Course: Digital Image Processing

By: Dr. Muhammad Fayaz

Department of Computer Science, University of Central Asia

RECAP

- What is an Image?
- Representation of an Image
- Types of Images
  - Binary Image
  - Grayscale Image
  - RGB Image
- Image Sampling and Quantization
- Spatial Resolution
- Libraries
- Relationship Between Pixels
- Neighborhoods: N_4, N_D, and N_8
- Adjacency: N_4, and N_8 , and Mix-adjacency
- Distances: N_4 (City Block Distance), and N_8 (Chess Board Distance)
- Region, Boundary, and Edge
- Types of Image Enhancement Operations
- Min-Max Marching
- Histogram Equalization
- Histogram Matching
- Image matching
- Image Restoration
- Convolution
- Noise
- Padding

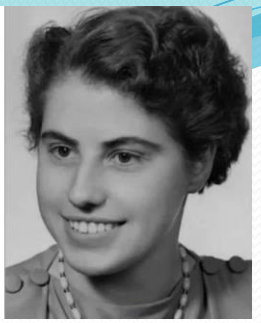
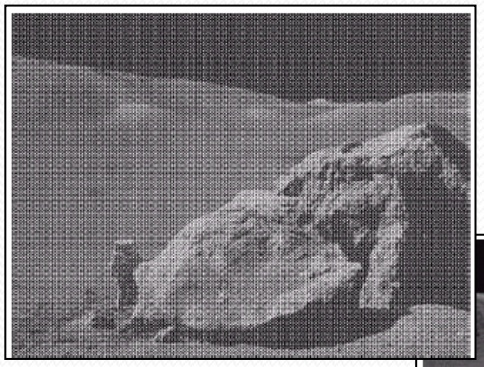
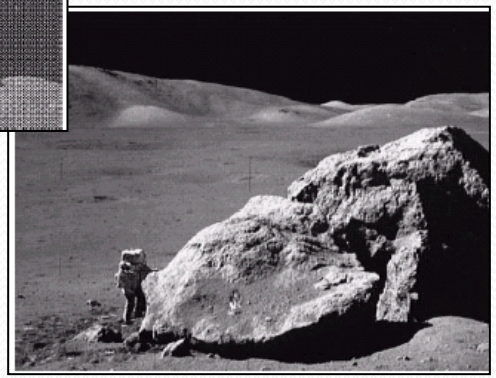


Image Restoration

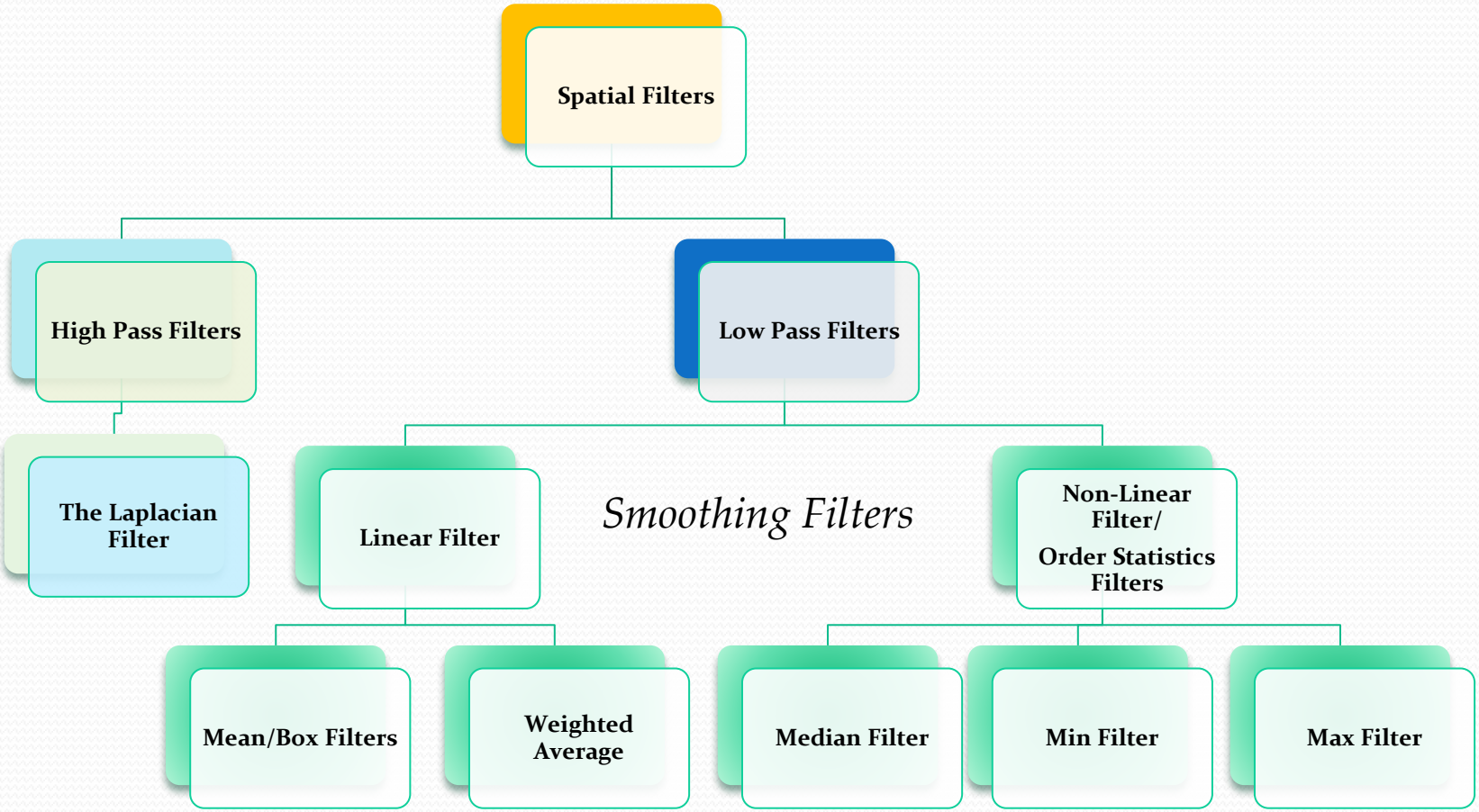




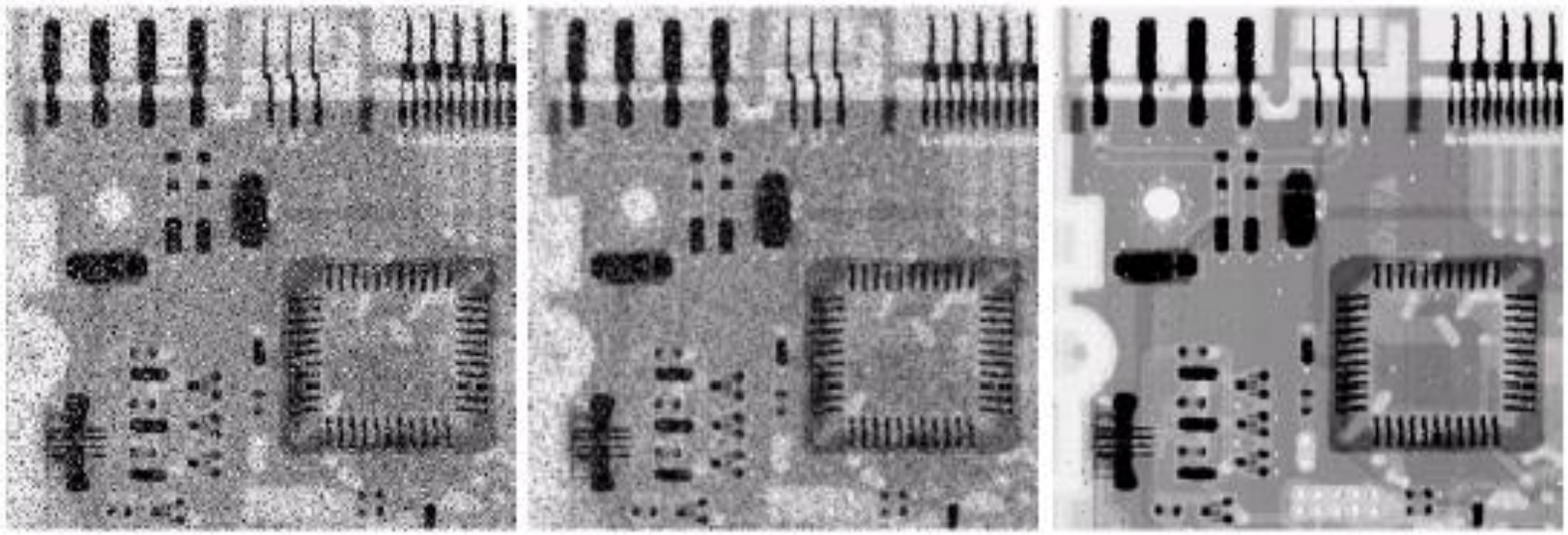
Assignment

Noise Reduction

- Noise reduction refers to the various methods and techniques used to enhance the quality of digital images by reducing unwanted noise while preserving important image features.



Median Filtering (Example)

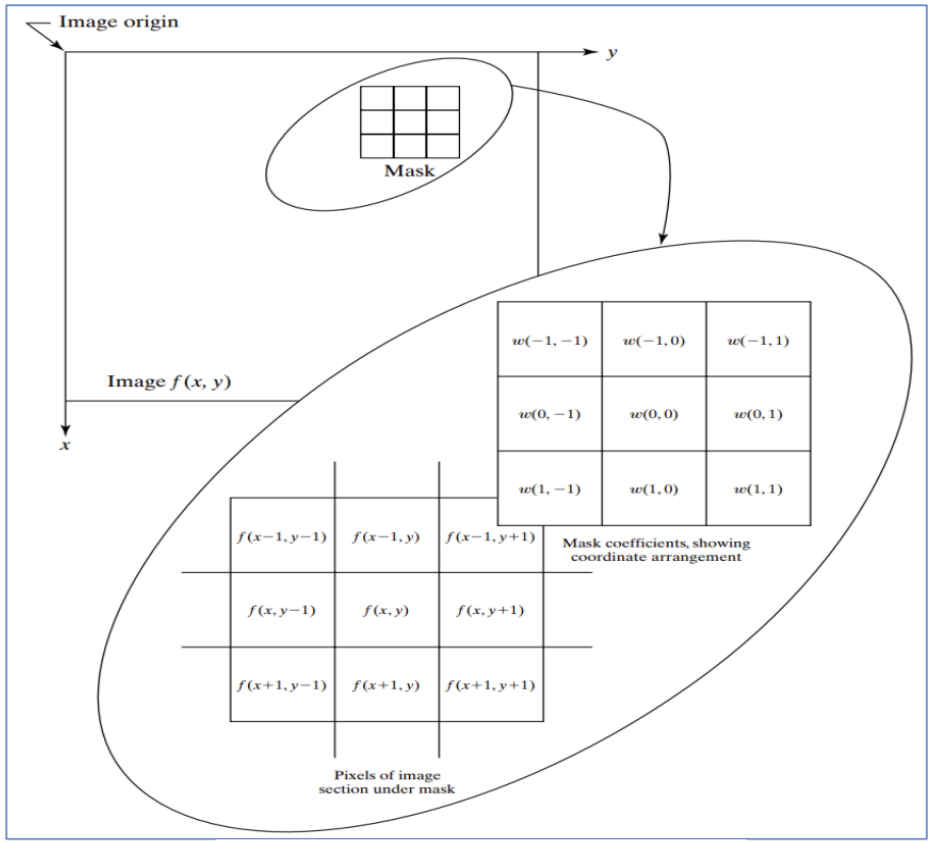


a b c

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.)

Spatial Filtering

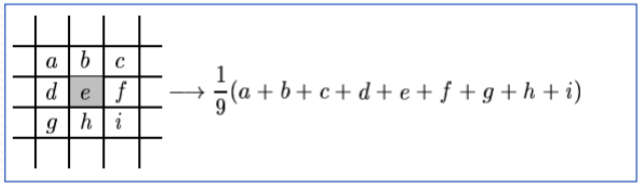
- **Linear Filters:** Operation consists of multiplying each pixel in the mask by corresponding element in the neighborhood, and adding up all these products.

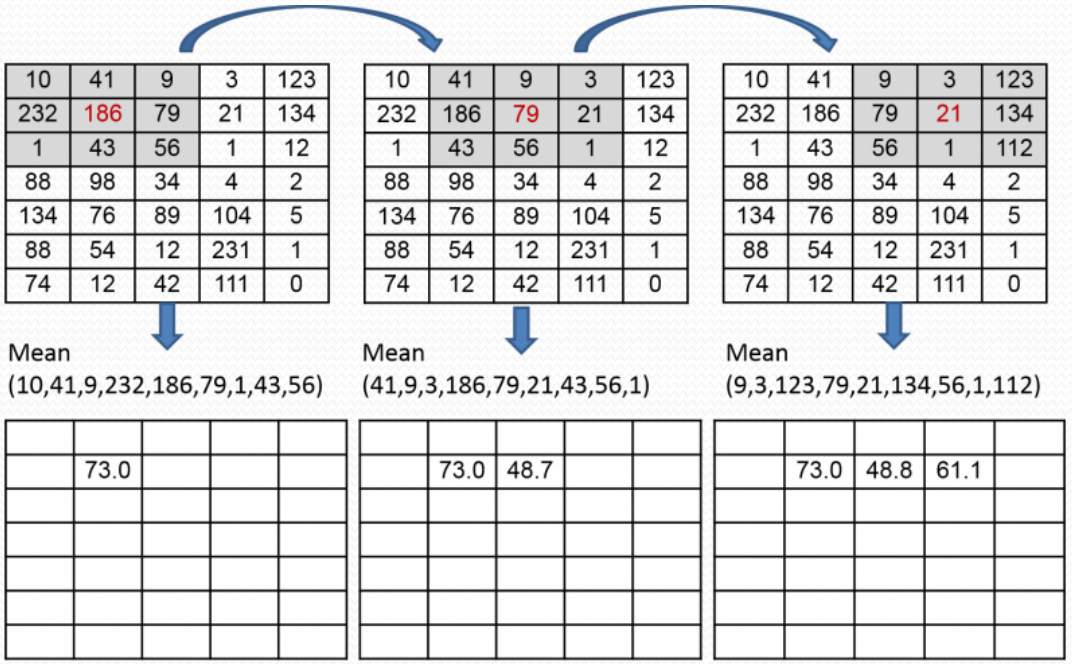


Mean/Averaging Filter

$$g(x,y) = \frac{1}{m\,x\,n} \sum_{x=-a}^{a} \sum_{y=-b}^{b} f(x,y) \qquad \textit{Linear Filter} \text{ "Average/mean Filter"}$$

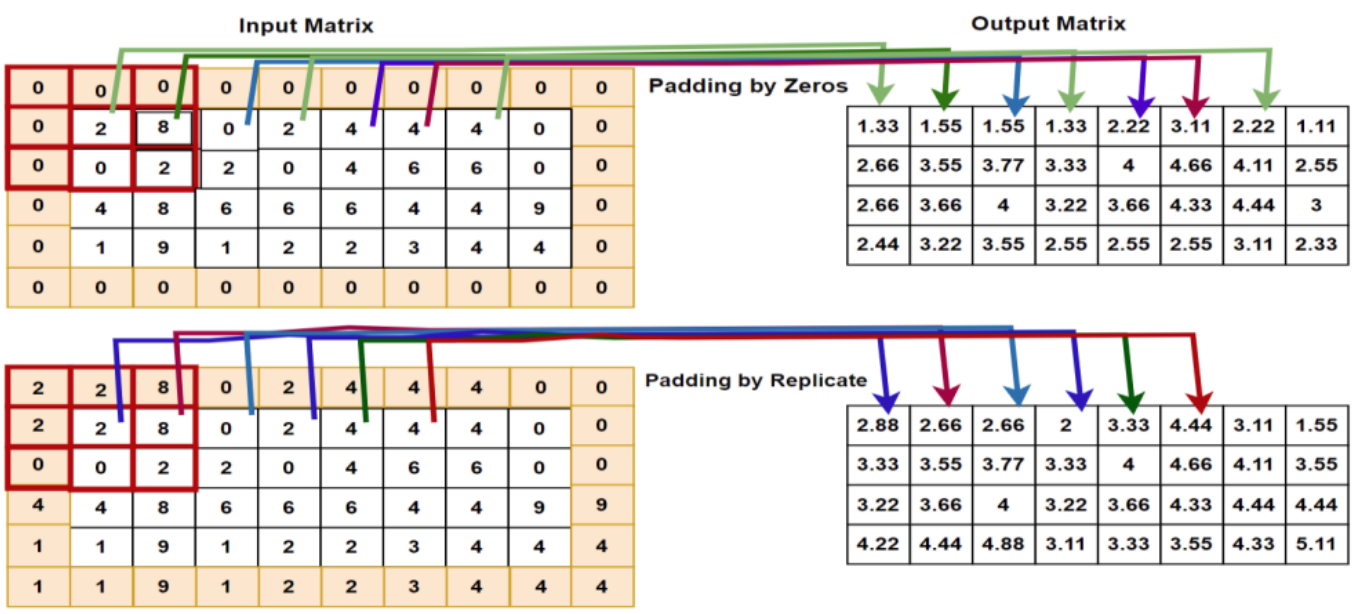
- **Average (or mean):** filtering is a method of 'smoothing' images by reducing the amount of intensity variation between neighboring pixels.
- The average filter works by moving through the image pixel by pixel, replacing each value with the average value of neighboring pixels, including itself.





Mean/Averaging Filter

$$g(x,y) = \frac{1}{m\,x\,n} \sum_{x=-a}^{a} \sum_{y=-b}^{b} f(x,y) \qquad \textit{Linear Filter} \text{ "Average/mean Filter"}$$



0	0	0	0	0	0	0	0	0	0
0	2	8	0	2	4	4	4	0	0
0	0	2	2	0	4	6	6	0	0
0	4	8	6	6	6	4	4	9	0
0	1	9	1	2	2	3	4	4	0
0	0	0	0	0	0	0	0	0	0

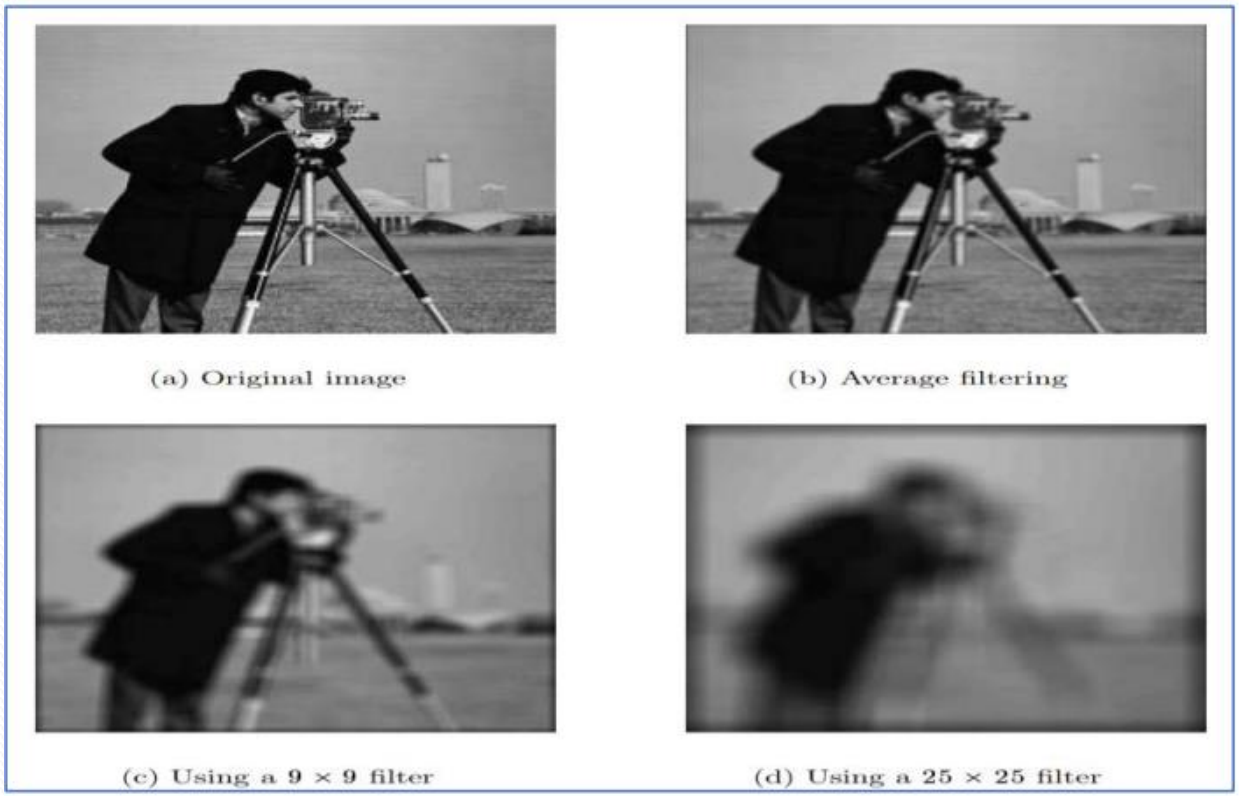
1.33	1.55	1.55	1.33	2.22	3.11	2.22	1.11
2.66	3.55	3.77	3.33	4	4.66	4.11	2.55
2.66	3.66	4	3.22	3.66	4.33	4.44	3
2.44	3.22	3.55	2.55	2.55	2.55	3.11	2.33

2	2	8	0	2	4	4	4	0	0
2	2	8	0	2	4	4	4	0	0
0	0	2	2	0	4	6	6	0	0
4	4	8	6	6	6	4	4	9	9
1	1	9	1	2	2	3	4	4	4
1	1	9	1	2	2	3	4	4	4

2.88	2.66	2.66	2	3.33	4.44	3.11	1.55
3.33	3.55	3.77	3.33	4	4.66	4.11	3.55
3.22	3.66	4	3.22	3.66	4.33	4.44	4.44
4.22	4.44	4.88	3.11	3.33	3.55	4.33	5.11

Mean/Averaging Filter

$$g(x,y) = \frac{1}{m\,x\,n} \sum_{x=-a}^{a} \sum_{y=-b}^{b} f(x,y)$$ *Linear Filter* "Average/mean Filter"



(a) Original image

(b) Average filtering

(c) Using a 9×9 filter

(d) Using a 25×25 filter

Weighted Average Filter

- In weighted average filter, we gave more weight to the centre value, due to which the contribution of centre becomes more than the rest of values.
- Due to the weighted average filtering, we can control the blurring of image.

ω_1	ω_2	ω_3
ω_4	ω_5	ω_6
ω_7	ω_8	ω_9

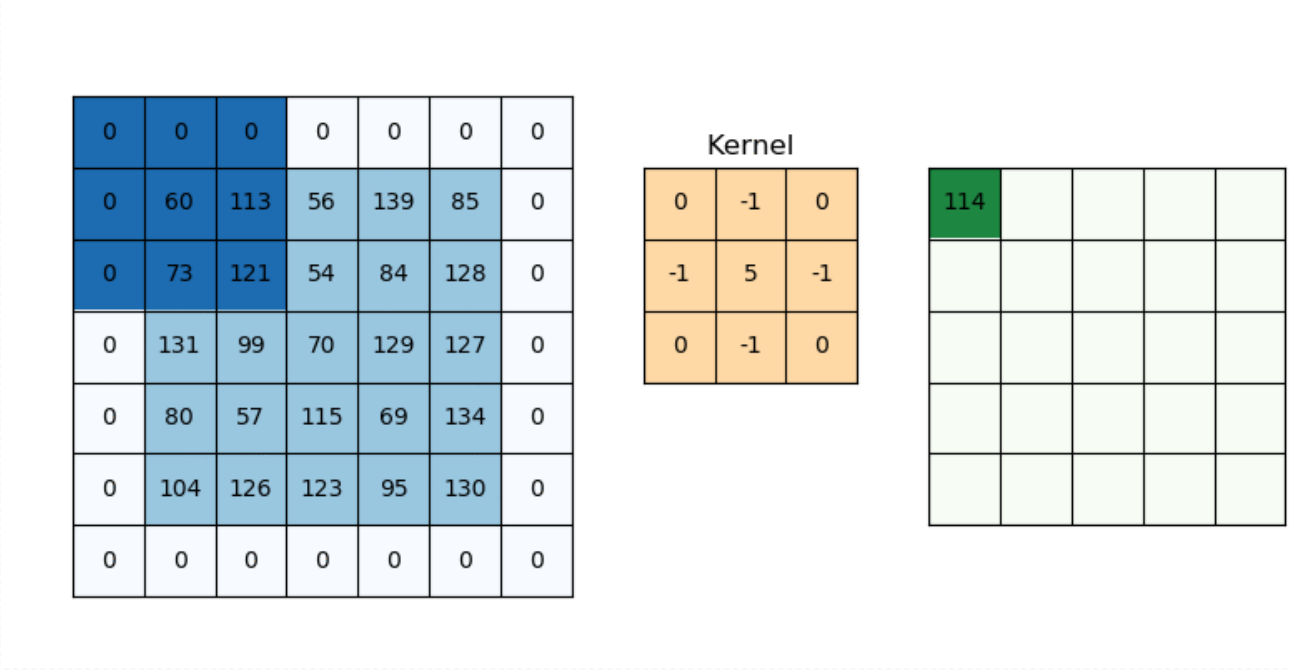
(a) A representation of a general 3×3 filter mask

$$\frac{1}{16}\times$$

1	2	1
2	4	2
1	2	1

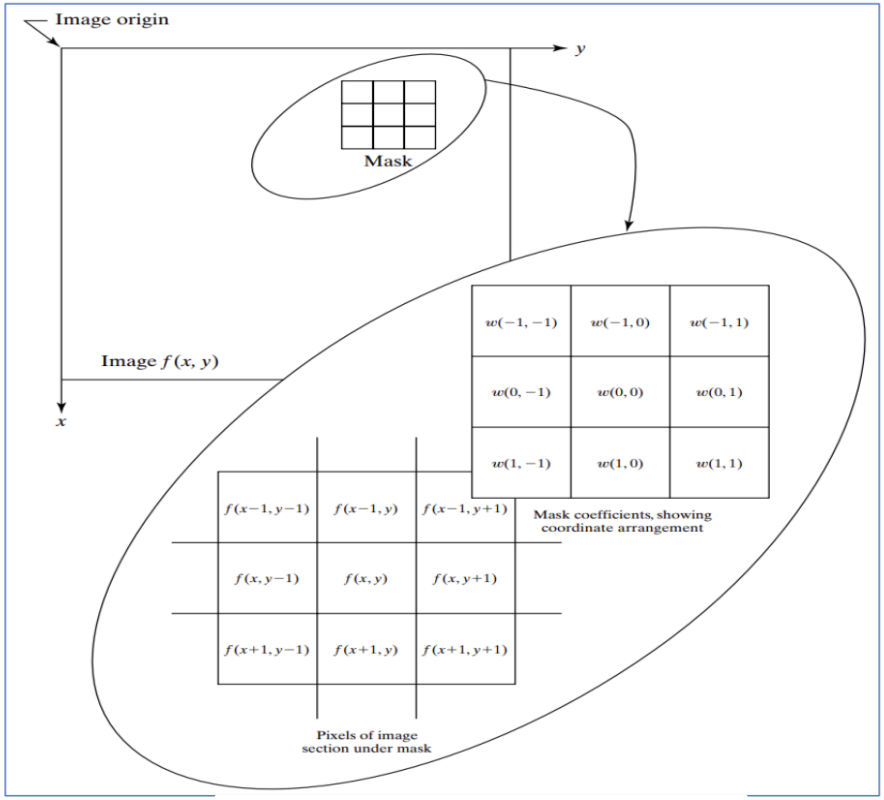
(b) A 3×3 weighted average filter mask

Weighted Average Filter



Spatial Filtering

- **Non-Linear Filters:** The non-linear spatial filters are based on linear operations such as taking the maximum element in the mask in max filter, or taking the minimum element in the min filter or taking the median element in the median filter.



Median Filter

- A median filter is a common noise reduction technique used in image processing to reduce impulsive noise, such as salt-and-pepper noise, from digital images.
- It works by replacing the value of a pixel with the median value of its neighbouring pixels within a defined window or kernel.
- The median is the middle value in a sorted list of numbers, making it effective at removing outliers or extreme values that could be caused by noise.
- **Here's how a median filter works:**
  - ***Sliding Window*:** The filter slides through the entire image, cantered on each pixel. The size of the window or kernel is defined by the user and determines the area over which the median is calculated.
  - ***Neighborhood Pixel Values:*** The pixel values within the window are collected and sorted in ascending order.
  - ***Median Calculation:*** The median value (middle value) of the sorted pixel values is then selected as the new value for the center pixel being processed.

Median Filtering

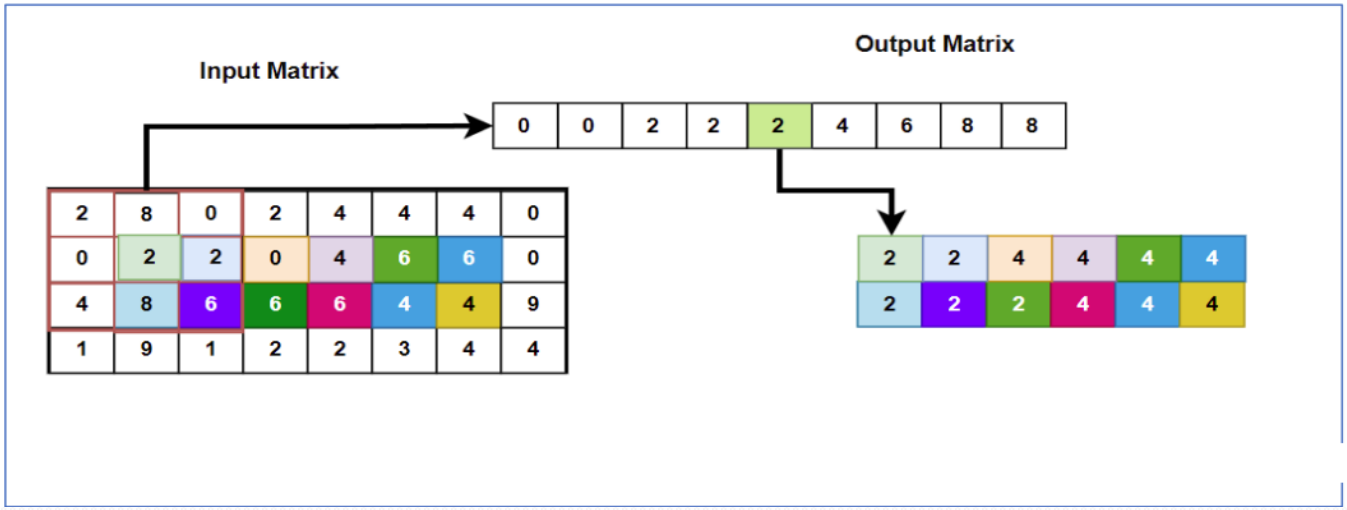
10	20	20
20	15	20
20	25	100

Output = ? **20**

$$g(x,y) = Med(f(x,y)) \qquad Non-linear\ Filter\ Median\ Filter$$

Median Filter

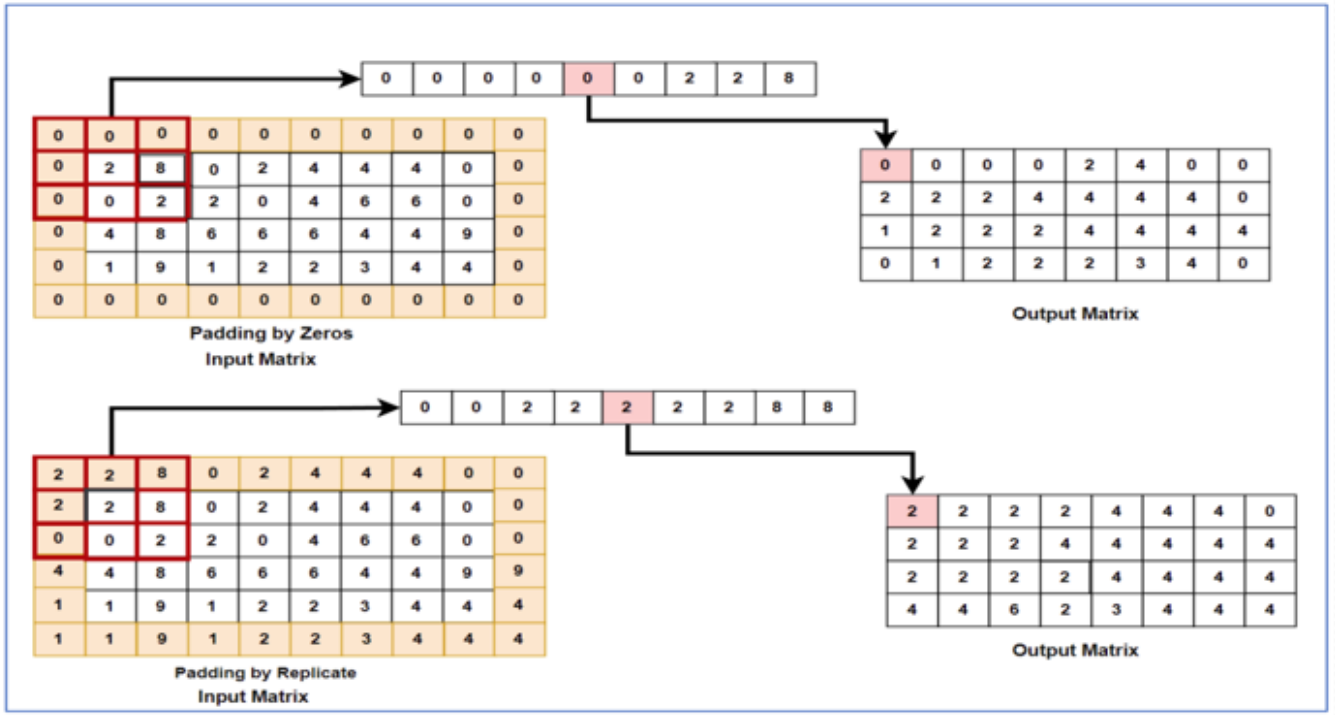
- **Median Filter with unpadded image**



$$g(x,y) = Med(f(x,y)) \qquad Non-linear\ Filter\ Median\ Filter$$

Median Filter

- **Median Filter with padded image**

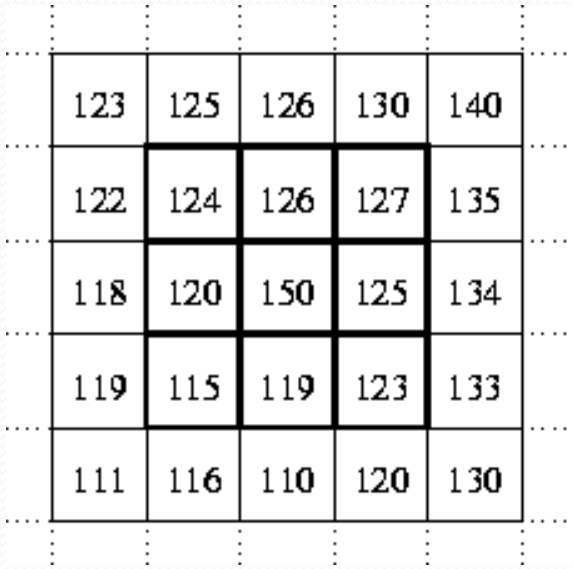


$$g(x,y) = Med\big(f(x,y)\big) \qquad Non-linear\ Filter\ Median\ Filter$$

Median Filter

- **Median Filter with unpadded image**

Min Filter

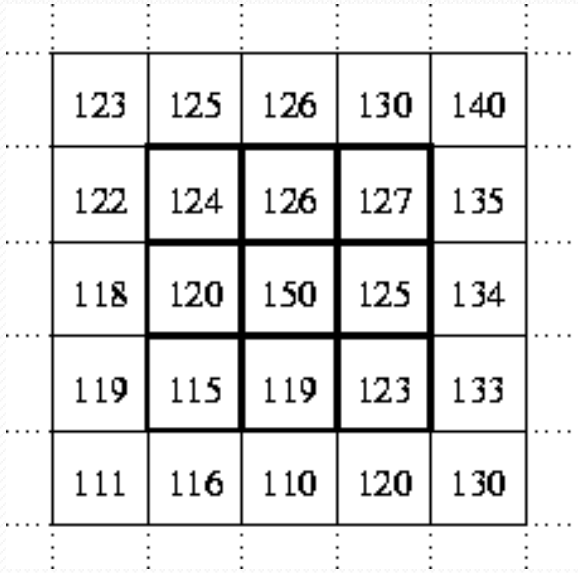


Neighborhood values:

115, 119, 120, 123, 124, 125, 126, 127, 150

Max: 127, Min: 115

Max Filter



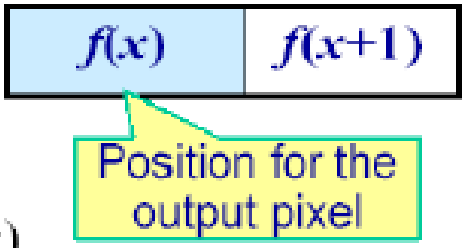
Neighborhood values:

115, 119, 120, 123, 124, 125, 126, 127, 150

Max: 127, Min: 115

First and Second Order Derivatives

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



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

$$= [f(x+2) - f(x+1)] - [f(x+1) - f(x)]$$

$$= f(x+2) - 2f(x+1) + f(x)$$

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

Laplacian for Image Enhancement

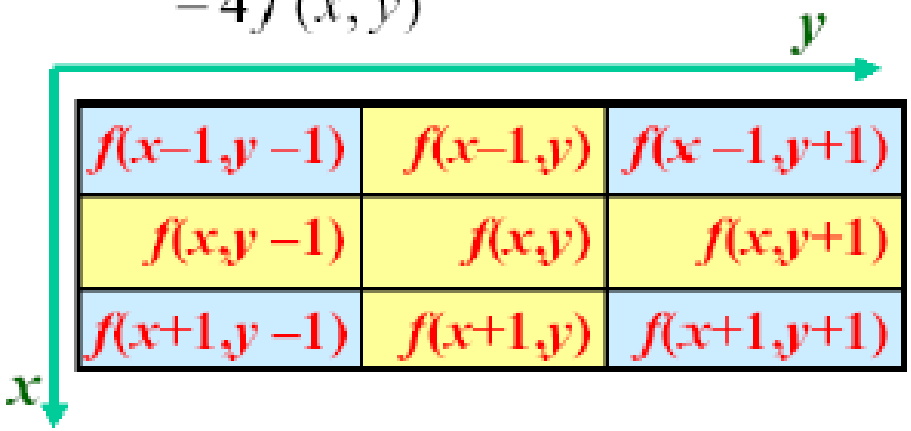
Laplacian operator

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

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

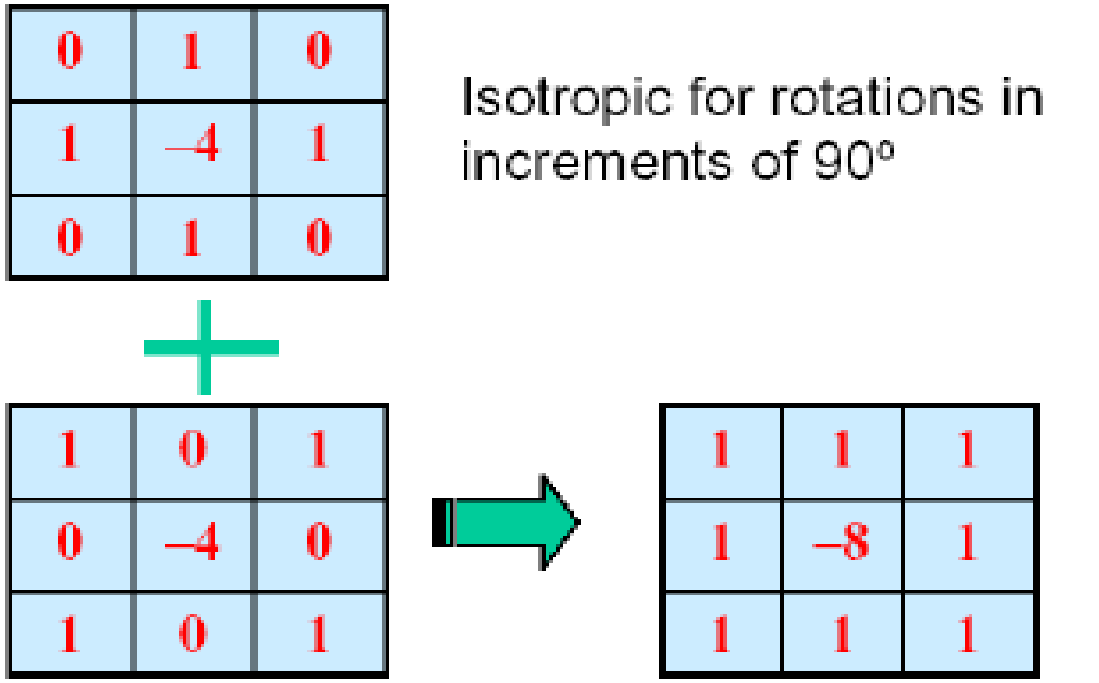
$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

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



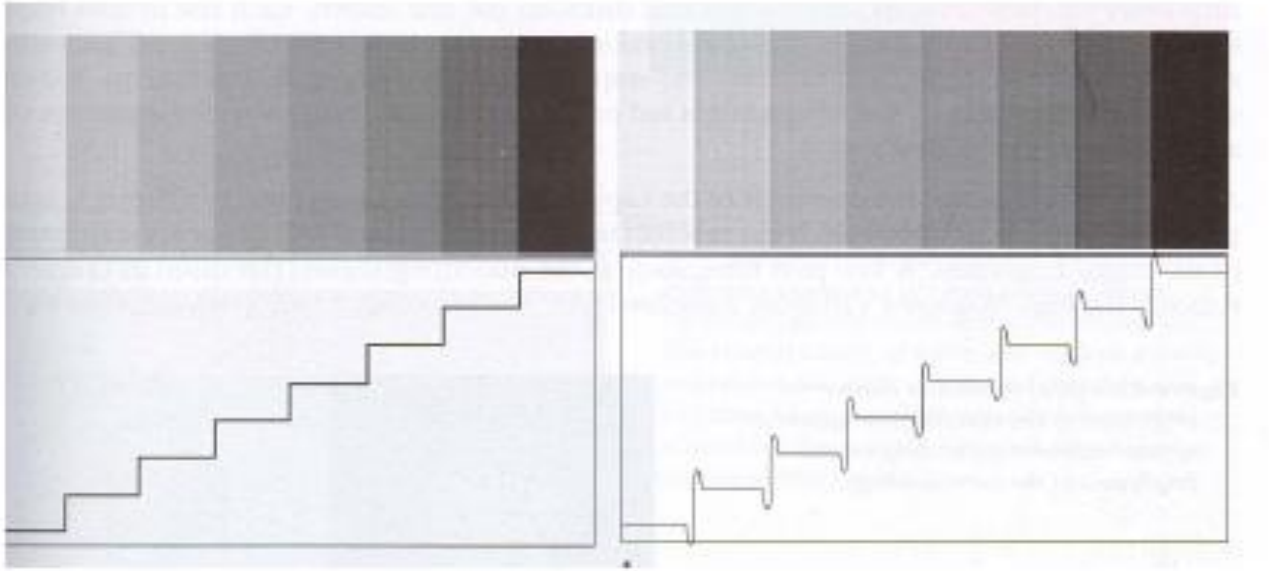
0	1	0
1	–4	1
0	1	0

Laplacian for Image Enhancement



Enhancement by 2nd Derivative (Example)

Applying the second derivative through 1-D kernel:									-1	2	-1
On a sequence	2	2	2	2	2	4	6	6	6	6	6
Will give	0	0	0	0	-2	0	+2	0	0	0	0
Adding them	2	2	2	2	0	4	8	0	0	0	0



Input image uniform gray bands

Enhanced image

Laplacian for Image Enhancement

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

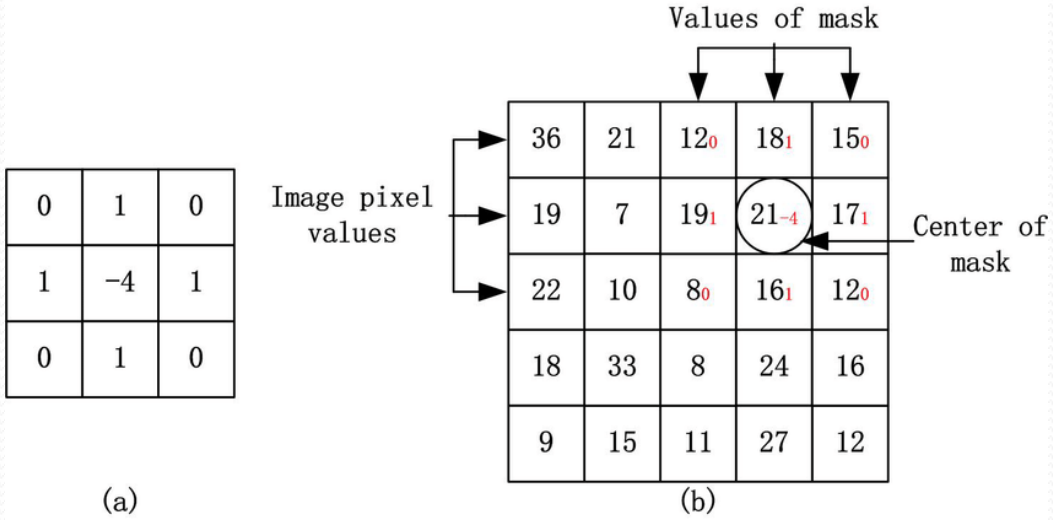
0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

a b
c d

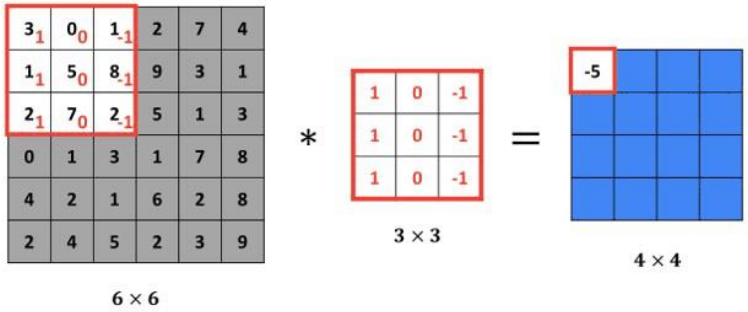
FIGURE 3.39
(a) Filter mask used to implement the digital Laplacian, as defined in Eq. (3.7-4). (b) Mask used to implement an extension of this equation that includes the diagonal neighbors. (c) and (d) Two other implementations of the Laplacian.

Laplacian for Image Enhancement



0	1	0
1	−4	1
0	1	0

36	21	12_0	18_1	15_0
19	7	19_1	21_{-4}	17_1
22	10	8_0	16_1	12_0
18	33	8	24	16
9	15	11	27	12



$3 \times 1 + 1 \times 1 + 2 \times 1 + 0 \times 0 + 5 \times 0 + 7 \times 0 + 1 \times -1 + 8 \times -1 + 2 \times -1 = -5$

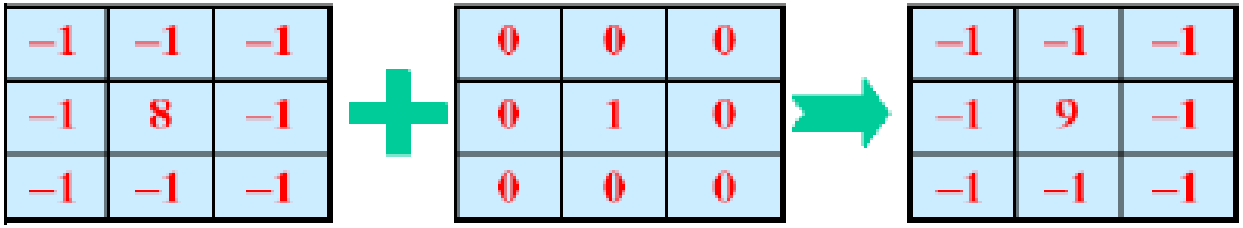
Laplacian for Image Enhancement

To obtain the enhanced image

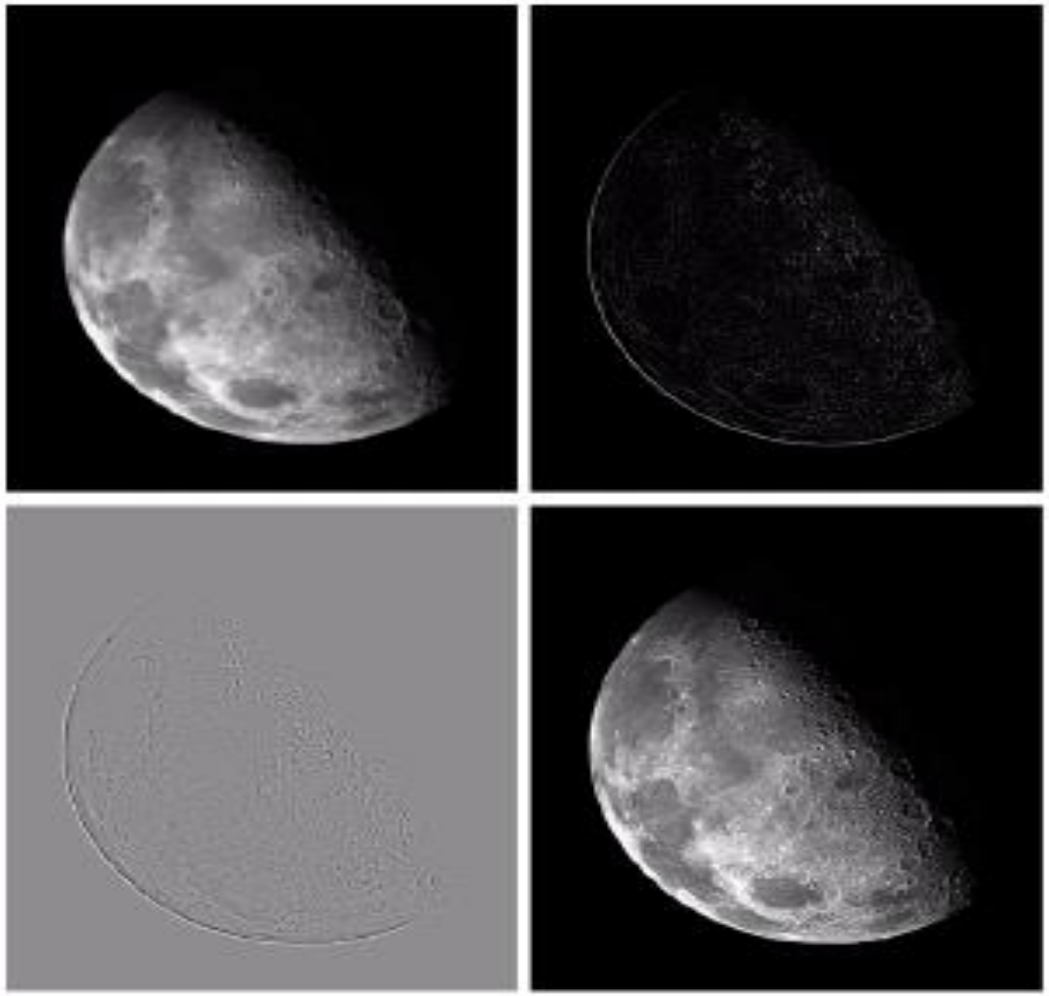
$$g(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y), & w_5<0 \\ f(x,y)+\nabla^2 f(x,y), & w_5>0 \end{cases}$$

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

In this way, background tonality can be perfectly preserved while details are enhanced



Laplacian for Image Enhancement (Example)



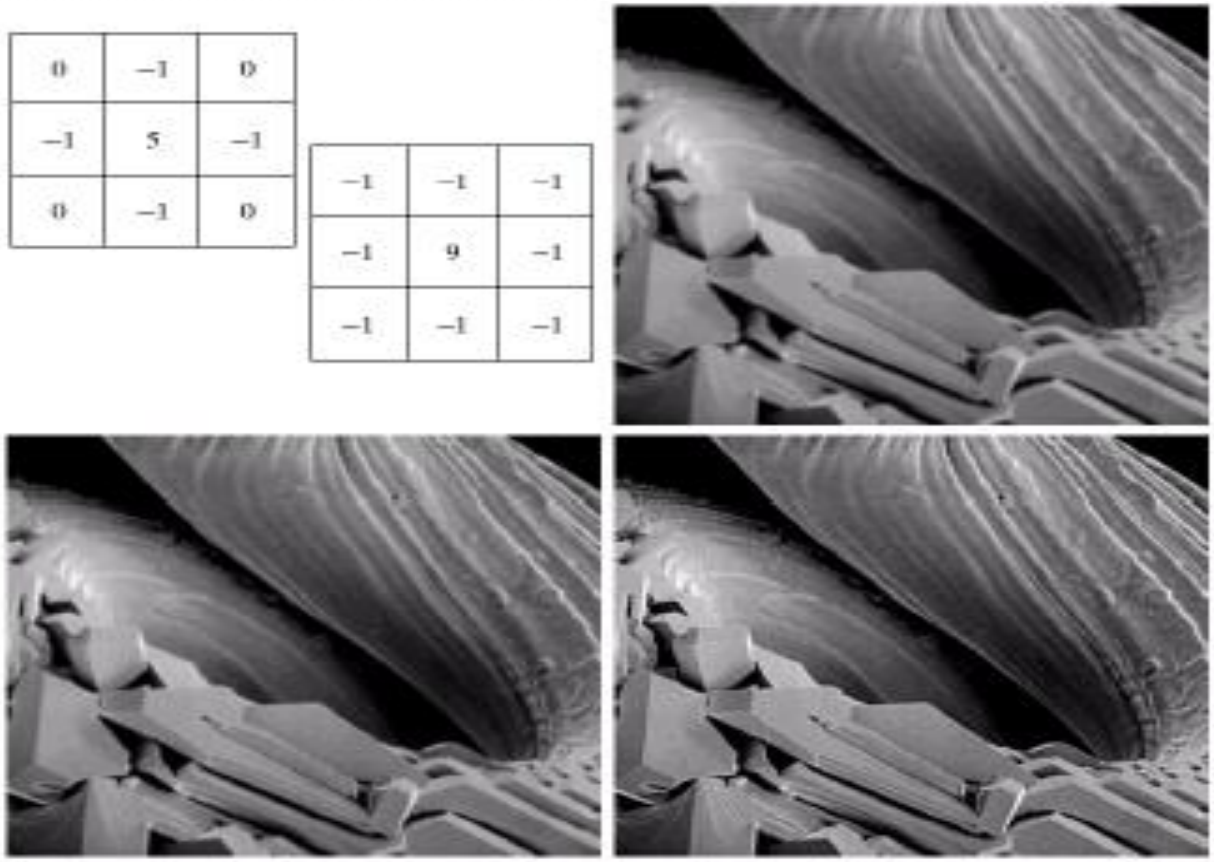
a b
c d

FIGURE 3.40
(a) Image of the North Pole of the moon.
(b) Laplacian-filtered image.
(c) Laplacian image scaled for display purposes.
(d) Image enhanced by using Eq. (3.7-5).
(Original image courtesy of NASA.)

Laplacian for Image Enhancement (Example)

0	−1	0
−1	5	−1
0	−1	0

−1	−1	−1
−1	9	−1
−1	−1	−1



a b c
d e

FIGURE 3.41 (a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Results of filtering with the masks in (a) and (b), respectively. Note how much sharper (e) is than (d). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

IMAGE RESIZING

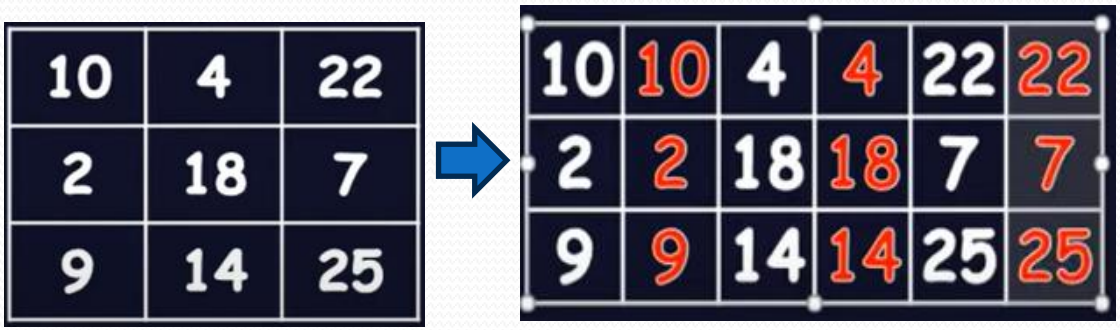
Image Resizing

- Image resizing techniques in image processing are used to change the dimensions of an image while preserving its visual content and aspect ratio.
- Resizing is commonly required to adapt images for various display or processing purposes.
- Here are some commonly used image resizing techniques:

  - *Nearest-Neighbor Interpolation*
  - *Bilinear Interpolation*

Nearest-Neighbor Interpolation

- Nearest-neighbor interpolation is a simple image resizing technique used in computer graphics and image processing.
- It involves determining the value of a new pixel in a resized image by selecting the value of the nearest pixel from the original image.
- This method is straightforward but can lead to blocky and jagged artifacts, especially when reducing the image size.
  - ***Duplicate Columns***
  - ***Duplicate Rows***



Duplicate Columns

Duplicate Rows

Nearest-Neighbor Interpolation

- Nearest-neighbour interpolation

10	10	4	4	22	22
10	10	4	4	22	22
2	2	18	18	7	7
2	2	18	18	7	7
9	9	14	14	25	25
9	9	14	14	25	25

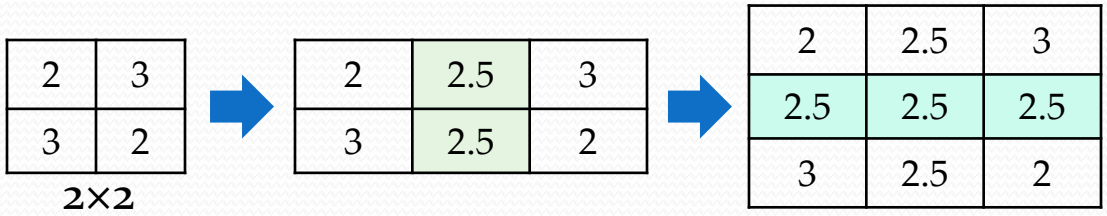
- **Calculate Scaling Factors:** The scaling factors for both dimensions are $\frac{3}{6} = \frac{1}{2}$.
- **Generate Output Pixels:** For each pixel in the output matrix, find the nearest neighbor corresponding pixel in the input matrix based on the scaling factors. Copy the values of the nearest input pixels to the output.
- .

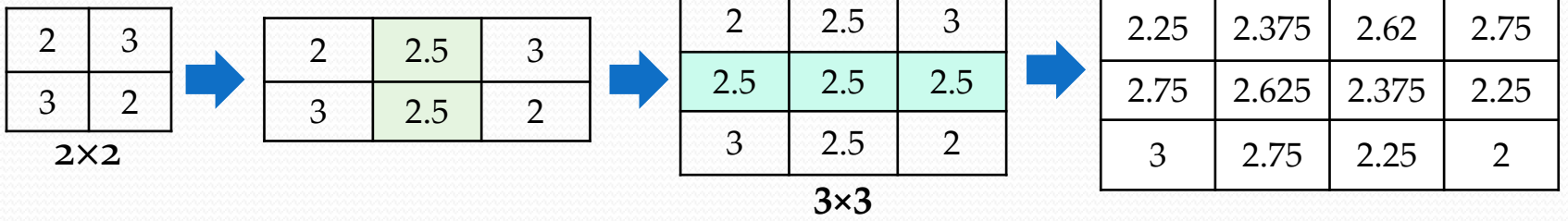
- Let's calculate the values of each output pixels
- **Output pixel at (1,1):** Corresponding input pixel: (1,1) in the input matrix. Value: 10
- **Output pixel at (1,2):** Corresponding input pixel: (1,3) in the input matrix. Value: 4
- **Output pixel at (1,3):** Corresponding input pixel: (1,5) in the input matrix. Value: 22
- **Output pixel at (2,1):** Corresponding input pixel: (3,1) in the input matrix. Value: 2
- **Output pixel at (2,2):** Corresponding input pixel: (3,3) in the input matrix. Value: 18
- **Output pixel at (2,3):** Corresponding input pixel: (3,5) in the input matrix. Value: 7
- **Output pixel at (3,1):** Corresponding input pixel: (5,1) in the input matrix. Value: 9
- **Output pixel at (3,2):** Corresponding input pixel: (5,3) in the input matrix. Value: 14
- **Output pixel at (3,3):** Corresponding input pixel: (5,5) in the input matrix. Value: 2

10	4	22
2	18	7
9	14	25

Bilinear Interpolation

- Bilinear interpolation is a commonly used technique in image processing and computer graphics for resizing images or changing their spatial resolution.
- It's used to estimate pixel values at non-integer coordinates based on the values of neighbouring pixels.
- Bilinear interpolation provides a smoother and more accurate result compared to nearest-neighbour interpolation.





Bilinear Interpolation

- $f(x, y_1) = \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21})$
- $f(x, y_2) = \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22})$

THANKS