Digital Image Processing

Lecture # 6

Image Enhancement in Spatial Domain- II

ALI JAVED

Lecturer

SOFTWARE ENGINEERING DEPARTMENT

U.E.T TAXILA

Email:: ali.javed@uettaxila.edu.pk

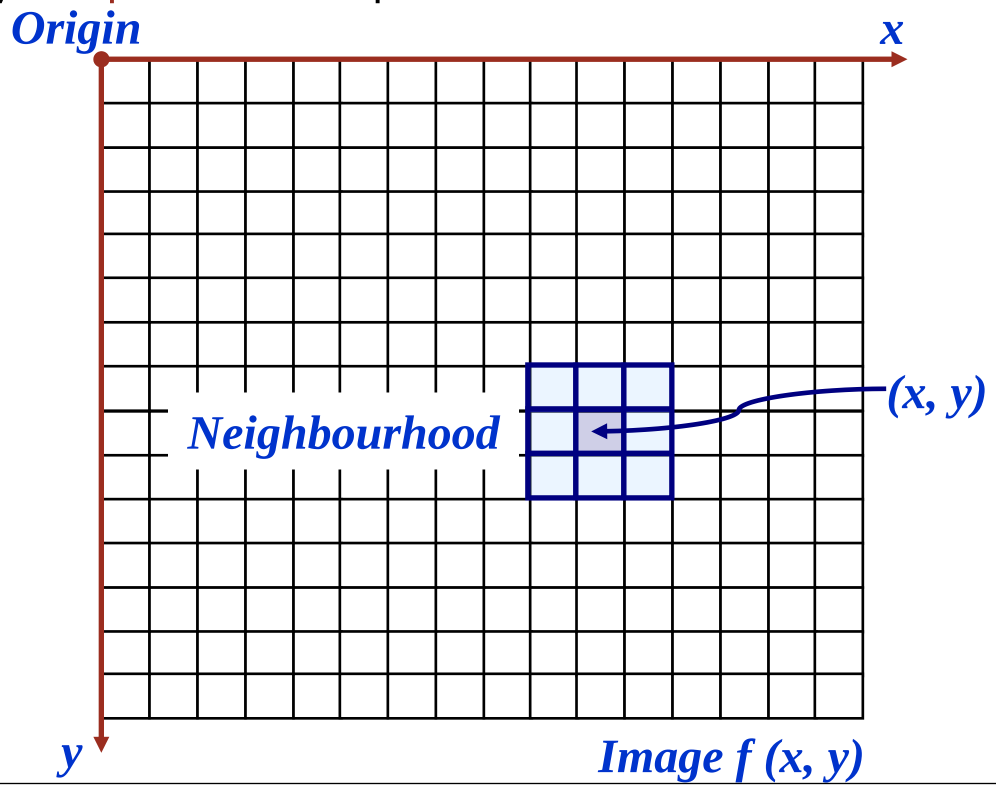
Office Room #:: 7

Presentation Outline

- Local/ Neighborhood Operations on images
- Local Enhancement through Spatial Filtering
- Smoothing Spatial Filtering
  - Linear Filters
  - Non-Linear Filters
- Linear Smoothing Spatial Filtering
  - Average Filters
  - Weighted Average Filters
- Non-Linear Smoothing Spatial Filtering
  - Median Filters
  - Minimum Filters
  - Maximum Filters
- Border Pixel Treatment
- Image Histogram
- Histogram Equalization

Neighborhood Operations on Images

- Neighbourhood operations simply operate on a larger neighbourhood of pixels than point operations
- Neighbourhoods are mostly selected a square regions around a central pixel
- Any size and any shape filter are possible

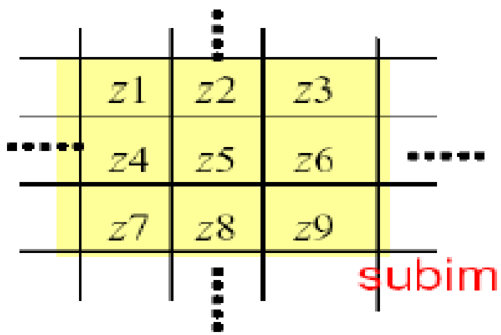


Local Operations through Spatial Filtering

- The output intensity value at (x,y) depends not only on the input intensity value at (x,y) but also on the specified number of neighboring intensity values around (x,y)
- Spatial masks (also called window, filter, kernel, template) are used and convolved over the entire image for local enhancement (spatial filtering)
- The size of the mask determines the number of neighboring pixels which influence the output value at (x,y)
- The values (coefficients) of the mask determine the nature and properties of enhancing technique

Basics of Spatial Filtering

- Given the 3×3 mask with coefficients: $w_1, w_2, \ldots, w_9$
- The mask covers the pixels with gray levels: $z_1, z_2, \ldots, z_9$



subimage

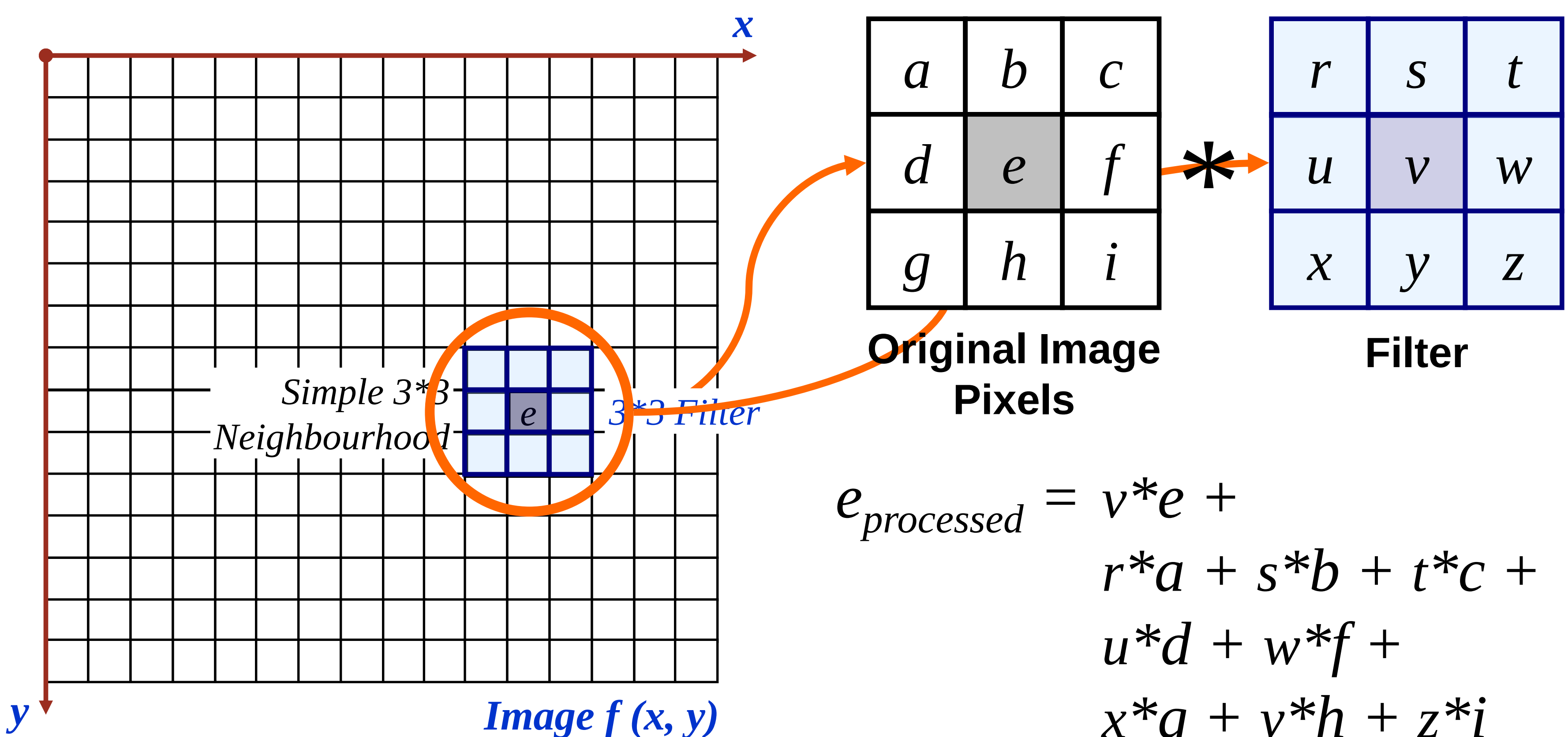
w1	w2	w3
w4	w5	w6
w7	w8	w9

Mask coefficients

$$z_5 \leftarrow z = w_1 z_1 + w_2 z_2 + \cdots + w_9 z_9 = \sum_{i=1}^{9} w_i z_i$$

- z gives the output intensity value for the processed image (to be stored in a new array) at the location of z_5 in the input image

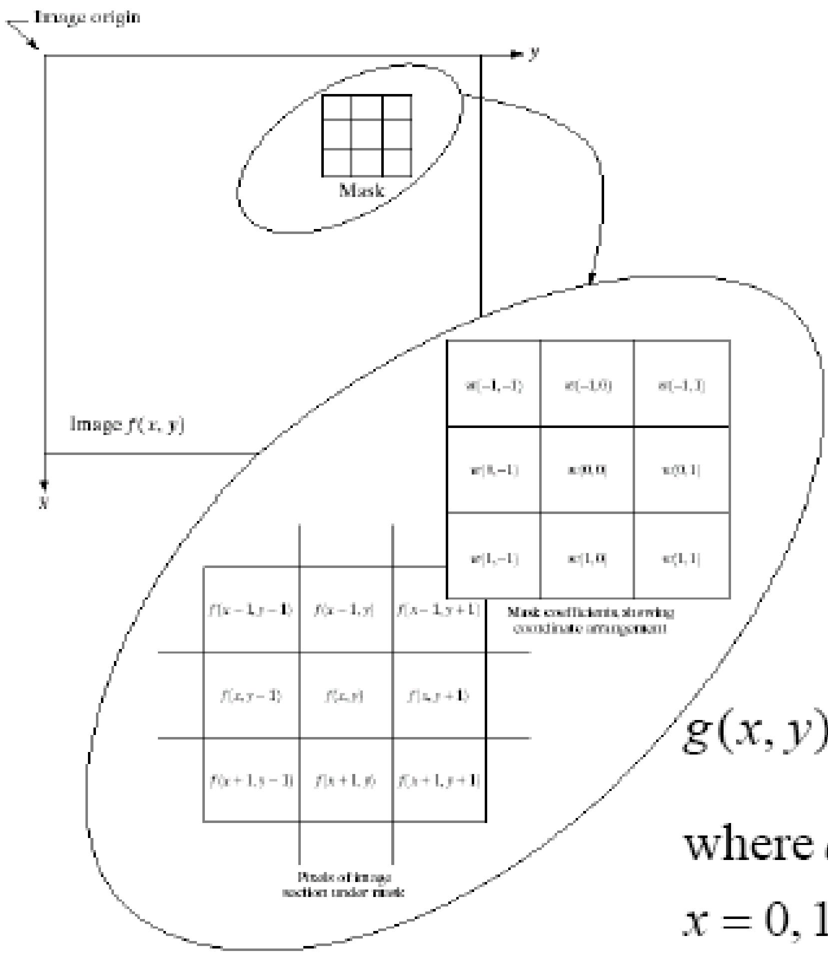
Local Operations through Spatial Filtering



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

- The above is repeated for every pixel in the original image to generate the smoothed image

Local Operations through Spatial Filtering



The mechanics of spatial filtering

For an image of size M x N and a mask of size m x n

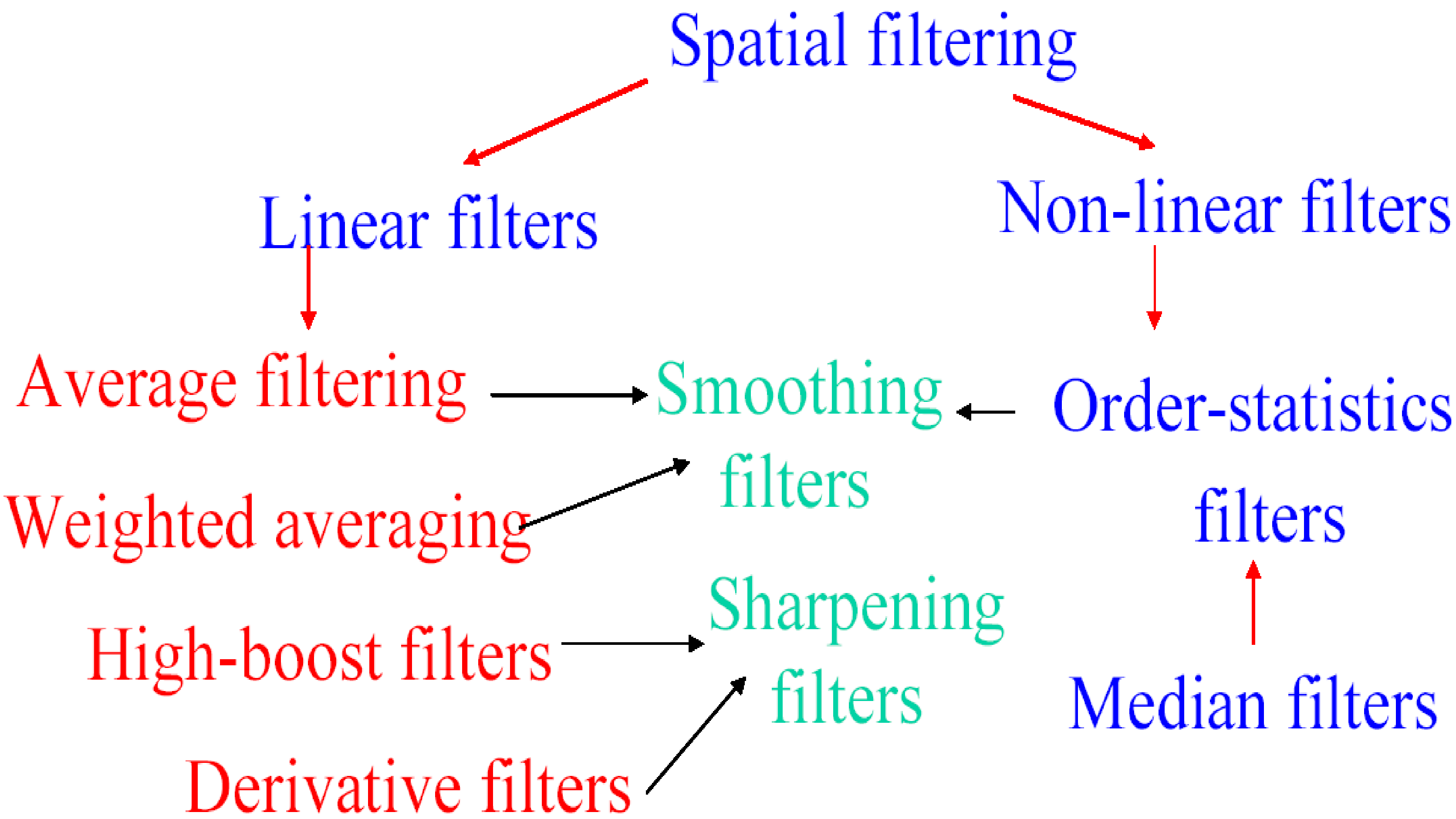
The resulting output gray level for any coordinates x and y is given by

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

where $a=(m-1)/2, \quad b=(n-1)/2$

$x=0,1,2,\ldots\ldots,M-1, \quad y=0,1,2,\ldots\ldots,N-1,$

Types of Spatial Filtering



Spatial Filtering for Smoothing

- For blurring/noise reduction;
- Blurring is usually used in preprocessing steps, e.g., to remove small details from an image prior to object extraction, or to bridge small gaps in lines or curves
- Equivalent to Low-pass spatial filtering in frequency domain because smaller (high frequency) details are removed based on neighborhood averaging (averaging filters)
- Implementation:
  - The simplest form of the spatial filter for averaging is a square mask (assume *m*×*m* mask) with the same coefficients 1/m2 to preserve the gray levels (averaging).
- Applications: Reduce noise; smooth false contours
- Side effect: Edge blurring

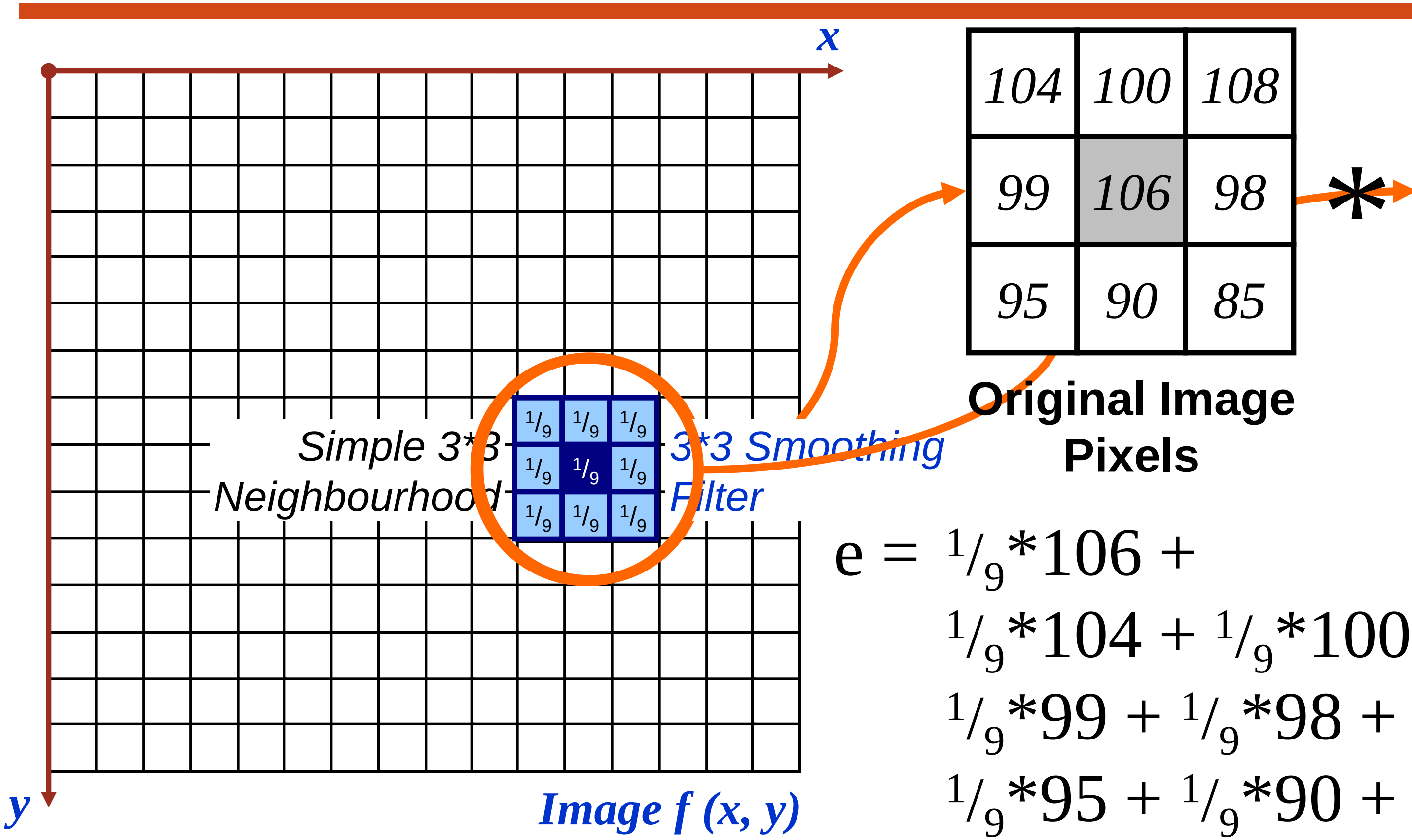
Smoothing Spatial Filters

- One of the simplest spatial filtering operations we can perform is a smoothing operation
  - Simply average all of the pixels in a neighbourhood around a central value
  - Especially useful in removing noise from images

$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$

Simple Averaging Filter

Smoothing Spatial Filters



$$
\begin{aligned}
e = {} & {}^1/_9*106 + \\
& {}^1/_9*104 + {}^1/_9*100 + {}^1/_9*108 + \\
& {}^1/_9*99 + {}^1/_9*98 + \\
& {}^1/_9*95 + {}^1/_9*90 + {}^1/_9*85 \\
= {} & 98.3333
\end{aligned}
$$

- The above is repeated for every pixel in the original image to generate the smoothed image

Spatial Filtering for Smoothing:: Example

Original image size: 500 x 500

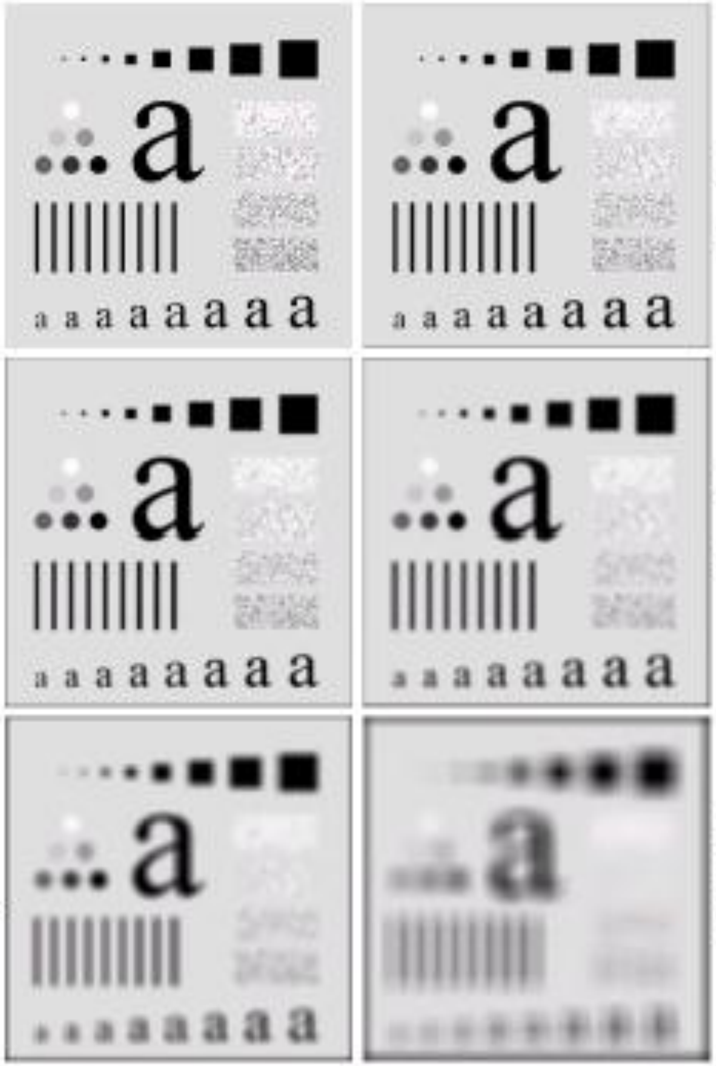
Smoothed by 3 x 3 box filter

Smoothed by 5 x 5 box filter

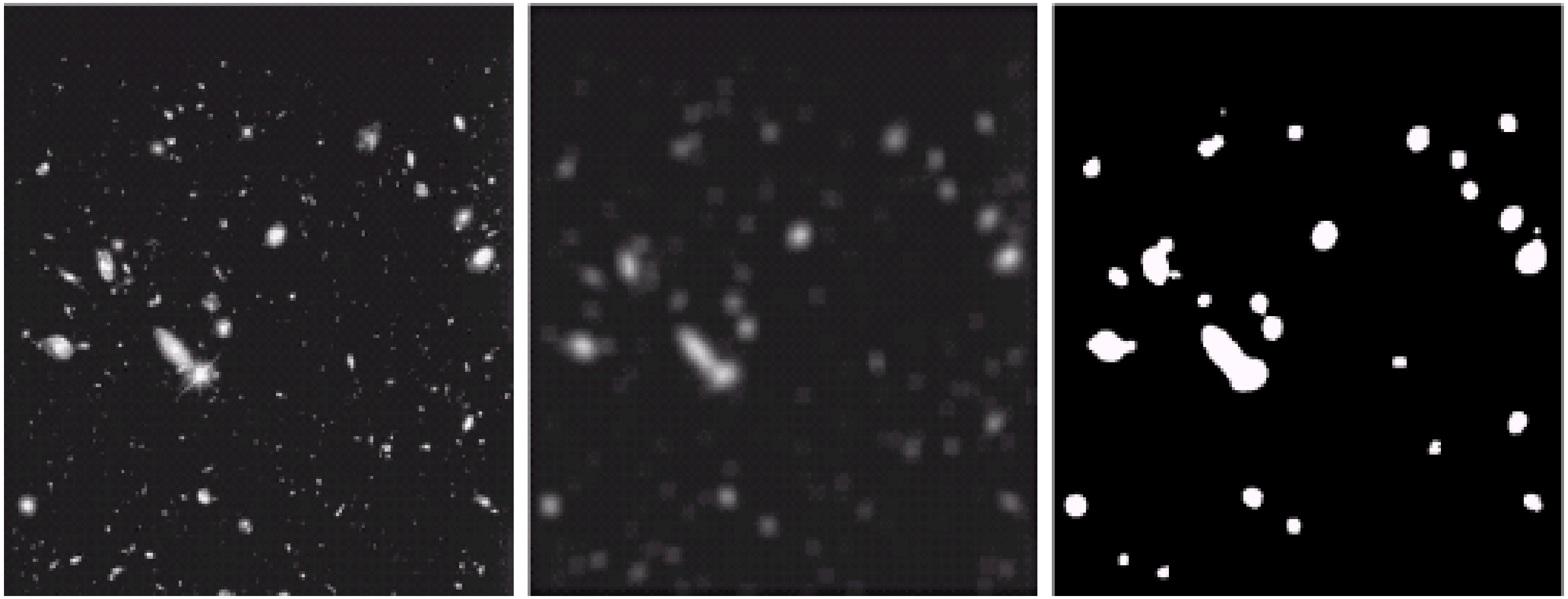
Smoothed by 9 x 9 box filter

Smoothed by 15 x 15 box filter

Smoothed by 35 x 35 box filter



Spatial Filtering for Smoothing:: Example



a b c

FIGURE 3.36 (a) Image from the Hubble Space Telescope. (b) Image processed by a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

Weighted Smoothing Spatial Filters

- More effective smoothing filters can be generated by allowing different pixels in the neighbourhood different weights in the averaging function
  - Pixels closer to the central pixel are more important
  - Often referred to as a *weighted averaging*

$^{1}/_{16}$	$^{2}/_{16}$	$^{1}/_{16}$
$^{2}/_{16}$	$^{4}/_{16}$	$^{2}/_{16}$
$^{1}/_{16}$	$^{2}/_{16}$	$^{1}/_{16}$

Weighted Averaging Filter

Order Statistics Filtering

- Nonlinear spatial filters

- Output is based on order of gray levels in the masked area (sub-image)

- Examples: Median filtering, Max & Min filtering

  - **Min: Set the pixel value to the minimum in the neighbourhood**

  - **Max: Set the pixel value to the maximum in the neighbourhood**

  - **Median: The median value of a set of numbers is the midpoint value in that set (e.g. from the set [1, 7, 15, 18, 24] 15 is the median). Sometimes the median works better than the average**

Order Statistics Filtering

Median Filtering

- Assigns the mid value of all the gray levels in the mask to the center of mask;

- Particularly effective when

  - *the noise pattern consists of strong, spiky components (salt-and-pepper)*

  - *edges are to be preserved*

  - *Force points with distinct gray levels to be more like their neighbors*

Order Statistics Filtering

10	20	20
20	15	20
20	25	100

↓

Output = ? **20**

Order Statistics Filtering

Minimum Filtering

- Assigns the minimum value of all the gray levels in the mask to the center of mask;

- Particularly effective when

    - ***Only Salt noise is present***

Maximum Filtering

- Assigns the maximum value of all the gray levels in the mask to the center of mask;

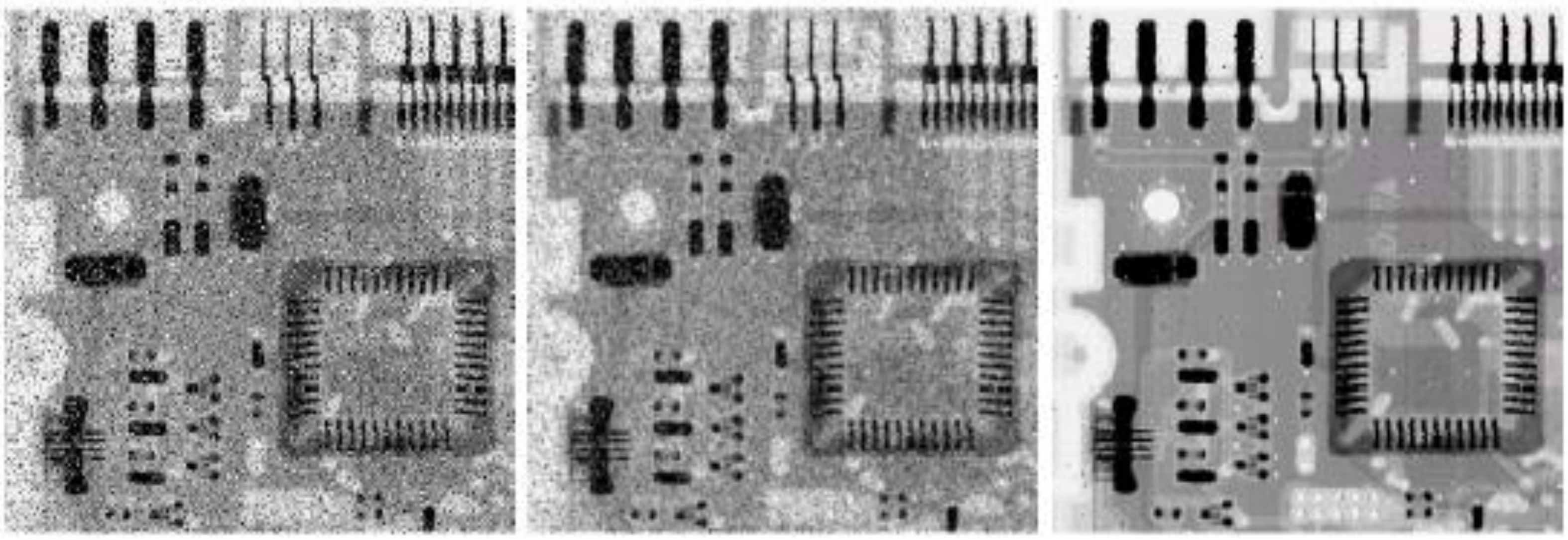
- Particularly effective when

    - ***Only Pepper noise is present***

- *Edges are to be preserved*

- *Force points with distinct gray levels to be more like their neighbors*

Order Statistics Filtering

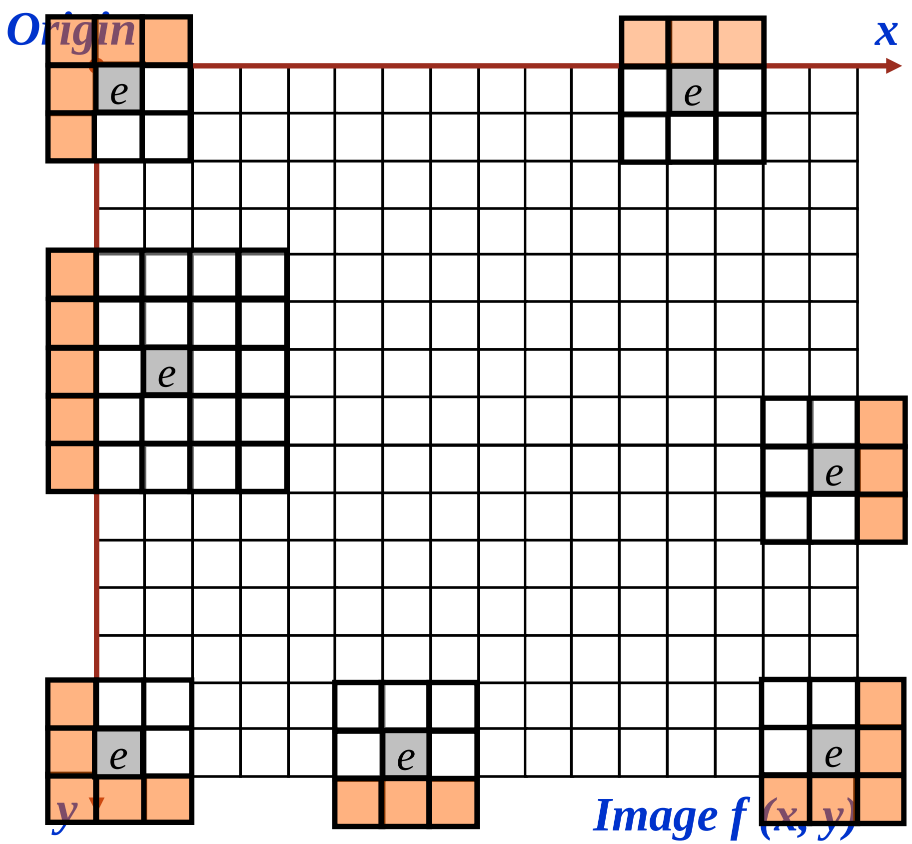


a b c

FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Strange Things Happen At The Edges!

- At the edges of an image we are missing pixels to form a neighbourhood



Border Pixels Treatment

- Mask operation near the image border

- Problem arises when part of the mask is located outside the image plane; to handle the problem:

  - **Truncate the image**

  - Discard the problem pixels (e.g. 512x512 input 510x510 output, if mask size is 3x3)

  - **Image Padding**

  - expand the input image by padding zeros or 255 (512x512 input 514x514output)

  - Image *padding is not good; creates artificial lines or edges on the border*

Border Pixels Treatment

▪ Replicate border pixels

- We normally use the gray levels of border pixels to fill up the expanded region (for 3x3 mask). For larger masks a border region equal to half of the mask size is mirrored on the expanded region.

102	102	130	143	123	115	...	...
102	102	130	143	123	115	...	...
93	93	...	...	...			
98	98	...					
82	82	...					
65	65						
...	...						
...	...						

Expanded area

Original image size (shaded area)

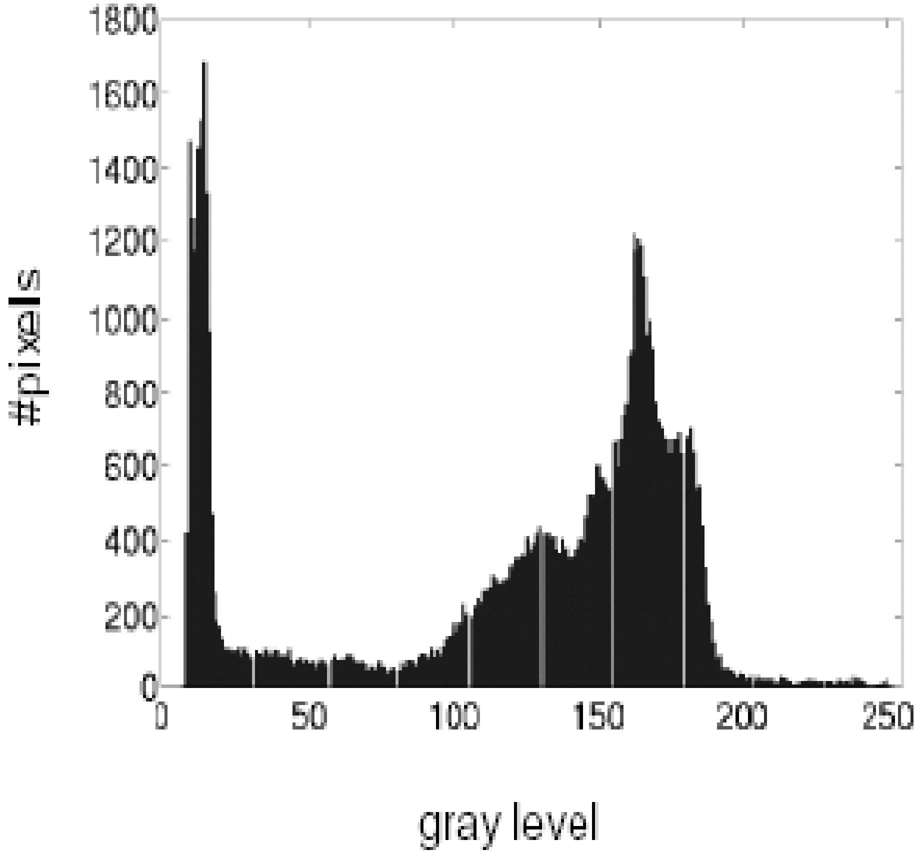
Histogram

- The histogram of an image represents the frequency of pixels in an image
- The histogram of an image shows the distribution of gray levels in the image
- Massively useful in image processing especially in image segmentation
- Histogram of an image with gray level (0 to L-1):

A discrete function $h(r_k) = n_k$, where r_k is the k^{th} gray level and n_k is the number of pixels in the image having gray level r_k.

- How to obtain histogram?
  - For B bit image, initialize 2^B counters with 0
  - Loop over all pixels x,y
  - When encouraging gray level f(x,y)=I, increment counter # i

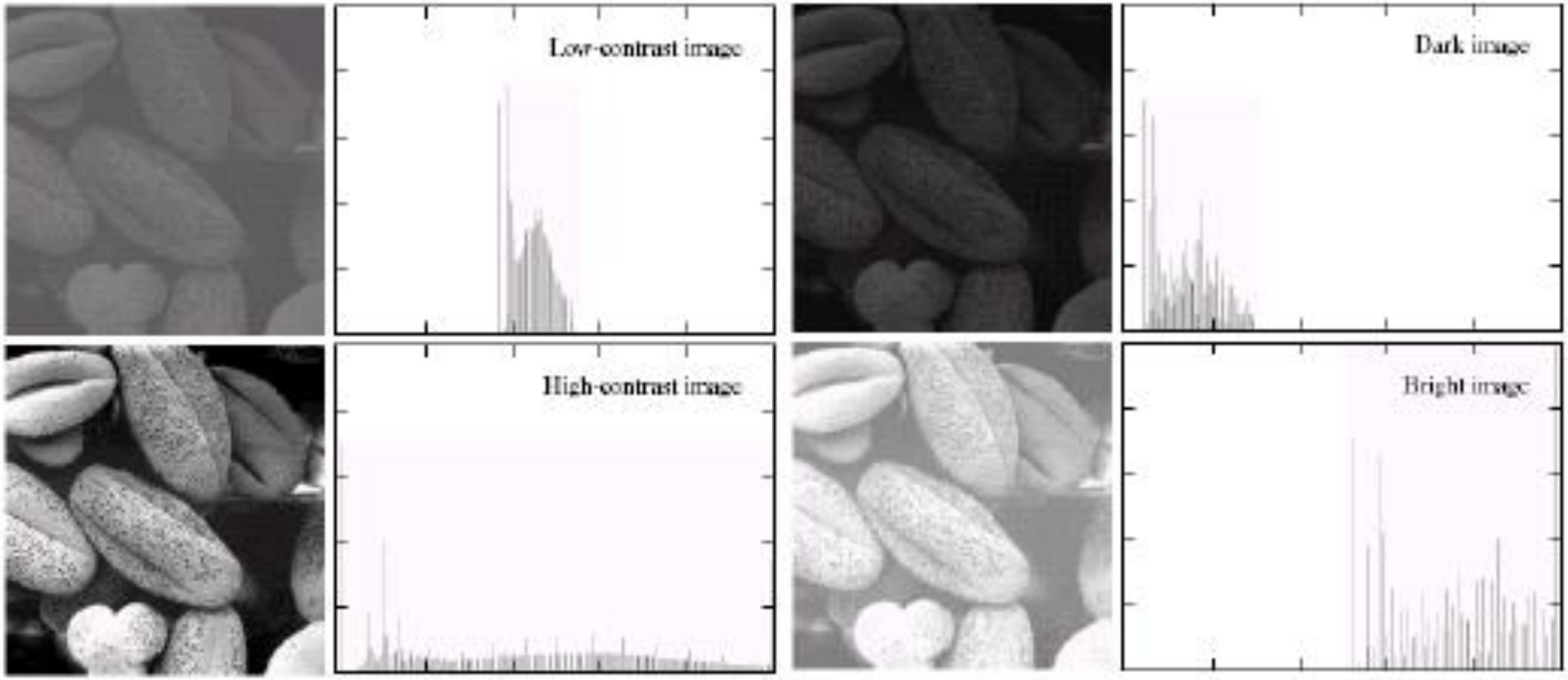
Histogram : Example



Cameraman image

Histogram : Example

- A selection of images and their Histograms
- Note that the high contrast image has the most evenly spaced histogram
- Histograms of low contrast images are located in certain portions and not in the entire gray scale range



Histogram Equalization

- The idea is to find a transformation **s=T(r)** to be applied to each pixel of the input image **f(x,y)** such that a uniform distribution of gray levels in the entire range results for the output image **g(x,y)**

The discrete approximation of the transformation function for histogram equalization is:

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) \quad \text{for} \quad 0 \le k \le L-1$$

where $p_r(r_j) = \frac{n_j}{n}, \quad j = 0, \ldots\ldots, L-1 \quad and \quad n = \sum_{j=0}^{L-1} n_j$

n_j : number of pixels with gray level r_j

n : total number of pixels

Histogram Equalization

- Spreading out the frequencies in an image (or equalising the image) is a simple way to improve dark or washed out images

- The formula for histogram equalisation is given where
  - r_k: input intensity
  - s_k: processed intensity
  - k: the intensity (e.g 0.0 – 1.0)
  - n_j: the frequency of intensity j
  - n: the sum of all frequencies

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

Histogram Equalization: Example

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

Value	Count	Value	Count	Value	Count	Value	Count	Value	Count
52	1	64	2	72	1	85	2	113	1
55	3	65	3	73	2	87	1	122	1
58	2	66	2	75	1	88	1	126	1
59	3	67	1	76	1	90	1	144	1
60	1	68	5	77	1	94	1	154	1
61	4	69	3	78	1	104	2		
62	1	70	4	79	2	106	1		
63	2	71	2	83	1	109	1		

Histogram Equalization: Example

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

Initial Image

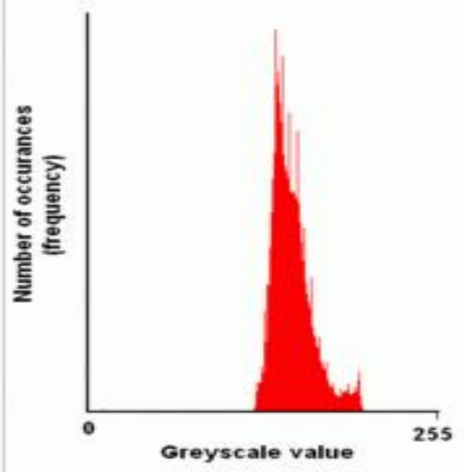
$$\begin{bmatrix} 0 & 12 & 53 & 93 & 146 & 53 & 73 & 166 \\ 65 & 32 & 12 & 215 & 235 & 202 & 130 & 158 \\ 57 & 32 & 117 & 239 & 251 & 227 & 93 & 166 \\ 65 & 20 & 154 & 243 & 255 & 231 & 146 & 130 \\ 97 & 53 & 117 & 227 & 247 & 210 & 117 & 146 \\ 190 & 85 & 36 & 146 & 178 & 117 & 20 & 170 \\ 202 & 154 & 73 & 32 & 12 & 53 & 85 & 194 \\ 206 & 190 & 130 & 117 & 85 & 174 & 182 & 219 \end{bmatrix}$$

Image After Equalization

Notice that the minimum value (52) is now 0 and the maximum value (154) is now 255.

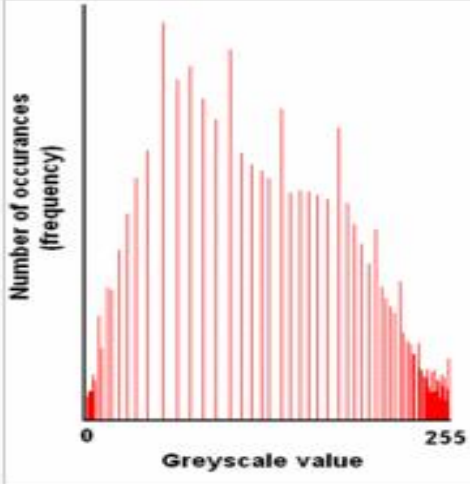
Histogram Equalization: Example

An unequalized image



Corresponding histogram

The same image after histogram equalization



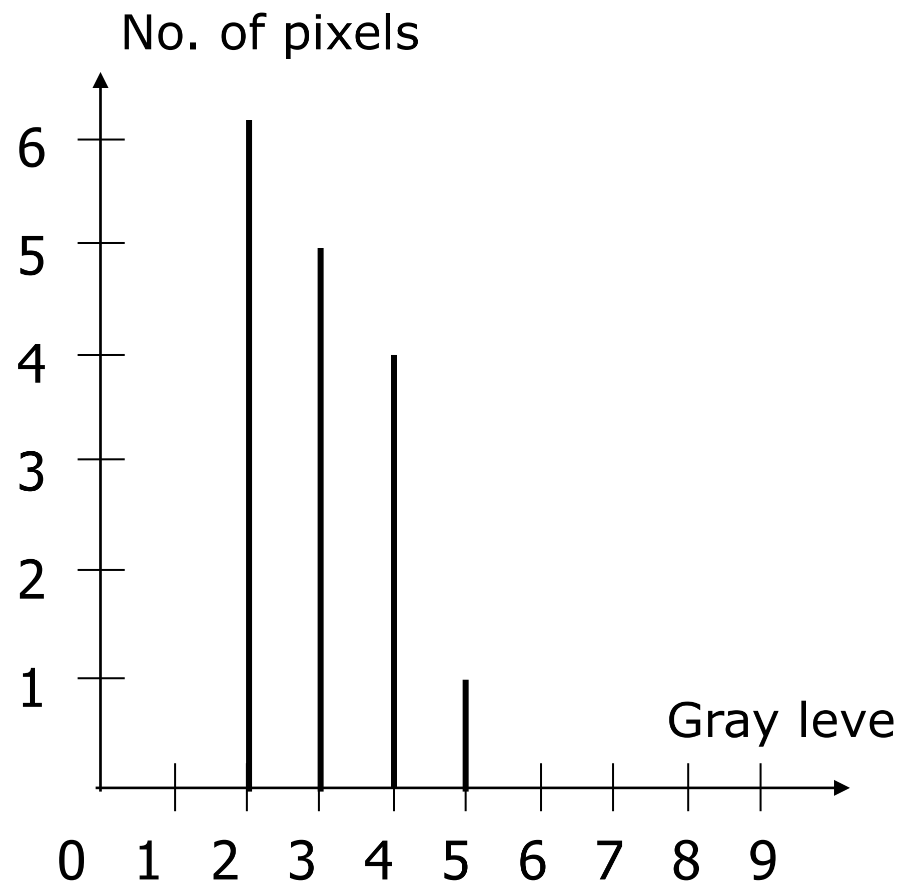
Corresponding histogram

Histogram Equalization: Example

2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

4x4 image

Gray scale = [0,9]



histogram

Histogram Equalization: Example

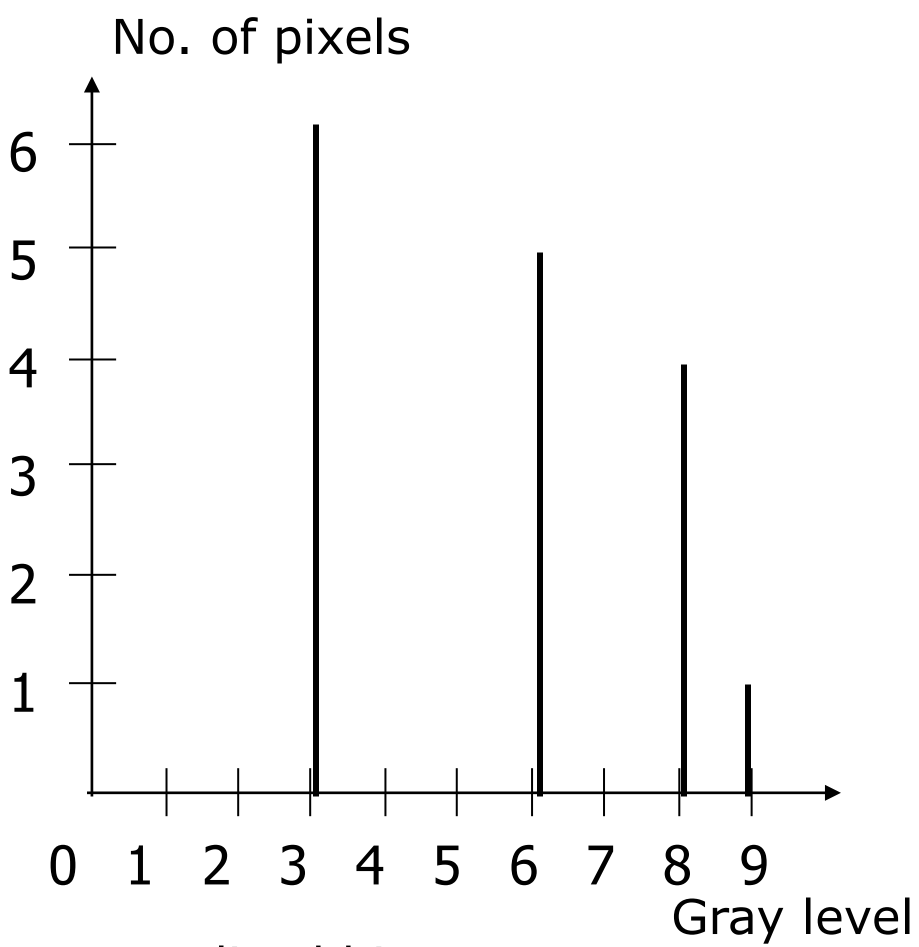
Gray Level(j)	0	1	2	3	4	5	6	7	8	9
No. of pixels	0	0	6	5	4	1	0	0	0	0
$\sum_{j=1}^{k} n_j$	0	0	6	11	15	16	16	16	16	16
$s = \sum_{j=1}^{k} \frac{n_j}{n}$	0	0	6 / 16	11 / 16	15 / 16	16 / 16	16 / 16	16 / 16	16 / 16	16 / 16
s* x *9	0	0	3.3 ≈3	6.1 ≈6	8.4 ≈8	9	9	9	9	9

Histogram Equalization: Example

3	6	6	3
8	3	8	6
6	3	6	9
3	8	3	8

Output image

Gray scale = [0,9]



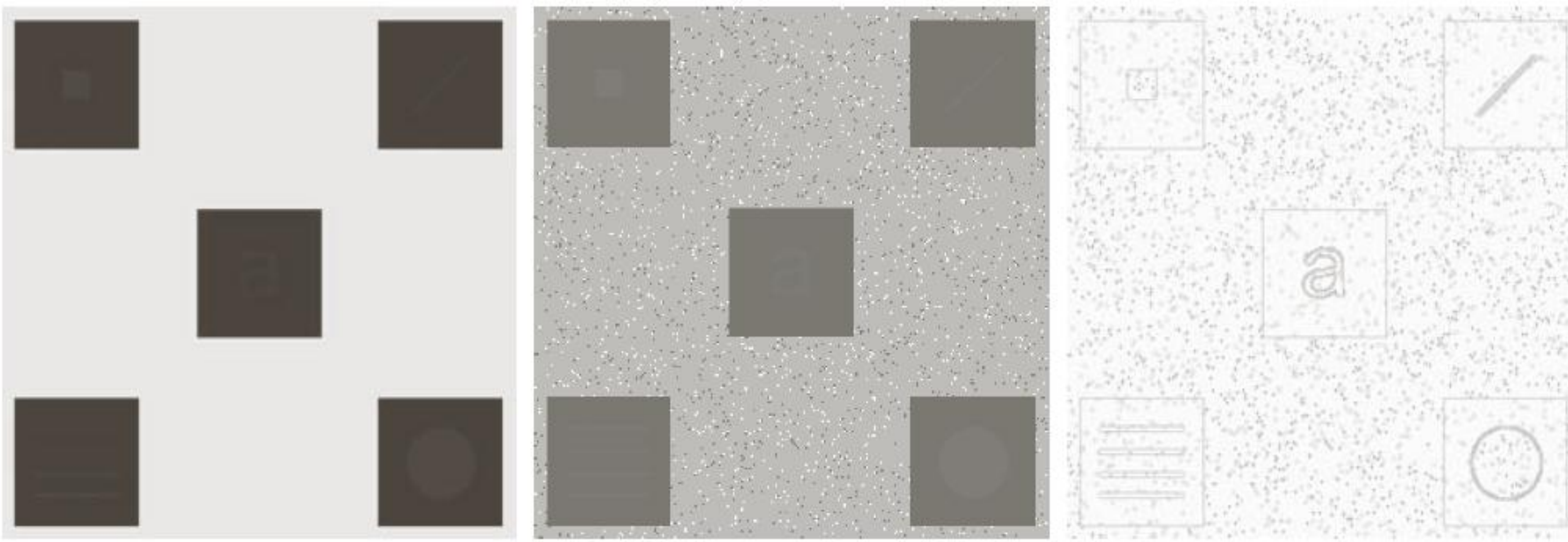
Equalized histogram

Local Enhancement through Histogram Processing

- The histogram processing methods discussed earlier are global in the sense that pixels are modified by a transformation function based on the gray level content of the entire image
- Although this global approach is suitable for overall enhancement. There are cases in which it is necessary to enhance details over small areas in the image
- The histogram equalization can be easily adapted to local enhancement
- The procedure is to define a neighborhood and move the center of this area from pixel to pixel
- At each location, the histogram of the points in the neighborhood is computed and then histogram transformation function is obtained

Local Enhancement through Histogram Processing

- Map the intensity of the pixel centered in the neighborhood
- Move to the next location and repeat the procedure



a b c

FIGURE 3.26 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization applied to (a), using a neighborhood of size 3 × 3.

Use of Histogram Statistics for Image Enhancement

Average Intensity

$$m = \sum_{i=0}^{L-1} r_i \, p(r_i) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$$

Variance

$$\sigma^2 = \sum_{i=0}^{L-1} (r_i - m)^2 p(r_i) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[f(x, y) - m \right]^2$$

Use of Histogram Statistics for Image Enhancement

Local average intensity

$$m_{s_{xy}} = \sum_{i=0}^{L-1} r_i \, p_{s_{xy}}(r_i)$$

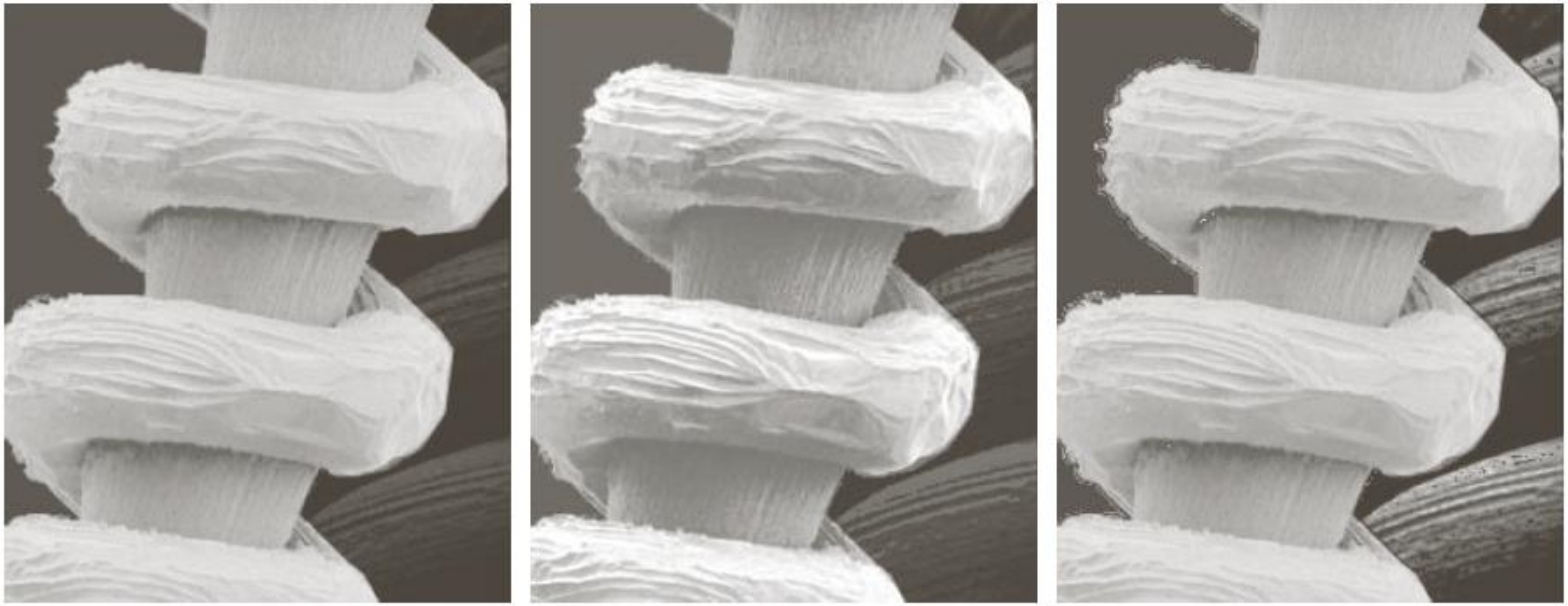
s_{xy} denotes a neighborhood

Local variance

$$\sigma_{s_{xy}}^2 = \sum_{i=0}^{L-1} (r_i - m_{s_{xy}})^2 \, p_{s_{xy}}(r_i)$$

- The following statistical conditions are used here for enhancement
- g(x,y) = { E.f(x,y) if $m_{S_{xy}} <= k_0 M_G$ AND $k_1 D_G <= \sigma_{s_{xy}} <= k_2 D_G$ }
- g(x,y) = f(x,y) otherwise

Use of Histogram Statistics for Image Enhancement



a b c

FIGURE 3.27 (a) SEM image of a tungsten filament magnified approximately 130×. (b) Result of global histogram equalization. (c) Image enhanced using local histogram statistics. (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

Any question