Computer Vision - CS452

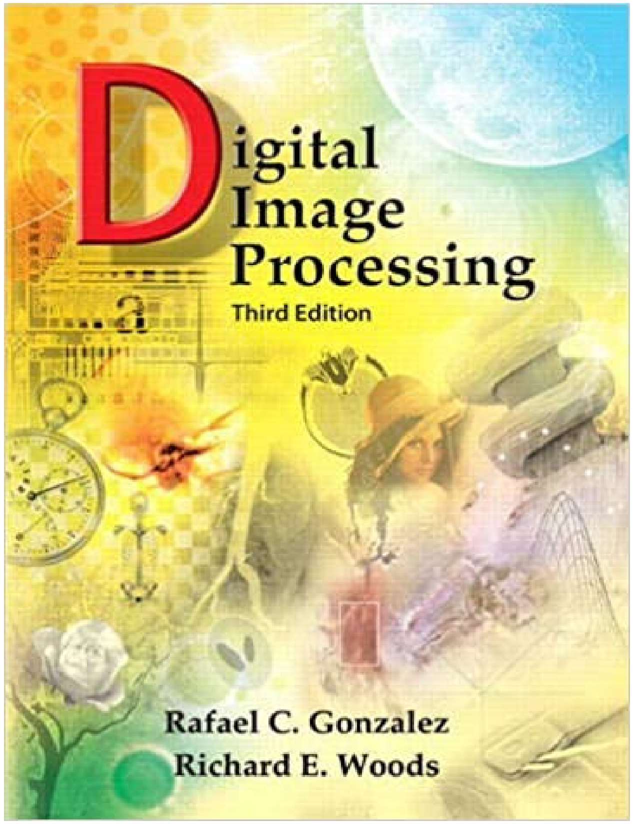
Introduced by

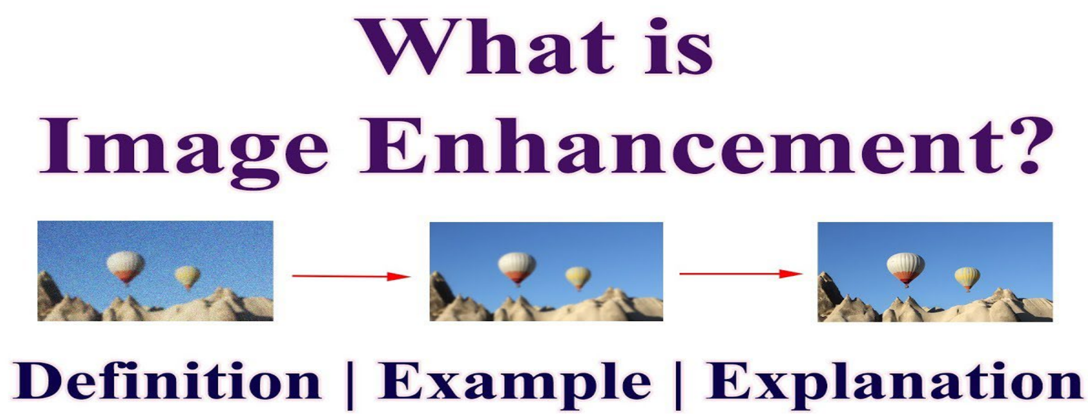
Dr. Ebtsam Adel

Associate professor at IS Dept.
Faculty of computers and information
Damanhour university

Lecture (02)

Chapter 3: IMAGE ENHANCEMENT



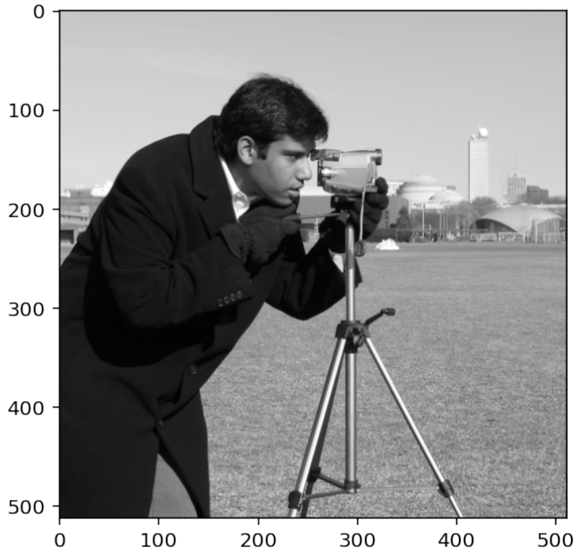


3.4 Fundamentals of Spatial Filtering: Page 144

Chapter 3: IMAGE ENHANCEMENT

This lecture will cover:

- ✓ Image enhancement in spatial
- ✓ Median filter
- ✓ Median filter with replicate border
- ✓ Sharpening Spatial Filters – Laplacian filters



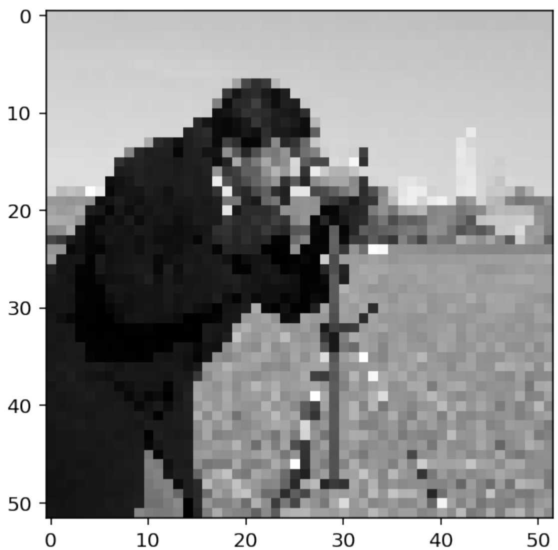


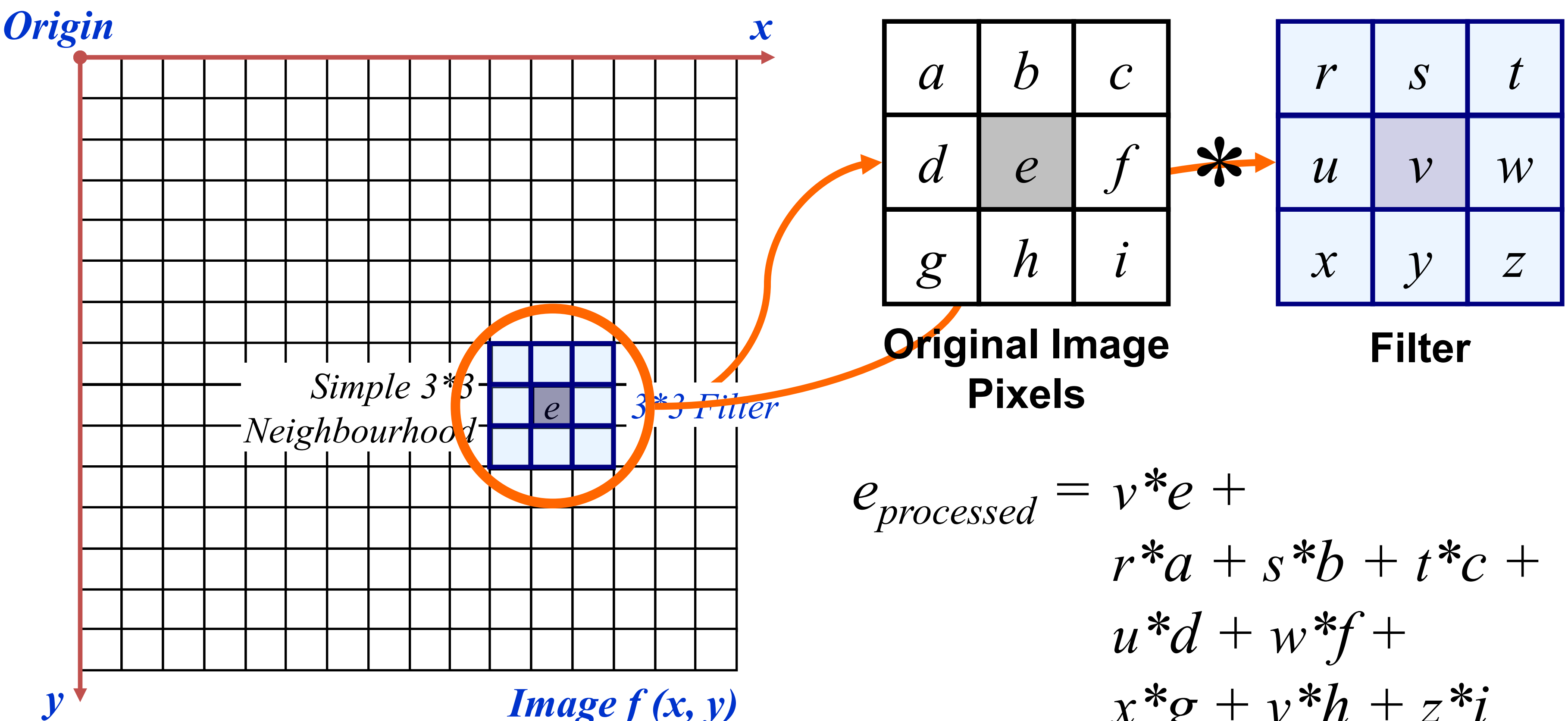
Image Filtering

- **Image filtering** is used to:
  - Remove noise
  - Sharpen contrast
  - Highlight contours
  - Detect edges
  - Other uses?

- By using spatial filters (also called spatial masks, kernels, templates, and windows).

- Image **filters** can be **classified** as **linear** or **nonlinear**.
- **Linear** filters are also know as **convolution filters** as they can be represented using a **matrix multiplication**.
- **Nonlinear** through an operation.

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.

Spatial Filtering

✓simply move the filter mask from point to point in an image.

Spatial filtering are filtering operations performed on the pixel intensities of an image and not on the frequency components of the image.

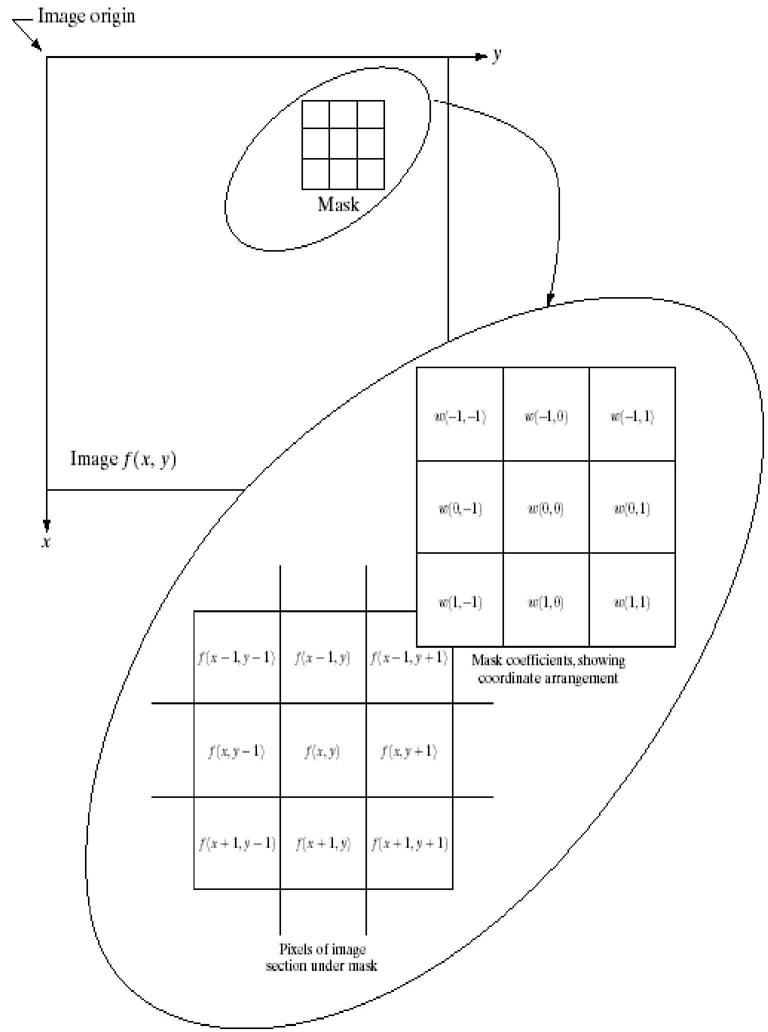


FIGURE 3.32 The mechanics of spatial filtering. The magnified drawing shows a 3×3 mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability.

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

Spatial Correlation and Convolution

- There are two closely related concepts that must be understood clearly when performing linear spatial filtering. One is *correlation* and the other is *convolution*.

- **Correlation** is the process of moving a filter mask over the image and computing the sum of products at each location.

- The mechanics of **convolution** are the same, except that the filter is first rotated by 180°.

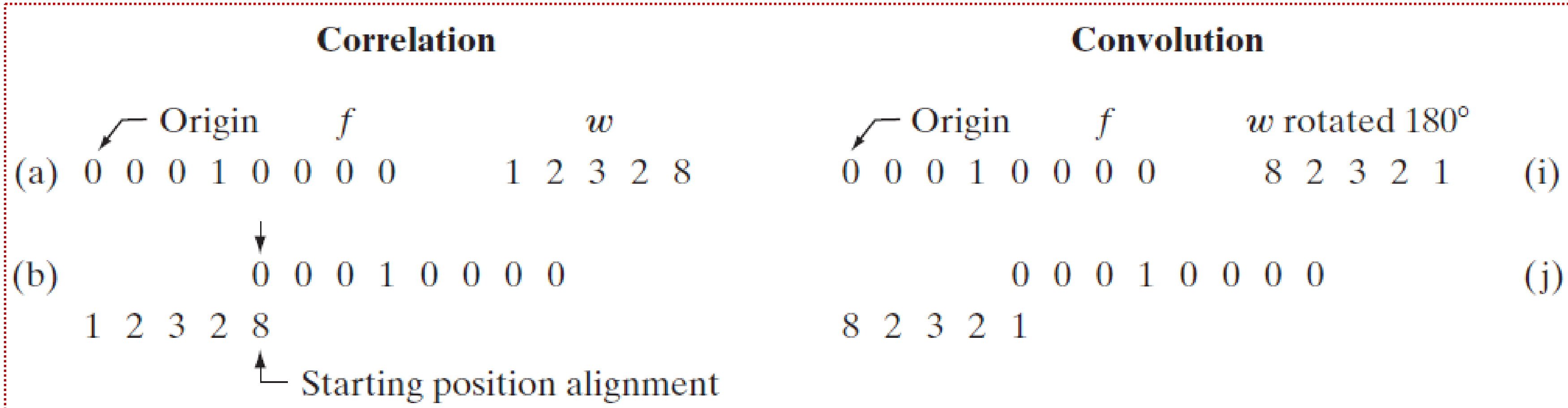


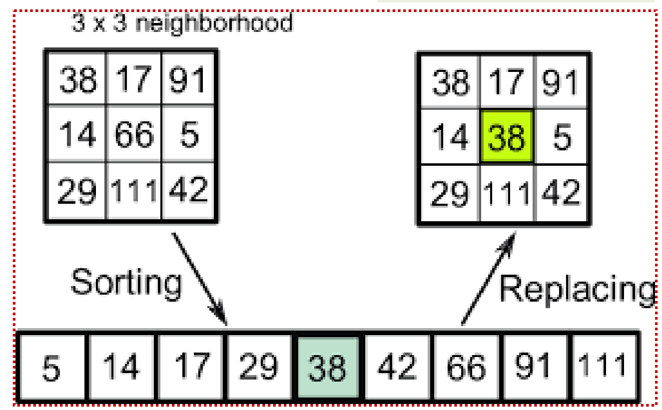
Image Enhancement in spatial domain- Spatial Filtering

1. **Smoothing** filter – low pass – reduction details.
2. **Sharpener** filter – high pass – enhance details.

Common Smoothing Filters

- Standard average (linear)
- Weighted average (linear)
- Median (order statistics)

Median filter



$$\begin{array}{|c|c|c|} \hline \omega_1 & \omega_2 & \omega_3 \\ \hline \omega_4 & \omega_5 & \omega_6 \\ \hline \omega_7 & \omega_8 & \omega_9 \\ \hline \end{array} \qquad \frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Smoothing Linear Filters

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

1: Diagonal

Standard average filter.

Blur edges.

weighted average filter

the center is the most important and other pixels are inversely weighted as a function of their distance from the center of the mask.

Smoothing Linear Filters

- The general implementation for filtering an M X N image with a weighted averaging filter of size (*m* and *n* odd) is given by the expression:

$$\frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

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

Image Enhancement in spatial domain

Linear Filtering

- In digital image processing, the process of applying a linear filter (also called convolution kernel) to a digital image is called linear filtering. The working of linear filtering is based on the concept of convolution.

Nonlinear Filtering

- The operation also consists of moving the filter mask from pixel to pixel in an image. The filtering operation is based conditionally on the values of the pixels in the neighborhood, and they do not explicitly use coefficients in the sum-of-products manner.

- Min, max, median filters.

Image Enhancement in spatial domain

Apply the **standard average filter** on the following sub image:

200	100	60	30
10	70	20	20
10	80	90	120
40	60	50	40
40	30	20	10

Standard Average filter

$$\frac{1}{9}\times\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

Solution

Pixel 1=1/9(200+100+10+70)=42

Pixel 2=1/9(200+100+60+10+70+20)=51

Pixel 3=1/9(100+60+30+70+20+20)=33

Pixel 4=1/9(60+30+20+20)=14

....

....

The output image

42	51	33	14
52	71	66	38
30	48	61	38
29	47	56	38
19	27	23	13

Image Enhancement in spatial domain

Example: Use the following 3×3mask to perform the convolution process on the shaded pixels in the 5×5 image below. Write the filtered image.

0	1/6	0
1/6	1/3	1/6
0	1/6	0

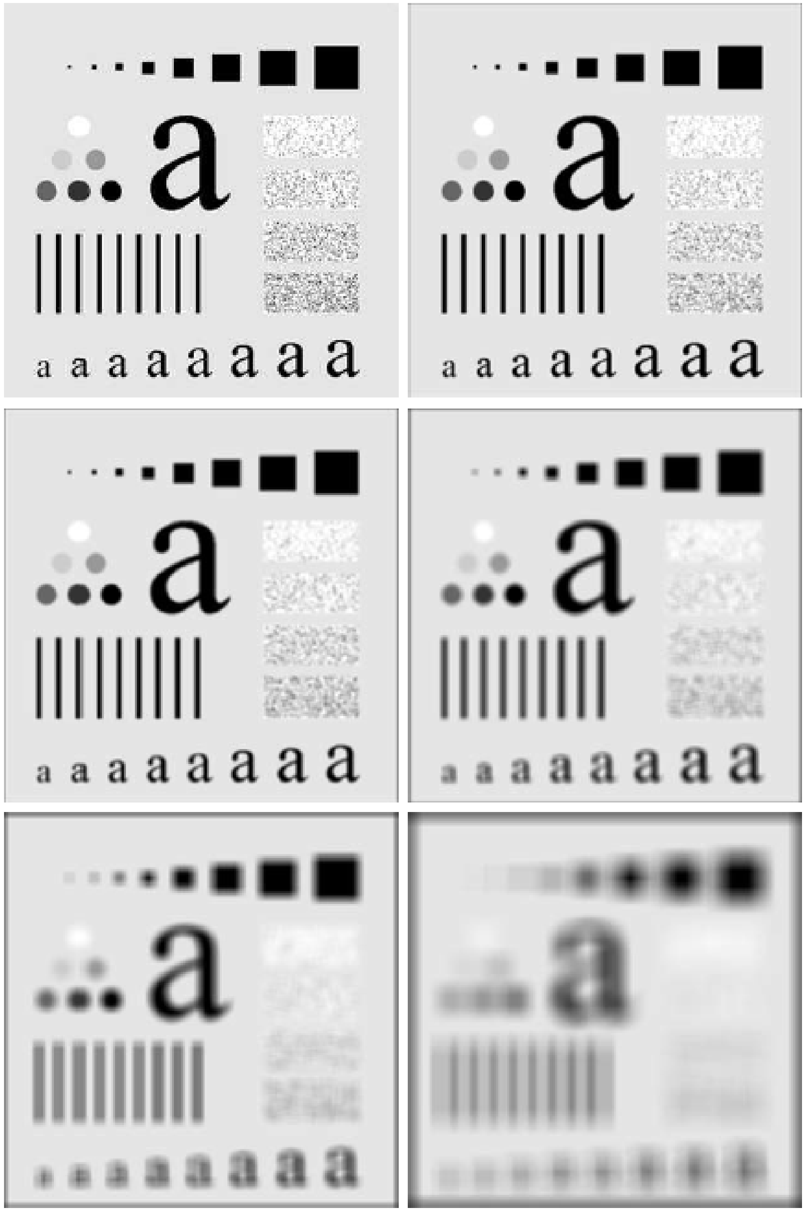
3×3 mask

30	40	50	70	90
40	50	80	60	100
35	255	70	0	120
30	45	80	100	130
40	50	90	125	140

5×5 image

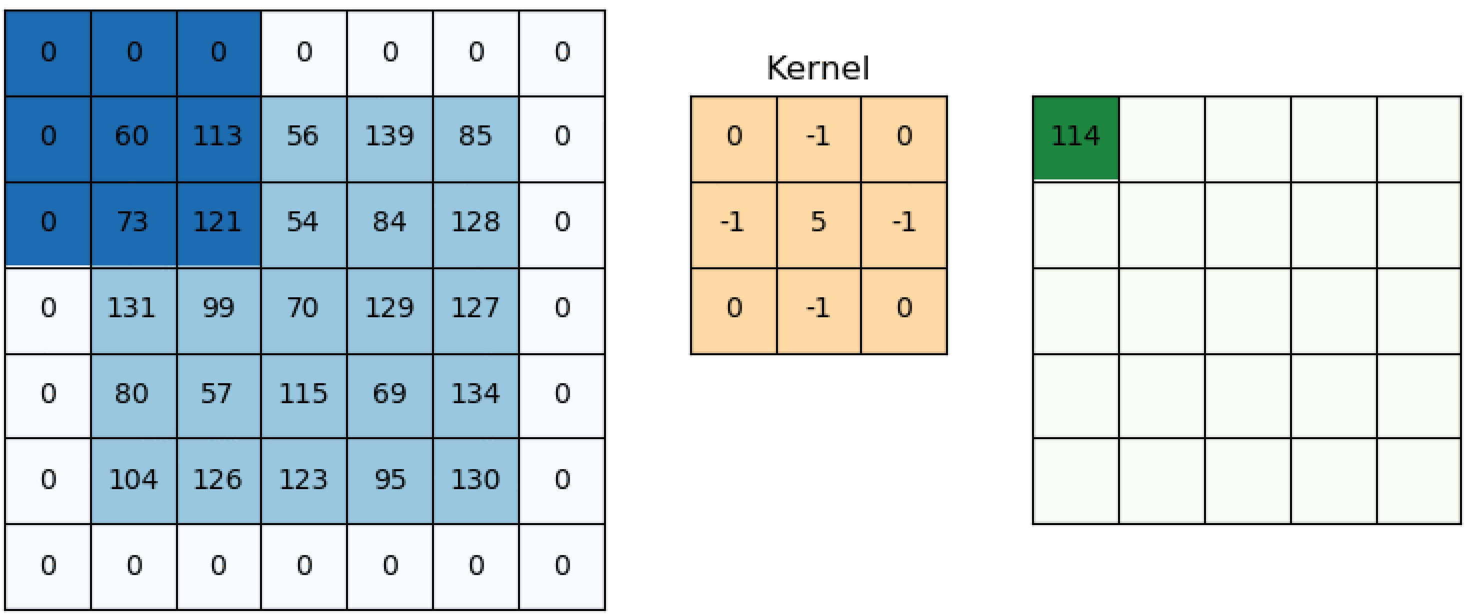
Image Enhancement in spatial domain

FIGURE 3.33 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $m = 3, 5, 9, 15$, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their intensity levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50 × 120 pixels.



Blurring + reducing noise

Zero padding in image filtering



- Zero padding
- Replicate border

“Order-Statistics Filters ”

Median filter

Original Image with Median Filter

Order-Statistics Filters - Median filtering

- **A median filter** is a type of nonlinear filter used in computer vision that replaces the value of each pixel in an image with the median value of the neighboring pixels.

- It is a simple and effective technique for reducing noise in images and can also be used for image smoothing and preserving edges.

Order-Statistics Filters - Median filtering

- Median filtering is a nonlinear method used to remove noise from images.
- It is widely used as it is very effective at removing noise while preserving edges.
- Very effective for removing “salt and pepper” noise (i.e., random occurrences of black and white pixels).

averaging | median filtering

Order-Statistics Filters (nonlinear Filters)

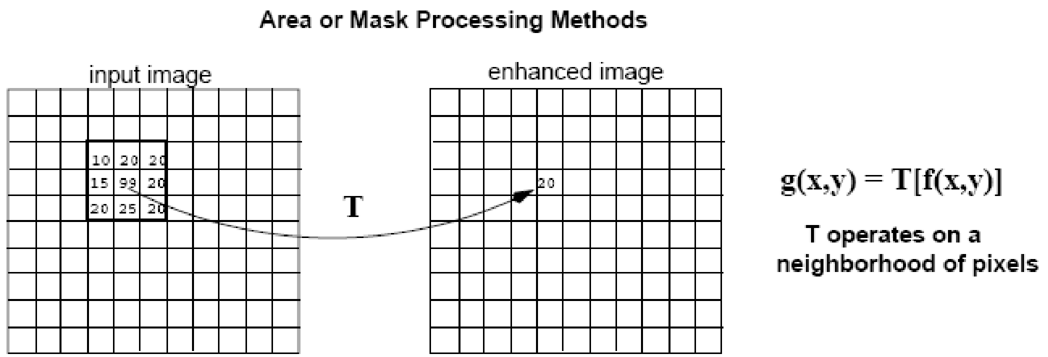
Median filtering

- Used primarily for **noise reduction without any changes in edge of objects:**
  - The gray level of each pixel is replaced by the median of the gray levels in the neighborhood of that pixel (instead of by the average as before).
  - quite popular because for certain types of random noise (impulse noise, salt and pepper noise) , they provide excellent noise-reduction capabilities, with considering less blurring than linear smoothing filters of similar size.

Smoothing Filters: Median Filtering (cont'd)

Steps:

1. Sort the pixels in ascending order.
2. Replace the original pixel value by the median.



10- 20- 20- 15- 99- 20- 20- 25- 20

Sorting

10- 15- 20- 20- **20**- 20-20- 25- 99

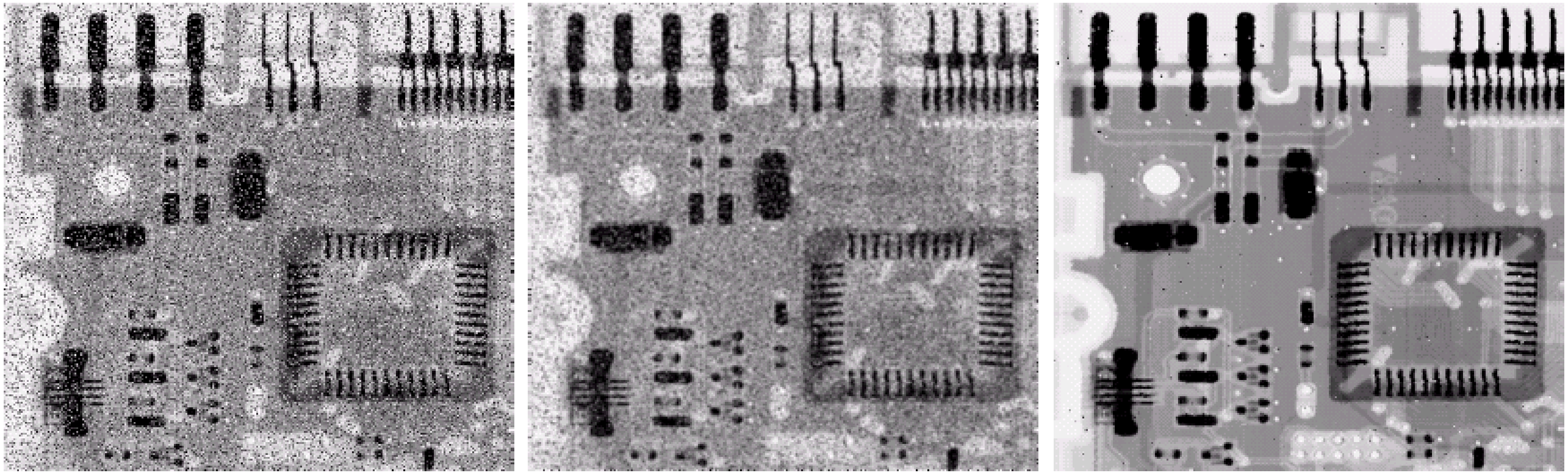
Smoothing Filters: Median Filtering (cont'd)

- Replace each pixel by the median in a neighborhood around the pixel.
- The size of the neighborhood controls the amount of smoothing.

- **Disadvantages**: small details are erased as very thin lines and sharp corners are damaged.
- Time consuming.

Smoothing Filters: Median Filtering (cont'd)

- forces the points with distinct gray levels to be more like their neighbors.



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.)

Example on median filter

Example on median filter

Apply 3 x 3 median filter on the following image.

Salt noise

150	151	155	150	151
152	255	153	150	152
153	154	155	0	153
157	158	159	157	155

Pepper noise

Example on median filter with replicate border

150	150	151			
150	**150**	**151**	**155**	**150**	**151**
152	152	255	153	150	152
	153	154	155	0	153
	157	158	159	157	155

151				

solution

With replicate border

{150,150,151,150,150,151,152,152,255}

After sorting

{150,150,150,150,**151**,151,152,152,255}

The median value is 151

Example on median filter with replicate border

150	151	155		
150	151	155	150	151
152	255	153	150	152
153	154	155	0	153
157	158	159	157	155

151	152			

With replicate border

{150,151,155,150,151,155,152,255,153}

After sorting

{150,150,151,151,152,153,155,155,255}

The median value is 152

Example on median filter with replicate border

	151	155	150	
150	151	155	150	151
152	255	153	150	152
153	154	155	0	153
157	158	159	157	155

151	152	151		

With replicate border

{151,155,150,151,155,150,255,153,150}

After sorting

{150,150,150,151,151,153,155,155,255}

The median value is 151

Example on median filter with replicate border

150	151	155	150	151
152	255	153	150	152
153	154	155	0	153
157	158	159	157	155

After applying 3*3 median filter

151	152	151	151	151
152	152	153	152	151
154	155	155	153	153
157	157	157	155	155

- As we note, the salt and pepper noise is removed completely from the image without blurring effect.

Sharpening Spatial Filters

Sharpening Spatial Filters

- Previously we have looked at **smoothing** filters which remove small detail.
- **Sharpening** spatial filters seek to highlight fine detail.
  - Remove blurring from images
  - Highlight edges

- **Image sharpening** is an effect applied to digital images to give them a sharper appearance. Sharpening enhances the definition of edges in an image.

- Sharpening filters are based on spatial differentiation.

Laplacian Filter

- It is a second-order derivative operator/filter/mask.
- It detects the image along with horizontal and vertical directions collectively.

Sharpening Spatial Filters

f

50	60	90	100
50	60	90	100
50	60	90	100
50	60	90	100

$\nabla^2 f$

-10	-20	20	10
-10	-20	20	10
-10	-20	20	10
-10	-20	20	10

$g = f + \nabla^2 f$

40	40	110	110
40	40	110	110
40	40	110	110
40	40	110	110

$$g(x, y) = f(x, y) + c\left[\nabla^2 f(x, y)\right] \qquad (3.6\text{-}7)$$

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.

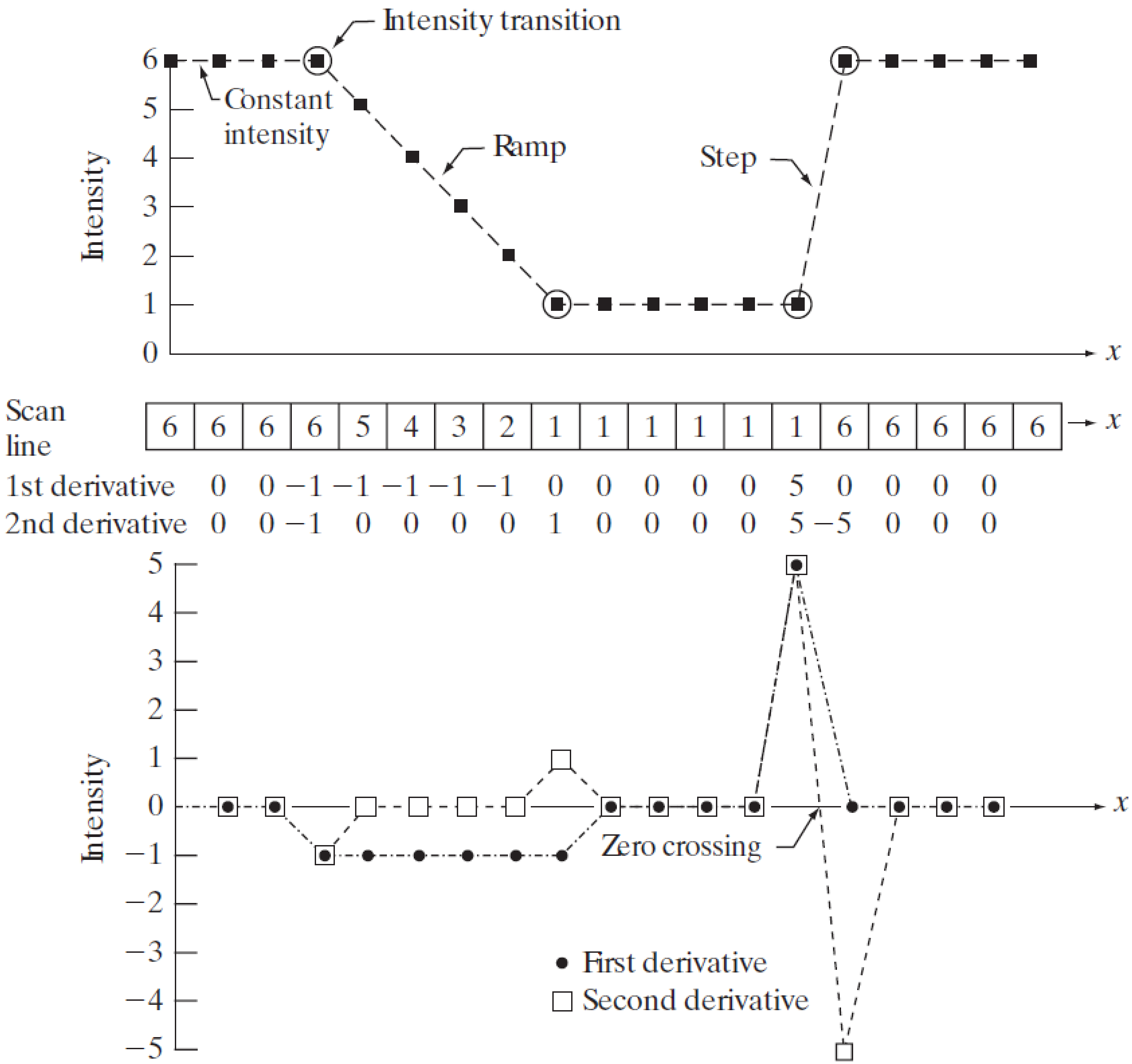
2^{nd} Derivative

The formula for **the** 2^{nd} 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.

Sharpening Spatial Filters



a
b
c

FIGURE 3.36 Illustration of the first and second derivatives of a 1-D digital function representing a section of a horizontal intensity profile from an image. In (a) and (c) data points are joined by dashed lines as a visualization aid.

2nd Derivative (cont.)

Original image	50	50	60	90	100	100	100
1st Derivative	0	10	30	10	0	0	

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

2nd Derivative		10	20	-20	-10	0

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

Using Second Derivatives For Image Enhancement

- The 2nd derivative is more useful for image enhancement than the 1st derivative
  - Stronger response to fine “small” detail
  - Simpler implementation.

The Laplacian- 2 D

The Laplacian is defined as follows:

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

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

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

Horizontally

and in the *y* direction as follows:

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

vertically

Using Second Derivatives For Image Enhancement

Exercise: Apply the second derivative on the shaded pixel.

20	20	20	20
20	10	10	10
20	10	10	10
20	10	10	10

$$\nabla^2 f(P1) = 20 + 20 + 10 + 10 - 4*10 = 20$$

$$\nabla^2 f(P2) = 10 + 10 + 10 + 10 - 4*10 = 0$$

The Laplacian

The Laplacian

Original
Image

-

Laplacian
Filtered Image

=

Sharpened
Image

The Laplacian

So, the Laplacian can be given as follows:

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

We can easily build a filter based on this

0	1	0
1	-4	1
0	1	0

Spatial filters : Sharpening
LAPLACE – 2st derivative

Example: apply the following Laplace on the highlighted pixel.

153	157	156	153	155
159	156	158	156	159
155	158	**<u>154</u>**	156	160
154	157	158	160	160
157	157	157	156	155

-158-156-158-158+4*154 = -14

So the value after filter = -14

We call the resultant image: **sharpened image.**

Filtered image=original +sharpened image

The value in the filter image=154+(-14) =140

0	−1	0
−1	4	−1
0	−1	0

Variants On
The Simple Laplacian

Variants On The Simple Laplacian

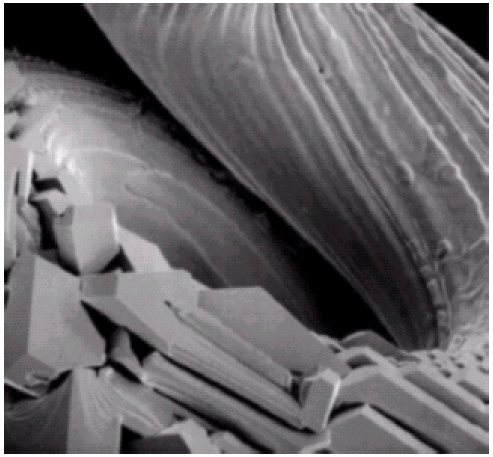
- There are lots of slightly different versions of the Laplacian that can be used:

0	1	0
1	-4	1
0	1	0

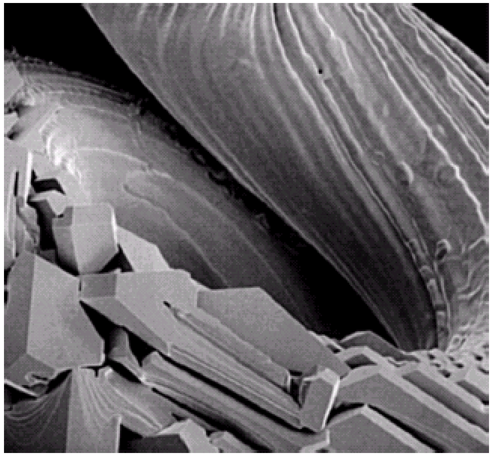
Simple Laplacian

1	1	1
1	-8	1
1	1	1

Variant of Laplacian



-1	-1	-1
-1	9	-1
-1	-1	-1



Variants On The Simple Laplacian

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 3.37
(a) Filter mask used to implement Eq. (3.6-6). (b) Mask used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other implementations of the Laplacian found frequently in practice.

Simplified Image Enhancement

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

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

Simplified Image Enhancement

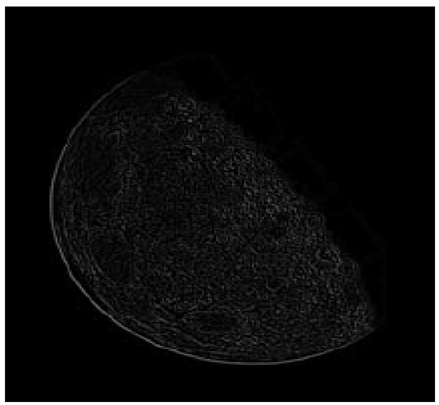
- it is important to keep in mind which definition of the Laplacian is used. If the definition used has a negative center coefficient, then we *subtract*, rather than add, the Laplacian image to obtain a sharpened result.

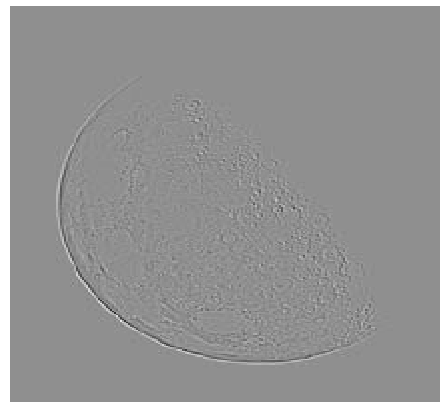
$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) & \text{if the center coefficient of the Laplacian mask is negative} \\ f(x,y) + \nabla^2 f(x,y) & \text{if the center coefficient of the Laplacian mask is positive.} \end{cases}$$

The Laplacian (cont.)

Original image

Laplacian





Scaled image

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

Sharpened image

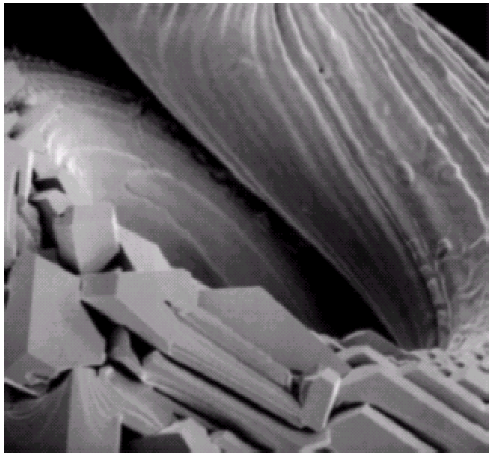
Laplacian Spatial Filtering in one step

Sharpening can be done in 1 pass:

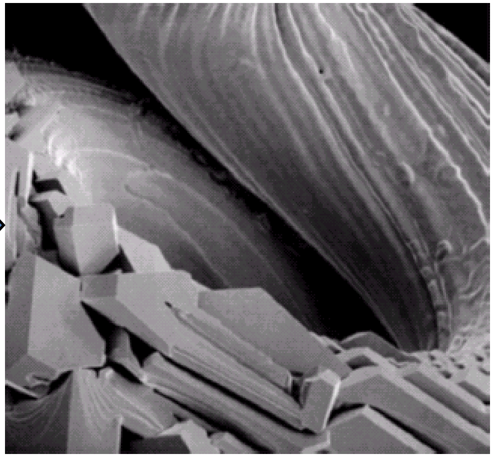
$$\underset{\text{LAPLACIAN}}{\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}} + \underset{\text{Original Image}}{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}} = \underset{\text{Sharpened Image}}{\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}}$$

Simplified Image Enhancement (cont…)

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



Composite Laplacian filter- better implementation

$$g(x, y) = f(x, y) + \nabla^2 f(x, y)$$ if the center coefficient of the Laplacian mask is positive.

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

$$g(x,y) = f(x,y) + \nabla^2 f(x,y$$

$$g(x,y) = f(x,y) + 4f(x,y) - f(x+1,y) - f(x-1,y) - f(x,y+1) - f(x,y-1)$$

$$g(x,y) = 5f(x,y) - f(x+1,y) - f(x-1,y) - f(x,y+1) - f(x,y-1)$$

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

Example

Laplacian vs. Composite Laplacian

Example: apply the Laplacian and composite Laplacian filters on the following blurred edge.

f

50	60	90	100
50	60	90	100
50	60	90	100
50	60	90	100

Laplacian

0	-1	0
-1	4	-1
0	-1	0

$\nabla^2 f$

-10	-20	20	10
-10	-20	20	10
-10	-20	20	10
-10	-20	20	10

$g = f + \nabla^2 f$

40	40	110	110
40	40	110	110
40	40	110	110
40	40	110	110

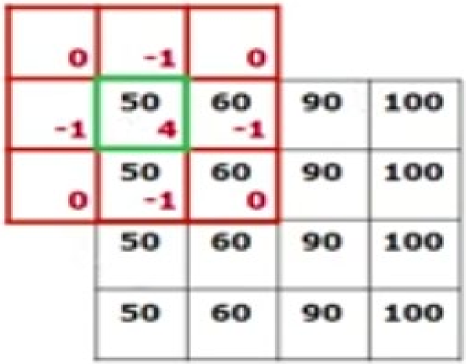
50	60	90	100
50	60	90	100
50	60	90	100
50	60	90	100

Composite Laplacian

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

40	40	110	110
40	40	110	110
40	40	110	110
40	40	110	110

Laplacian vs. Composite Laplacian



$\nabla^2 f$

-10			

$g = f + \nabla^2 f$

40			

$$\nabla^2 f = 4 * 50 - 50 - 50 - 60 - 50$$
$$= 200 - 210$$
$$= -10$$

$$g(x,y) = 50 - 10$$

$$g(x,y) = 40$$

0	**-1**	**0**
-1	4	-1
0	-1	0

Laplacian vs. Composite Laplacian

0	-1	0	
50 -1	60 4	90 -1	100
50 0	60 -1	90 0	100
50	60	90	100
50	60	90	100

$\nabla^2 f$

-10	-20		

$g = f + \nabla^2 f$

40	40		

$$\nabla^2 f = 4 * 60 - 50 - 60 - 60 - 90$$
$$= 240 - 260$$
$$= -20$$

$$g(x,y) = 60 - 20$$

$$g(x,y) = 40$$

Laplacian vs. Composite Laplacian

	0	-1	0
50	60 -1	90 4	100 -1
50	60 0	90 -1	100 0
50	60	90	100
50	60	90	100

$\nabla^2 f$

-10	-20	20	

$g = f + \nabla^2 f$

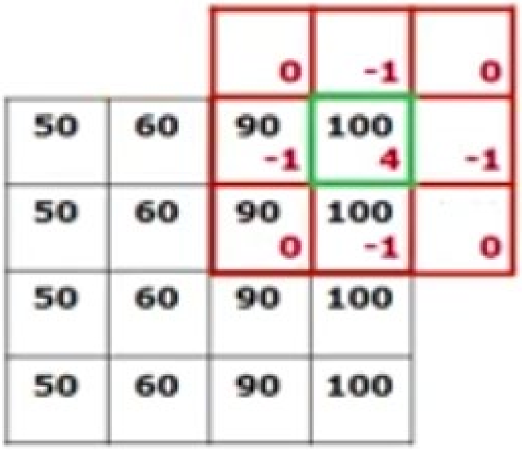
40	40	110	

$$\nabla^2 f = 4 * 90 - 90 - 90 - 60 - 100$$
$$= 360 - 340$$
$$= 20$$

$$g(x, y) = 90 + 20$$

$$g(x, y) = 110$$

Laplacian vs. Composite Laplacian



$\nabla^2 f$

-10	-20	20	10

$g = f + \nabla^2 f$

40	40	110	110

$$\nabla^2 f = 4 * 100 - 100 - 100 - 100 - 90$$
$$= 400 - 390$$
$$= 10$$

$$g(x, y) = 100 + 10$$

$$g(x, y) = 110$$

Laplacian vs. Composite Laplacian

50	60	90	100	
50	60	90	100	
50	60	90 0	100 -1	0
50	60	90 -1	100 4	-1
		0	-1	0

$\nabla^2 f$

-10	-20	20	10
-10	-20	20	10
-10	-20	20	10
-10	-20	20	10

$g = f + \nabla^2 f$

40	40	110	110
40	40	110	110
40	40	110	110
40	40	110	110

$$\nabla^2 f = 4 * 100 - 100 - 100 - 100 - 90$$
$$= 400 - 390$$
$$= 10$$

$$g(x, y) = 100 + 10$$
$$g(x, y) = 110$$

Laplacian vs. Composite Laplacian

g

40	40	110	110
40	40	110	110
40	40	110	110
40	40	110	110

$$
\begin{aligned}
g(x,y) &= 5 * 100 - 100 - 100 - 100 - 90 \\
&= 500 - 390 \\
&= 110
\end{aligned}
$$

Composite Laplacian

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

PRACTICAL PART

Practical part

- Implement 2D Convolution Image Filtering in python.
- Implement Image Smoothing (Averaging filter) in python.
- Implement Image Smoothing (Median Filtering) in python.
- Implement Laplacian filter in spatial domain using python.


Thank You !
Any questions?