value by convention.
4. Form the histogram-specified image by mapping every equalized pixel with value s_k to the corresponding pixel with value z_q in the histogram-specified image, using the mappings found in Step 3.

49

Discussion and Classwork

- Compare Histogram Equalization and Histogram Specification.
- For given histogram (a) and (b) modify histogram as given by histogram (b).

a)

Gray Levels	0	1	2	3	4	5	6	7
No of Pixels	11	8	12	10	6	2	4	1

b)

Gray Levels	0	1	2	3	4	5	6	7
No of Pixels	0	4	11	15	8	2	0	0

50

Lecture – 05

Harimohan Khatri

51

In this lecture…

i. Enhancement using Arithmetic and Logical Operations
ii. Spatial Filters
iii. Correlation and Convolution

52

Elementwise vs Matrix Operations

- The elementwise product is obtained by multiplying pairs of corresponding pixels. Whereas the matrix product of the images is formed using the rules of matrix multiplication:
- For example, consider the following 2*2 images (matrices):

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

- The elementwise product of these two images is:

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

53

Elementwise vs Matrix Operations

- The matrix product of the images is formed using the rules of matrix multiplication:

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

54

Arithmetic Operations

- Arithmetic operations between two images f(x,y) and g(x,y) are denoted as

$$s(x,y) = f(x,y) + g(x,y)$$
$$d(x,y) = f(x,y) - g(x,y)$$
$$p(x,y) = f(x,y) \times g(x,y)$$
$$v(x,y) = f(x,y) \div g(x,y)$$

- These are elementwise operations performed between corresponding pixel pairs in f and g for x = 0,1,2,…, M-1 and y = 0,1,2,…,N-1. And s, d, p, and v are images of size M*N also.

55

Use cases of Arithmetic Operations

- Image addition (averaging) for noise reduction: The true image content remains consistent across images and is preserved, while noise, being random, averages out.
- Suppose that g(x,y) is a corrupted image formed by the addition of noise, η(x,y), to a noiseless image f(x,y);

$$g(x,y) = f(x,y) + \eta(x,y)$$

where the assumption is that at every pair of coordinates (x,y) the noise is uncorrelated and has zero average value.

- If an image $\bar{g}(x,y)$ is formed by averaging K different noisy images

$$\bar{g}(x,y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x,y)$$

56

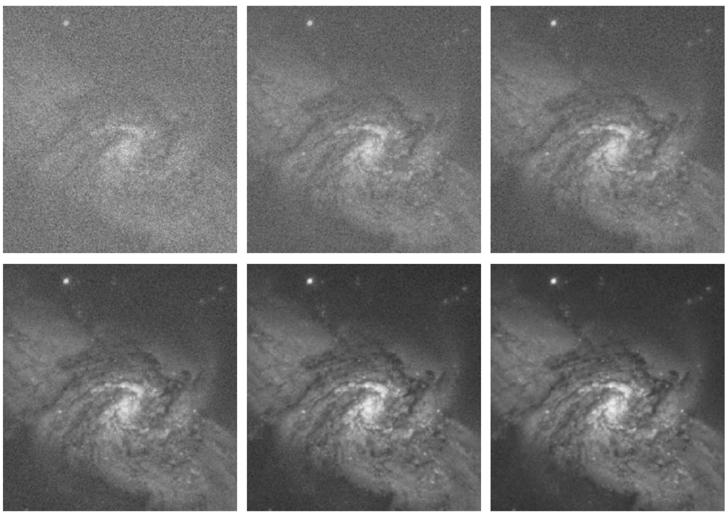
Use cases of Arithmetic Operations

- Then it follows, $E\{\bar{g}(x,y)\} = f(x,y)$

  and, $\sigma^2_{\bar{g}(x,y)} = \frac{1}{K}\sigma^2_{\eta(x,y)}$
- where E{ḡ(x,y)} is the expected value of ḡ(x,y), and $\sigma_{\bar{g}}^2$(x,y) and σ_{η}^2(x,y) are the variances of ḡ(x,y)} and η(x,y) respectively, all at coordinates (x,y).
- As K increases, the standard deviation of the pixel values at each location (x,y) decreases.
- Because E{ḡ(x,y)} = f(x,y) this means that g(x,y) approaches the noiseless image f (x,y) as the number of noisy images used in the averaging process increases.

57

Use cases of Arithmetic Operations



(a) Image of Galaxy Pair NGC 3314 corrupted by additive Gaussian noise.
(b) (b)-(f) Result of averaging 5, 10, 20, 50, and 1,00 noisy images, respectively. All images are of size 566*598 pixels, and all were scaled so that their intensities would span the full [0, 255] intensity scale.

58

Use cases of Arithmetic Operations

- Comparing images using subtraction.
- Visually indistinguishable images can be compared using subtraction operation.

(a) Infrared image of the Washington, D.C. area.
(b) Image resulting from 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.

59

Use cases of Arithmetic Operations

- Using image multiplication and division for shading correction and for masking.
- Let imaging sensor produces images f(x,y) and a shading function, h(x,y);

  that is,

  $$g(x,y) = f(x,y)\, h(x,y).$$

  Where, h(x,y) is known or can be estimated.
- We can get original image by

  $$f(x,y) = g(x,y) / h(x,y).$$

60

Use cases of Arithmetic Operations

Shading correction. (a) Shaded test pattern. (b) Estimated shading pattern. (c) Product of (a) by the reciprocal of (b). (See Section 3.5 for a discussion of how (b) was estimated.)

61

Use cases of Arithmetic Operations

- For masking, the process consists of multiplying a given image by a mask image that has 1's in the ROI and 0's elsewhere.
- There can be more than one ROI in the mask image, and the shape of the ROI can be arbitrary.

(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).

62

Enhancement using Set Operations

- To apply set concepts in image processing, let sets represent objects (regions) in a binary image, and the elements of the sets are the (x,y) coordinates of those objects.
- For example, if we want to know whether two objects, A and B, of a binary image overlap, all we have to do is compute $A \cap B$.
  - If the result is not the empty set, some of the elements of the two objects overlap.

Description	Expressions
Operations between the sample space and null sets	$\Omega^c = \varnothing$; $\varnothing^c = \Omega$; $\Omega \cup \varnothing = \Omega$; $\Omega \cap \varnothing = \varnothing$
Union and intersection with the null and sample space sets	$A \cup \varnothing = A$; $A \cap \varnothing = \varnothing$; $A \cup \Omega = \Omega$; $A \cap \Omega = A$
Union and intersection of a set with itself	$A \cup A = A$; $A \cap A = A$
Union and intersection of a set with its complement	$A \cup A^c = \Omega$; $A \cap A^c = \varnothing$
Commutative laws	$A \cup B = B \cup A$ $A \cap B = B \cap A$
Associative laws	$(A \cup B) \cup C = A \cup (B \cup C)$ $(A \cap B) \cap C = A \cap (B \cap C)$
Distributive laws	$(A \cup B) \cap C = (A \cap C) \cup (B \cap C)$ $(A \cap B) \cup C = (A \cup C) \cap (B \cup C)$
DeMorgan's laws	$(A \cup B)^c = A^c \cap B^c$ $(A \cap B)^c = A^c \cup B^c$

63

Set Operations in Grayscale Images.

- Let the elements of a grayscale image be represented by a set A whose elements are triplets of the form(x,y,z), where x and y are spatial coordinates, and z denotes intensity values.
- We define the complement of A as the set

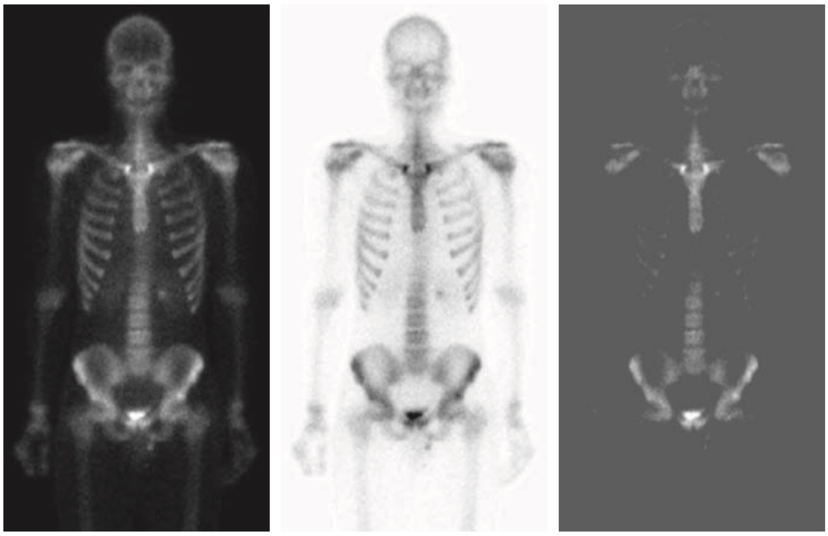
  $A^c = \{(x,y,K-z) | (x,y,z) \in A\}$ where K is maximum intensity value
- The negative is the set complement, and for 8-bit image, K= 255

  $A^c = \{(x,y,255-z) | (x,y,z) \in A\}$
- The union of two grayscale sets A and B with the same number of elements is defined as the set

$$A \cup B = \left\{ \max_z (a, b) \;\middle|\; a \in A, b \in B \right\}$$

64

Set Operations in Grayscale Images.



Set operations involving grayscale images.
(a) Original image.
(b) Image negative obtained using grayscale set complementation.
(c) The union of image (a) and a constant image.

65

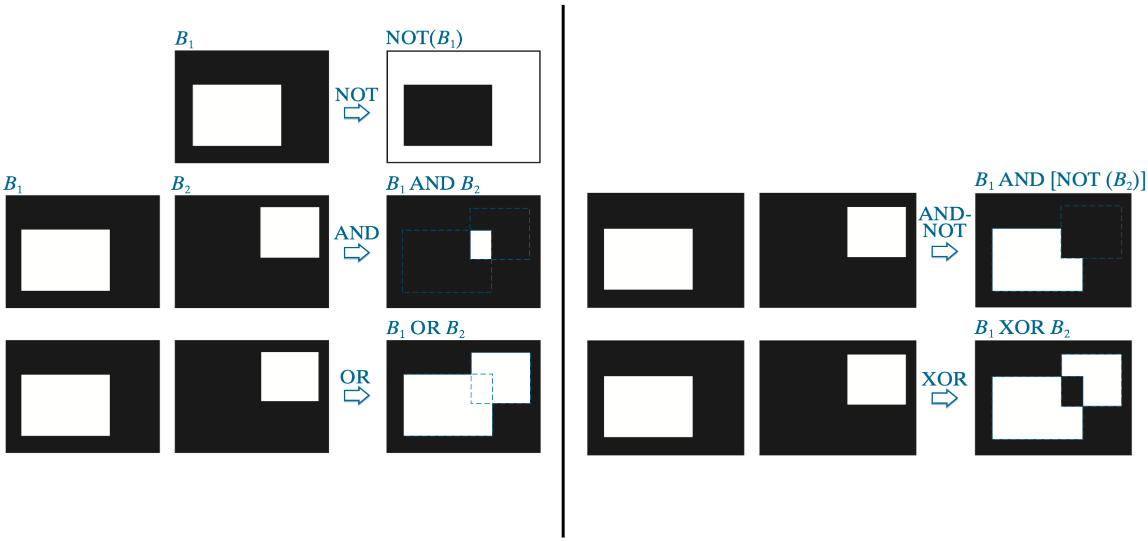
Enhancement using Logical Operations

- We work with set and logical operators on binary images using one of two basic approaches:
  - we can use the coordinates of individual regions of foreground pixels in a single image as sets, or
  - we can work with one or more images of the same size and perform logical operations between corresponding pixels in those arrays.

a	b	a AND b	a OR b	NOT(a)
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

66

Enhancement using Logical Operations



67

Spatial Filters

- Spatial filtering modifies an image by replacing the value of each pixel by a function of the values of the pixel and its neighbors.
- If the operation performed on the image pixels is linear, then the filter is called a linear spatial filter. Otherwise, the filter is a nonlinear spatial filter.
- Examples of linear filters: Averaging filter / Mean filter, Gaussian filter, Laplacian filter, Sobel filter (for edge detection)
- Examples of non-linear filters: Median filter, Max/min filters

68

Mechanics of Linear Spatial Filtering

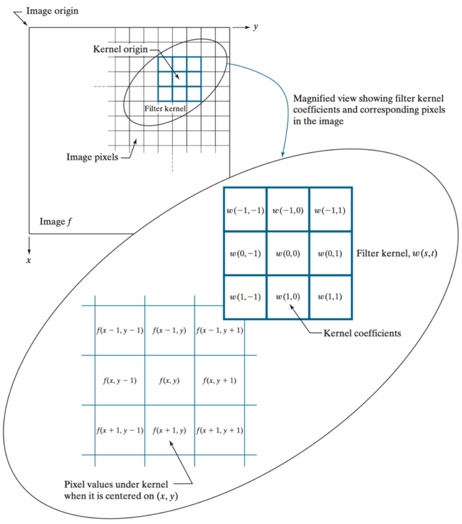
- A linear spatial filter performs a sum-of-products operation between an image ‘f’ and a filter kernel, ‘w’.
- The kernel is an array whose size defines the neighborhood of operation, and whose coefficients determine the nature of the filter.
- For a 3*3 kernel at any point (x,y) in the image, the response, g(x,y), of the filter is the sum of products of the kernel coefficients and the image pixels encompassed by the kernel:

$$g(x,y) = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + \dots + w(0,0)f(x,y) + \dots + w(1,1)f(x+1,y+1)$$

69

Mechanics of Linear Spatial Filtering

- As coordinates x and y are varied, the center of the kernel moves from pixel to pixel, generating the filtered image, g, in the process.



70

Spatial Correlation and Convolution

- Spatial correlation consists of moving the center of a kernel over an image, and computing the sum of products at each location.

a	b	c
d	e	f
g	h	i

Input Image Pixels

☆

r	s	t
u	v	w
x	y	z

Correlation Filter

$$e_{processed} = e*v + a*r + b*s + c*t + d*u + f*w + g*x + h*y + i*z$$

71

Spatial Correlation and Convolution

- The mechanics of spatial convolution are the same, except that the correlation kernel is rotated by 180°.
- We need to flip correlation filter horizontally and vertically to get convolution filter

r	s	t
u	v	w
x	y	z

Correlation Filter

H →

x	y	z
u	v	w
r	s	t

V →

z	y	x
w	v	u
t	s	r

Convolution filter

72

Spatial Correlation and Convolution

a	b	c
d	e	f
g	h	i

Input Image Pixels

★

z	y	x
w	v	u
t	s	r

Convolution Filter

$$e_{processed} = e*v + a*z + b*y + c*x + d*w + f*u + g*t + h*s + i*r$$

- When the values of a kernel are symmetric about its center, correlation and convolution yield the same result.

73

Spatial Correlation and Convolution

- In equation form, the correlation of a kernel w of size m*n with an image f(x,y), denoted as (w☆f)(x,y), is given by:

$$(w \star f)(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

- the convolution of a kernel w of size m*n with an image f (x,y), denoted by (w★f)(x,y) is defined as:

$$(w \bigstar f)(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t)$$

a and b are padding values, a = (m-1)/2 , b = (n-1)/2 and we assume that m and n are odd integers.

74

Spatial Correlation and Convolution

- Some fundamental properties of convolution and correlation. A dash means that the property does not hold.

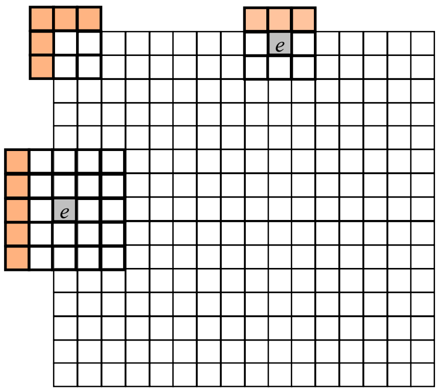
Property	Convolution	Correlation
Commutative	$f \star g = g \star f$	—
Associative	$f \star (g \star h) = (f \star g) \star h$	—
Distributive	$f \star (g + h) = (f \star g) + (f \star h)$	$f \mathbin{\text{☆}} (g + h) = (f \mathbin{\text{☆}} g) + (f \mathbin{\text{☆}} h)$

75

Handling Pixels Closer to Boundaries

There are a few approaches to deal with missing edge pixels:

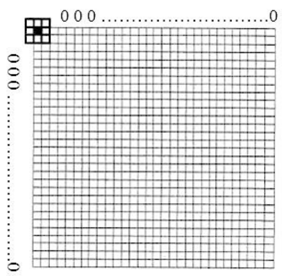
- Omit missing pixels
  - Ignores border pixels that don't have a full neighborhood around them
- Pad the image
  - Add extra layer with either all white or all black pixels
  - Replicate border pixels
- Truncate the image
  - Only processes the **interior** part of the image where the filter fits completely. Removes outer pixels.
- Circular Padding
  - Allow pixels wrap around the image
- Mirror Padding
  - reflects pixel values at the border of the image



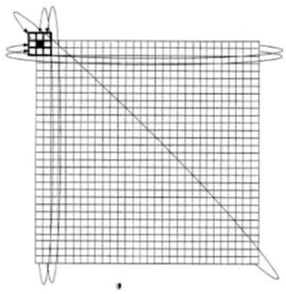
76

Handling Pixels Closer to Boundaries

- Padding
  - Pad with zeros: pad around the input image so that first pixel of image becomes center pixel of the filter
  - Wrap around: get pixel values from around the corner of the image to extend the boundaries.



or



77

Handling Pixels Closer to Boundaries

0	0	0	0	0	0	0
0	1	2	3	4	5	0
0	6	7	8	9	10	0
0	11	12	13	14	15	0
0	16	17	18	19	20	0
0	21	22	23	24	25	0
0	0	0	0	0	0	0

Zero Padding

1	1	2	3	4	5	5
1	1	2	3	4	5	5
6	6	7	8	9	10	10
11	11	12	13	14	15	15
16	16	17	18	19	20	20
21	21	22	23	24	25	25
21	21	22	23	24	25	25

Replication

25	21	22	23	24	25	21
5	1	2	3	4	5	1
10	6	7	8	9	10	6
15	11	12	13	14	15	11
20	16	17	18	19	20	16
25	21	22	23	24	25	21
5	1	2	3	4	5	1

Wrap Around

78

Handling Pixels Closer to Boundaries

7	6	7	8	9	10	9
2	1	2	3	4	5	4
7	6	7	8	9	10	9
12	11	12	13	14	15	14
17	16	17	18	19	20	19
22	21	22	23	24	25	24
17	16	17	18	19	20	19

Mirror Padding

Discussions and Classwork

- Perfrom correlation and convulotion for given image pixels and kernel values using zero padding, wrap around and replication techniques.

6	6	7	5	8
12	12	9	14	15
6	4	4	5	2
1	4	3	11	13
3	5	6	7	14

Input Image

1	3	1
2	8	2
3	1	3

Kernel

Lecture – 06

Harimohan Khatri

81

In this lecture…

i. Smoothing and Sharpening Spatial Filters
   i. Averaging
   ii. Median Filtering
   iii. Spatial Low Pass
   iv. High Pass Filtering
   v. Modification by replication and interpolation

82

Smoothing (Lowpass) Spatial Filter

- Smoothing spatial filters are used to reduce sharp transitions in intensity, that means noise reduction.
- Another application is for smoothing the false contours that result from using an insufficient number of intensity levels in an image.
- Mainly there are two types of smoothing filters:
  - Box Filter
  - Gaussian Filter

83

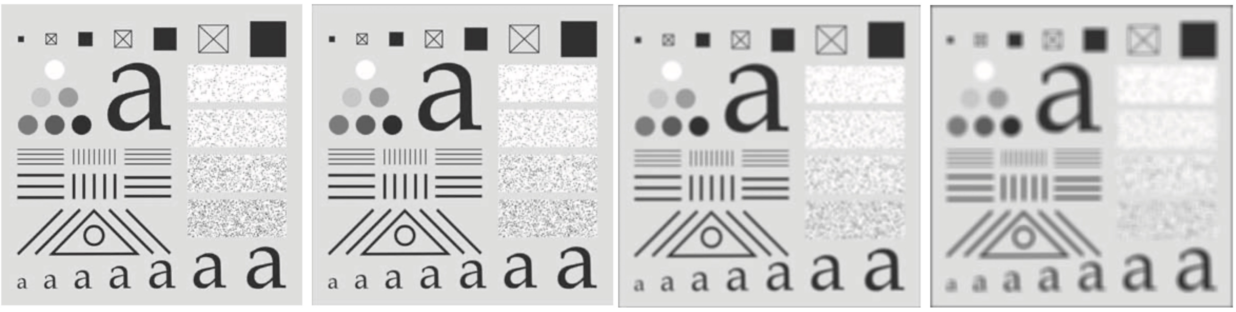
Smoothing Box Kernels(Averaging Filter)

- An m×n box filter is an m×n array of 1's, with a normalizing constant in front, whose value is 1 divided by the sum of the values of the coefficients (i.e., 1/mn when all the coefficients are 1's).
- The normalization in this kernel has two purposes:
  - First, the average value of an area of constant intensity would equal that intensity in the filtered image, as it should.
  - Second, the sum of the pixels in the original and filtered images will be the same

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

84

Smoothing Box Kernels



(a) Test pattern of size 1024×1024 pixels.
(b)-(d) Results of lowpass filtering with box kernels of sizes 3×3, 11×11, and 21×21, respectively

85

Limitations of Box Kernels

- Box filters favor blurring along perpendicular directions. For images with high level of detail or with strong geometrical components, the directionality of box filters often produces undesirable results.
- Also, a defocused lens is often modeled as a lowpass filter, but box filters are poor approximations to the blurring characteristics of lenses.
- In such cases, choice of kernel must be a circularly symmetric also called isotropic, (their response is independent of orientation)

86

Lowpass Gaussian Kernels

- Gaussian kernels of the form

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

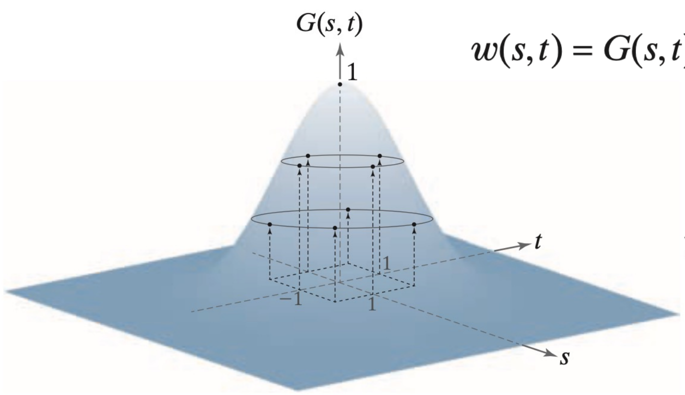
where: x, y are the coordinates
PI = 3.14
σ is the Standard Deviation

are the only circularly symmetric kernels that are also separable

Note: A separable kernel is a 2D filter kernel that can be broken down into the product of two 1D kernels — one for rows and one for columns. This reduces computational complexity significantly.

87

Lowpass Gaussian Filter



$$w(s,t) = G(s,t) = Ke^{-\frac{s^2+t^2}{2\sigma^2}}$$

$$\frac{1}{4.8976} \times \begin{bmatrix} 0.3679 & 0.6065 & 0.3679 \\ 0.6065 & 1.0000 & 0.6065 \\ 0.3679 & 0.6065 & 0.3679 \end{bmatrix}$$

Kernel value with K =1 and σ =1

88

Gaussian Lowpass Filter

- For simplicity, we can approximate the Gaussian filter in form of weighted average filter as:

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Q. Apply 3*3 Gaussian filter on the given image f(x,y). Use zero padding and circular padding.

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

89

Gaussian Low Pass Filter

(a)Input Image
(b) Result of filtering using a Gaussian kernels of size 43*43 and σ = 7
(c)Result of using a kernel of 85*85 of same σ
(d) Difference image.

90

Gaussian Filter in Different Padding

Result of filtering the test pattern
(a) using zero padding,
(b) mirror padding, and
(c) replicate padding.

91

Order Statistic (Nonlinear) Filters

- spatial filters whose response is based on ordering(ranking) the pixels contained in the region encompassed by the filter.
- Smoothing is achieved by replacing the value of the center pixel with the value determined by the ranking result.
- The best-known filter in this category are
  - median filter,
  - Min, max filter

92

Median Filter

- It replaces the value of the center pixel by the median of the intensity values in the neighborhood of that pixel.
- Median filters are particularly effective in the presence of impulse noise (salt-and-pepper noise)

Q. Apply 3*3 Median filter on the given image f(x,y). Use zero padding and pixel replication for padding.

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

93

Min/Max Filter

- Max-Filter replaces the value of the center pixel by the maximum of the intensity values in the neighborhood of that pixel. Where Min-Filter replaces the value by the minimum of the intensity values.

Q. Apply 3*3 Min-Filter and Max-Filter on the given image f(x,y). Use Mirror padding.

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

94

Sharpening (High Pass) Filters

- Sharpening spatial filters seek to highlight fine detail
  - Remove blurring from images
  - Highlight edges
- Sharpening filters are based on spatial differentiation
- A derivative operator is proportional to the magnitude of the intensity discontinuity at the point at which the operator is applied.
- Thus, image differentiation enhances edges and other discontinuities and de-emphasizes areas with slowly varying intensities.

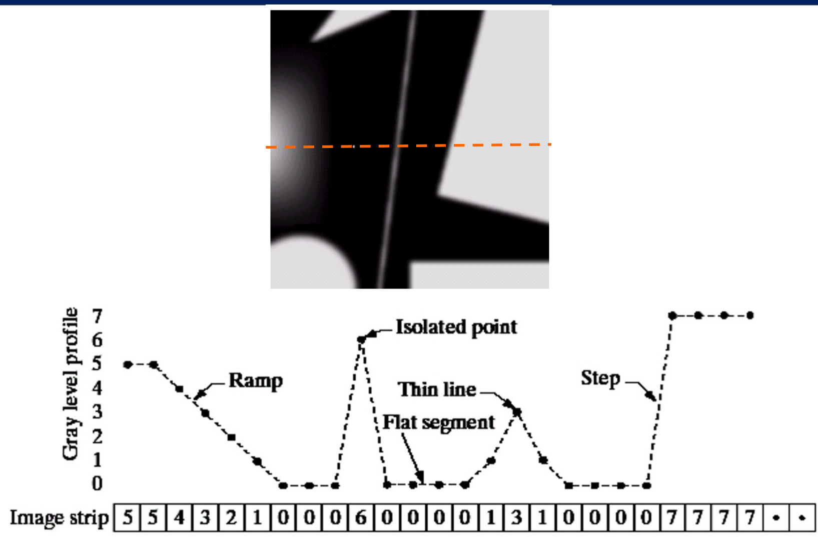
95

Key Points on Derivatives of Digital Functions

- First derivative of digital function must include:
  - Must be zero in areas of constant intensity.
  - Must be nonzero at the onset of an intensity step.
  - Must be nonzero along intensity ramps(gradual change).
- Similarly, Second derivative of digital function must include:
  - Must be zero in areas of constant intensity.
  - Must be nonzero at the onset and end of an intensity step or ramp.
  - Must be zero along intensity ramps.

96

Key Points on Derivatives of Digital Functions



97

Key Points on Derivatives of Digital Functions

1st Derivative:

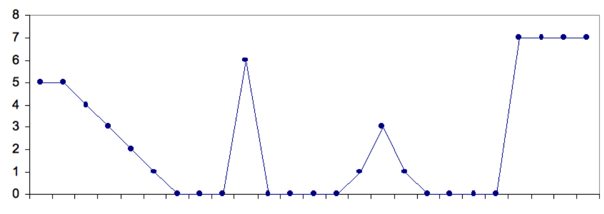
- The formula for the 1st derivative of a function is as follows:

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

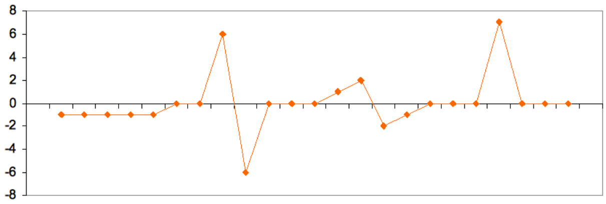
- It's just the difference between subsequent values and measures the rate of change of the function.

98

Key Points on Derivatives of Digital Functions



5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7
	0	-1	-1	-1	-1	0	0	6	-6	0	0	0	1	2	-2	-1	0	0	0	7	0	0	0	



Key Points on Derivatives of Digital Functions

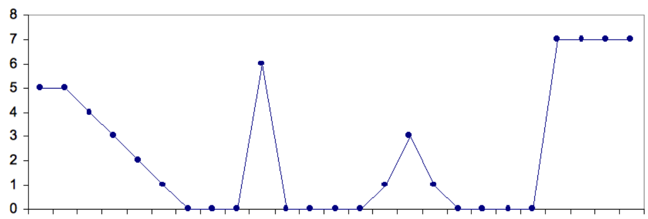
2nd Derivative:

- The formula for the 2nd derivative of a function is as follows:

$$\frac{\partial^2 f}{\partial^2 x} = f(x+1) + f(x-1) - 2f(x)$$

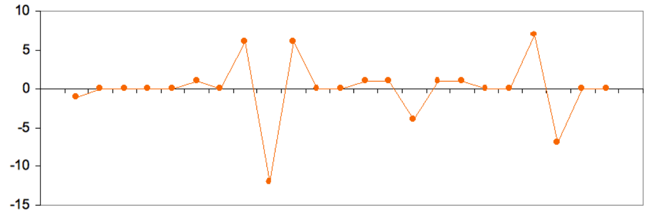
- Simply takes into account the values both before and after the current value.

Key Points on Derivatives of Digital Functions



5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7

	-1	0	0	0	0	1	0	6	-12	6	0	0	1	1	-4	1	1	0	0	7	-7	0	0	



101

Isotropic kernels: response is independent of direction of intensity discontinuities in the image to which the filter is applied.

2^{nd} Derivative for Image Sharpning – The Laplacian

- The simplest isotropic derivative operator is the Laplacian,
- For a function (image) f(x,y), 2^{nd} derivative is defined as:

$$\nabla^2 f = \frac{\partial^2 f}{\partial^2 x} + \frac{\partial^2 f}{\partial^2 y}$$

where the partial 1st order derivative in the x direction is defined as follows:

$$\frac{\partial^2 f}{\partial^2 x} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

and in the y direction as follows:

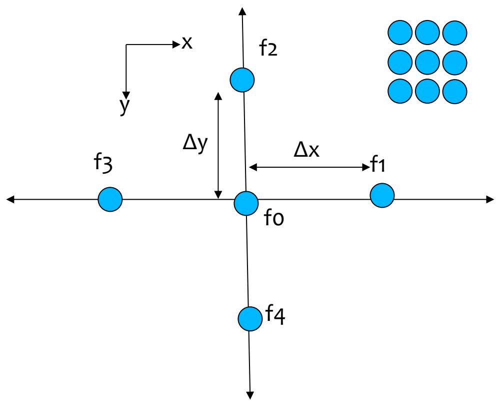
$$\frac{\partial^2 f}{\partial^2 y} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

102

First Derivatives

- $df/dx = f(x+1) - f(x)$
- $df/dy = f(y+1) - f(y)$
    - $df/dx = (f1-f0)/\Delta x$
    - $df/dy = (f2-f0)/\Delta y$
    - $df/dx = (f0-f3)/\Delta x$
    - $df/dy = (f0-f4)/\Delta y$

For $\Delta x = \Delta y = 1$
$df/dx = f1-f0$
$df/dy = f2-f0$
$df/dx = f0-f3$
$df/dy = f0-f4$



Second Derivatives

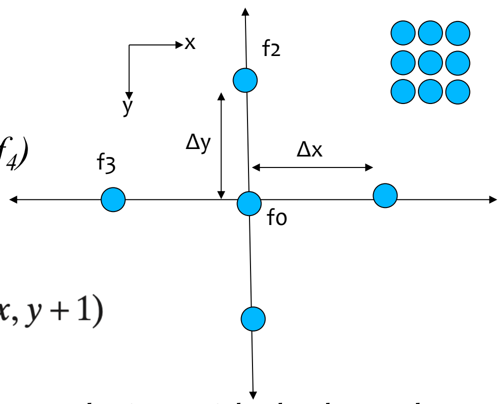
- $d^2f/dx^2 = f(x-1) - 2f(x) + f(x+1)$
- $d^2f/dy^2 = f(y-1) - 2f(y) + f(y+1)$

For $\Delta x = \Delta y = 1$
$d^2f/dx^2 = [(f1-f0)/\Delta x] - [(f0-f3)/\Delta x] = (f1-f0)-(f0-f3) = f1 - 2f0 + f3$
$d^2f/dy^2 = [(f2-f0)/\Delta y] - [(f0-f4)/\Delta y] = (f2-f0)-(f0-f4) = f2 - 2f0 + f4$

103

The Laplacian

$$\nabla^2 f(x,y) = d^2f/dx^2 + d^2f/dy^2$$
$$= (f_1 - 2f_0 + f_3) + (f_2 - 2f_0 + f_4)$$
$$= f_1 + f_2 + f_3 + f_4 - 4f_0$$



Laplacian of two variables:

$$\nabla^2 f(x,y) = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

Equation can be implemented using convolution with the kernel

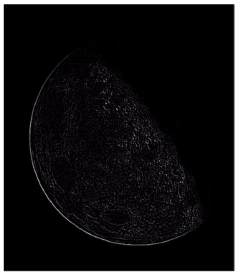
0	1	0
1	−4	1
0	1	0

104

2^{nd} Derivative for Image Sharpning – The Laplacian

- Applying the Laplacian to an image we get a new image that highlights edges and other discontinuities


Original Image


Laplacian Filtered Image


Laplacian Filtered Image Scaled for Display

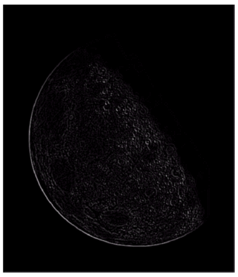
105

2^{nd} Derivative for Image Sharpning – The Laplacian

- The result of a Laplacian filtering is not an enhanced image
- We have to subtract the Laplacian result from the original image to generate our final sharpened enhanced image


Original Image

-


Laplacian Filtered Image

=


Sharpened Image

106

2nd Derivative for Image Sharpning – The Laplacian

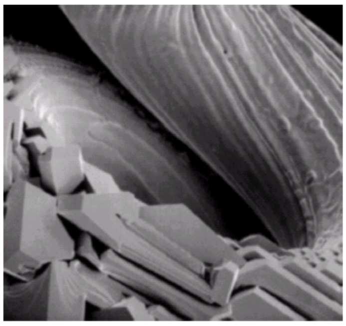
- The entire enhancement can be combined into a single filtering operation

$$
\begin{aligned}
g(x, y) &= f(x, y) - \nabla^2 f \\
&= f(x, y) - [f(x+1, y) + f(x-1, y) \\
&\qquad + f(x, y+1) + f(x, y-1) - 4f(x, y)] \\
&= 5f(x, y) - f(x+1, y) - f(x-1, y) \\
&\qquad - f(x, y+1) - f(x, y-1)
\end{aligned}
$$

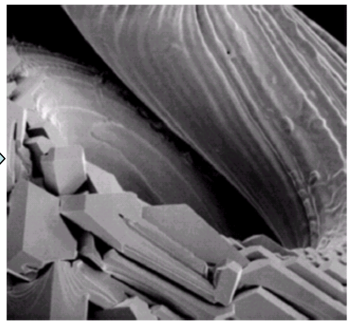
107

2nd Derivative for Image Sharpning – The Laplacian

- This gives us a new filter which does the whole job for us in one step



0	-1	0
-1	5	-1
0	-1	0



108

2nd Derivative for Image Sharpning – The Laplacian

Some more Laplacian kernels

- Generalized equation of Laplacian kernel for image sharpening is:

$$g(x,y) = f(x,y) + c\left[\nabla^2 f(x,y)\right]$$

where f(x,y) and g(x,y) are the input and sharpened images, respectively.

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

c = −1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

c = 1

109

Unsharp Masking (Highboost Filtering)

- Subtracting an unsharp (smoothed) version of an image from the original image is called unsharp masking, consists of the following steps:
  - Blur the original image.
  - Subtract the blurred image from the original (the resulting difference is called the mask.)
  - Add the mask to the original.
- Letting $\bar{f}(x,y)$ denote the blurred image, the mask in equation form is given by:

$$g_{\text{mask}}(x,y) = f(x,y) - \bar{f}(x,y)$$

- Then we add a weighted portion of the mask back to the original image:

$$g(x,y) = f(x,y) + k\,g_{\text{mask}}(x,y)$$

The weightage value k defines sharpness in output image; greater the value more the sharpness.

110

Unsharp Masking (Highboost Filtering)

(a) Original image of size 600×259 pixels. (b) Image blurred using a 31×31 Gaussian lowpass filter with σ = 5. (c) Mask. (d) Result of unsharp masking with k= 1. (e) Result of highboost filtering with k= 4.5

111

First Derivative for Sharpening – The Gradient

- For a function f(x, y) the gradient of f at coordinates (x, y) is given as the column vector:

$$\nabla \mathrm{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

- The magnitude of this vector is given by:

$$\nabla f = mag(\nabla \mathrm{f}) = \left[G_x^2 + G_y^2\right]^{1/2} = \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}$$

- This can be simplified as: $\nabla f \approx |G_x| + |G_y|$

112

First Derivative for Sharpening – The Gradient

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- The magnitude of the gradient at z_5 can be approximated in number of ways:
- The simplest is to use the difference (z5-z8) in x-direction and (z5-z6) in y-direction

$$\nabla f = [(z_5 - z_8)^2 + (z_5 - z_6)^2]^{1/2}$$

- Or approximated as:

$$\nabla f \approx |z_5 - z_8| + |z_5 - z_6|$$

113

First Derivative for Sharpening – Cross Gradient

- The cross difference may also be used to approximate the magnitude of the gradient

$$\nabla f = [(z_5 - z_9)^2 + (z_6 - z_8)^2]^{1/2}$$

$$\nabla f \approx |z_5 - z_9| + |z_6 - z_8|$$

- This can be implemented by taking the absolute value of the response of the following two masks(the ***Roberts cross-gradient operators***) and summing the results.

1	0
0	-1

0	1
-1	0

114

First Derivative for Sharpening – Prewitt Operator

- Extension to a 3*3 mask yields the following

$$\nabla f \approx |(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)| + |(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)|$$

- The difference between the first and third rows approximates the derivative in the x direction.
- The difference between the first and third columns approximates the derivative in the y direction.
- The ***Prewitt operator*** masks may be used to implement the above approximation

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

115

First Derivative for Sharpening – Sobel Operator

- The ***Sobel operator*** masks may also be used to implement the derivative approximation

$$\nabla f \approx |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

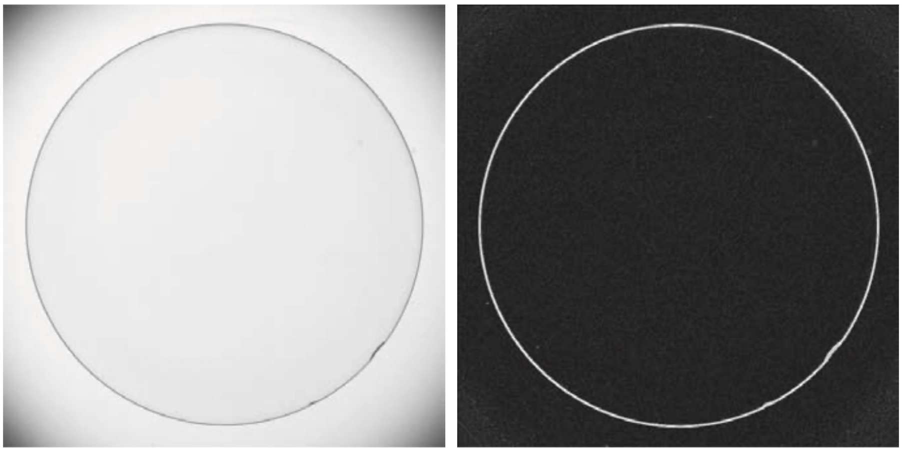
- The Sobel operators are widely used for edge detection.

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

116

First Derivative for Sharpening – Sobel Operator



(a) Image of a contact lens (note defects on the boundary at 4 and 5 o'clock).
(b) Sobel gradient.

117

Combining Spatial Enhancement Method

- Successful image enhancement is typically not achieved using a single operation
- Rather we combine a range of techniques in order to achieve a final result
- This example will focus on enhancing the bone scan to the right
- Our objective is to enhance this image by sharpening it and by bringing out more of the skeletal detail.

118

Combining Spatial Enhancement Method

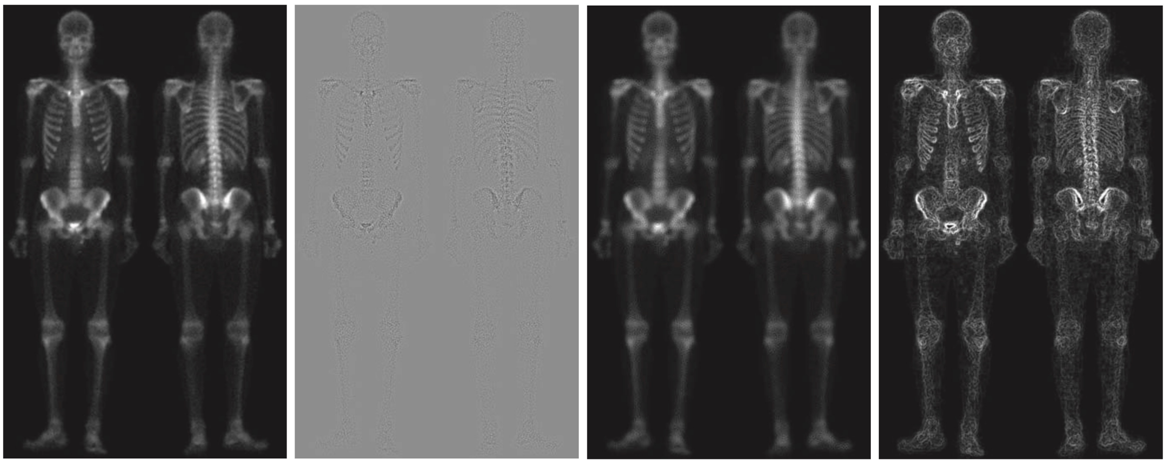
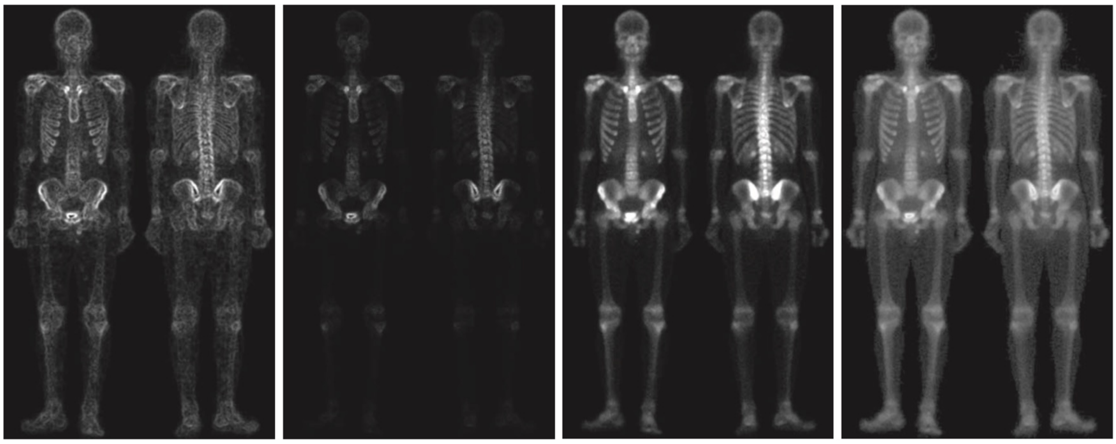


Image of whole body bone scan. → Laplacian of (a). → Sharpened image obtained by adding (a) and (b). →Sobel gradient of image →

119

Combining Spatial Enhancement Methods



→ Sobel image smoothed with a 5*5 box filter. → Mask image formed by the product of (b) and (e). →Sharpened image obtained by the adding images (a) and (f). →Final result obtained by applying a power-law transformation to (g). Compare images (g) and (h) with (a).

120

Magnification of an Image

- Magnification in image processing refers to enlarging an image or a region of interest while preserving visual quality.
- It involves two steps:
  - Creating new pixels and,
  - Assigning gray levels to those new levels
- Basic magnification techniques are:
  - Replication
  - Interpolation

121

Magnification by Replication

- Each pixel is replicated between existing pixels in an image to increase its resolution.

4	2
6	5

→

4	4	2	2
4	4	2	2
6	6	5	5
6	6	5	5

122

Magnification by Interpolation

Step – 01: Zero Interlacing

- Inserting zero-valued pixels between existing pixels in an image to increase its resolution.

4	2
6	5

→

4	0	2	0
0	0	0	0
6	0	5	0
0	0	0	0

123

Magnification by Interpolation

Step-02: Interpolation

- Each pixel is interpolated in row and column.

4	0	2	0
0	0	0	0
6	0	5	0
0	0	0	0

→

4	3	2	1
5	3.75	3.5	1.75
6	4.5	5	2.5
3	2.25	2.5	1.25

→

4	3	2	1
5	4	4	2
6	5	5	3
3	2	3	2

124

Discussions and Classwork

- Compare smooting and sharpning filters with their kernel values and working mechanism.
- How combination of spatial smooting and sharpning filters contribute to image enhancement. Discuss.

125