Digital Image Processing and Pattern Recognition

E1528

Fall 2022-2023

Lecture 6

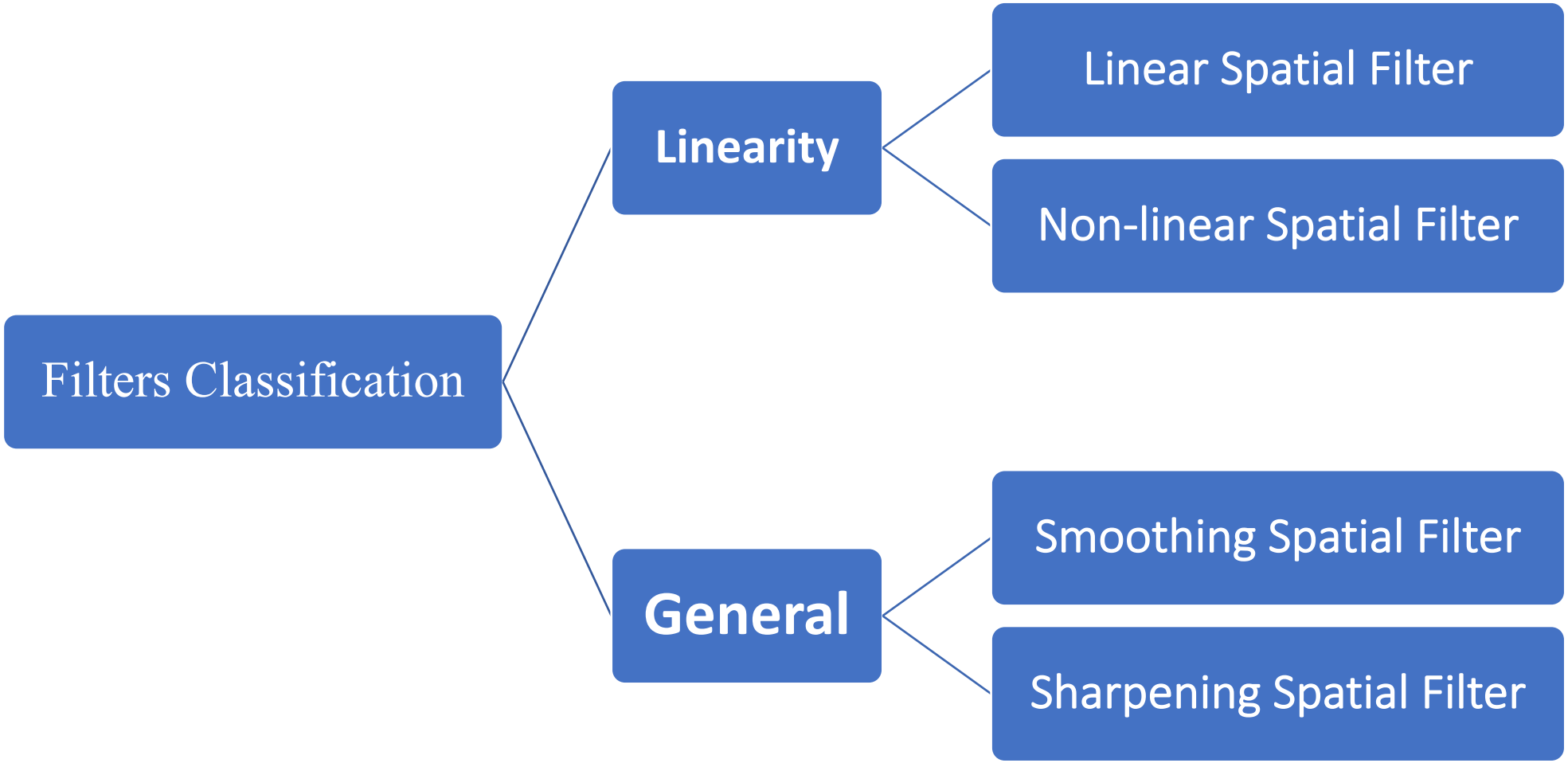
Filters Classification

INSTRUCTOR

DR / AYMAN SOLIMAN

Contents

- Filters Classification
- Non-linear Spatial Filter
- Smoothing spatial filters
- Averaging linear filters
- Order-statistics non-linear filter
- Median filters
- Minimum Filter
- Minimum Filter
- Maximum Filter


Filters Classification
Linearity
Linear Spatial Filter
Non-linear Spatial Filter
General
Smoothing Spatial Filter
Sharpening Spatial Filter

➢ Non-linear Spatial Filter

- 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.
- For example, noise reduction can be achieved effectively with a nonlinear filter whose basic function is to compute the median gray-level value in the neighborhood in which the filter is located. Computation of the median is a nonlinear operation.

➢ Example 1

➢ Use the following 3*3 mask to perform the convolution process on the shaded pixels in the 5*5 image below. Write the filtering 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

➢ Example 1 solution

$$0 \times 30 + \frac{1}{6} \times 40 + 0 \times 50 + \frac{1}{6} \times 40 + \frac{1}{3} \times 50 + \frac{1}{6} \times 80 + 0 \times 35 + \frac{1}{6} \times 255 + 0 \times 70 = 85$$

$$0 \times 40 + \frac{1}{6} \times 50 + 0 \times 70 + \frac{1}{6} \times 50 + \frac{1}{3} \times 80 + \frac{1}{6} \times 60 + 0 \times 255 + \frac{1}{6} \times 70 + 0 \times 0 = 65$$

$$0 \times 50 + \frac{1}{6} \times 70 + 0 \times 90 + \frac{1}{6} \times 80 + \frac{1}{3} \times 60 + \frac{1}{6} \times 100 + 0 \times 70 + \frac{1}{6} \times 0 + 0 \times 120 =$$

$$0 \times 40 + \frac{1}{6} \times 50 + 0 \times 80 + \frac{1}{6} \times 35 + \frac{1}{3} \times 255 + \frac{1}{6} \times 70 + 0 \times 30 + \frac{1}{6} \times 45 + 0 \times 80 = 118$$

and so on …

Filtered image =

30	40	50	70	90
40	**85**	**65**	**61**	100
35	**118**	**92**	**58**	120
30	**84**	**77**	**89**	130
40	50	90	125	140

➢ Smoothing spatial filters

- Also called low pass filter.

- They include:

1. Averaging linear filters
2. Order-statistics non-linear filter

➢ Smoothing spatial filters

- Smoothing Spatial Filters are used for blurring and for noise reduction.

- Blurring is used in preprocessing steps to:

  1. remove small details from an image prior to (large) object Extraction

  2. bridge small gaps in lines or curves.

- Noise reduction can be accomplished by blurring with a linear filter and also by nonlinear filtering.

➢ Averaging linear filters

- The response of averaging filter is simply the average of the pixels contained in the neighborhood of the filter mask.
- The output of averaging filters is a smoothed image with reduced "sharp" transitions in gray levels.
- Noise and edges consist of sharp transitions in gray levels. Thus smoothing filters are used for noise reduction; however, they have the undesirable side effect that they blur edges.

➢ Averaging linear filters

- The figure below shows two 3*3 averaging filters

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

Standard average filter

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

Weighted average filter

- Weighted average filter has different coefficients to give more importance (weight) to some pixels at the expense of others. The idea behind that is to reduce blurring in the smoothing process.

➢ Averaging linear filters

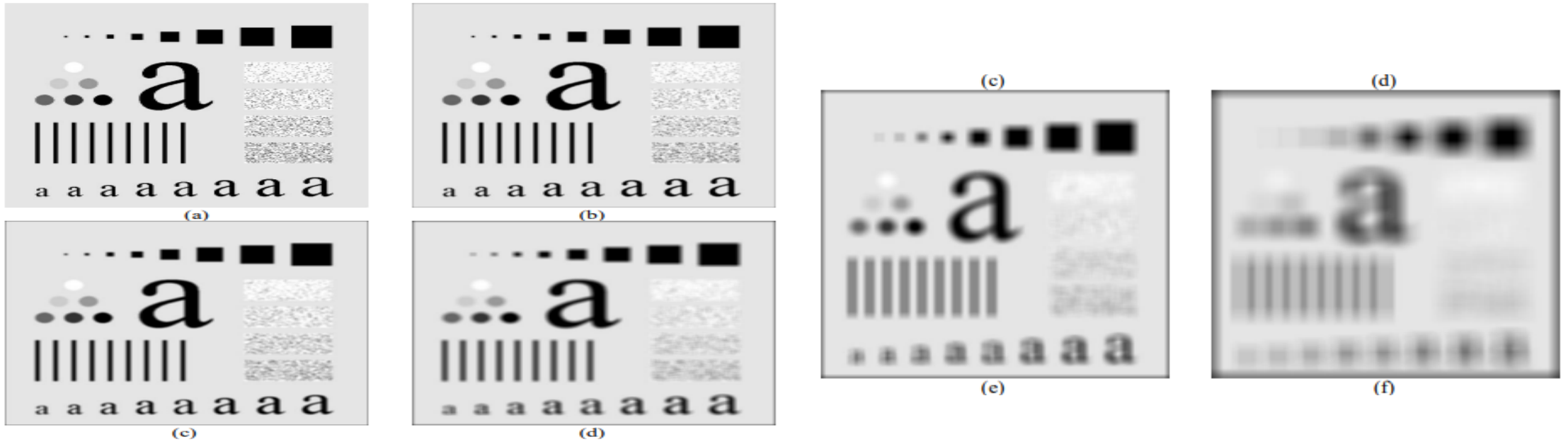
- Averaging linear filtering of an image f of size M*N with a filter mask of size m*n is given by the expression:

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

- To generate a complete filtered image this equation must be applied for x = 0, 1, 2, …, M-1 and y = 0, 1, 2, …, N-1.

➢ Example 2

➢ Figure below shows an example of applying the standard averaging filter.



(a) Original image. (b)-(f) results of smoothing with square averaging filter masks of size

n= 3,5,9,15 and 35 respectively

➢ Example 2 comments

➢ As shown in the figure, the effects of averaging linear filter are:

1. Blurring which is increased whenever the mask size increases.

2. Blending (removing) small objects with the background. The size of the mask establishes the relative size of the blended objects.

3. Black border because of padding the borders of the original image.

4. Reduced image quality.

➢ Order-statistics non-linear filter

- are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the neighborhood, and then replacing the value of the center pixel with the value determined by the ranking result.
- Examples include Max, Min, and Median filters.

➢ Median filters

- Median filter it replaces the value at the center by the median pixel value in the neighborhood, (i.e., the middle element after they are sorted). Median filters are particularly useful in removing impulse noise (also known as salt-and-pepper noise). Salt = 255, pepper = 0 gray levels.
- In a 3x3 neighborhood the median is the 5th largest value, in a 5x5 neighborhood the 13th largest value, and so on.
- For example, suppose that a 3x3 neighborhood has gray levels (10, 20, 0, 20, 255, 20, 20, 25, 15). These values are sorted as (0,10,15,20,20,20,20,25,255), which results in a median of 20 that replaces the original pixel value 255 (salt noise).

➢ Example 3

Consider the following 5×5 image:

20	30	50	80	100
30	20	80	100	110
25	255	70	0	120
30	30	80	100	130
40	50	90	125	140

Apply a 3×3 median filter on the shaded pixels, and write the filtered image.

➢ Example 3 solution

Solution

20	30	50	80	100
30	20	80	100	110
25	255	70	0	120
30	30	80	100	130
40	50	90	125	140

Sort:
20, 25, 30, 30, **30**, 70, 80, 80, 255

20	30	50	80	100
30	20	80	100	110
25	255	70	0	120
30	30	80	100	130
40	50	90	125	140

Sort
0, 20, 30, 70, **80**, 80, 100, 100, 255

20	30	50	80	100
30	20	80	100	110
25	255	70	0	120
30	30	80	100	130
40	50	90	125	140

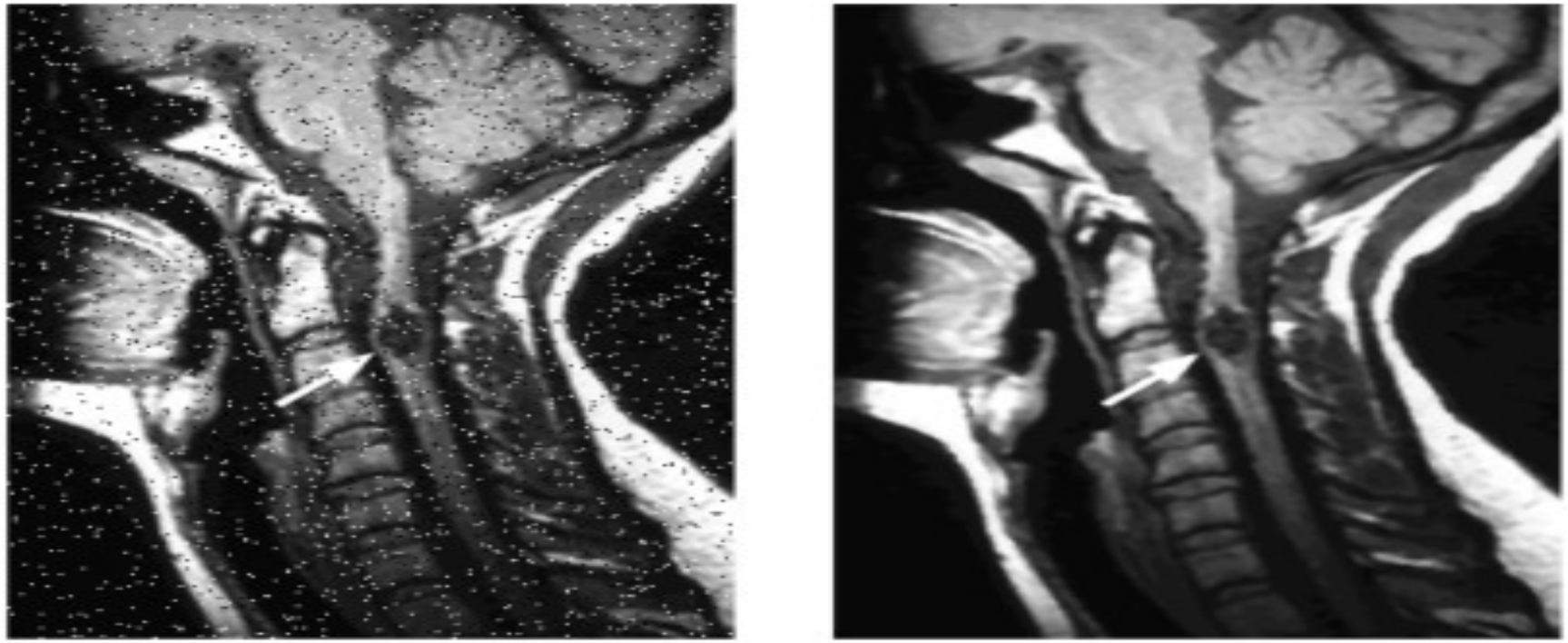
Sort
0, 70, 80, 80, **100**, 100, 110, 120, 130

Filtered Image =

20	30	50	80	100
30	20	80	100	110
25	**30**	**80**	**100**	120
30	30	80	100	130
40	50	90	125	140

➢ Example 4

➢ Figure below shows an example of applying the median filter



(A) (B)

(A) Noisy image (corrupted by “salt and pepper”). (B) The enhanced image using the 3 × 3 median filter.

➢ Median filters comments

- As shown in the figure, the effects of median filter are:

1. Noise reduction.

2. Less blurring than averaging linear filter.

Minimum Filter

- The o^{th} percentile filter is the min filter.
- Minimum filter selects the smallest value in the window and replace the center by the smallest value
- Using comparison, the minimum value can be obtained fast.(not necessary to sort)
- It enhances the dark areas of image

(mask size =3 x 3)

(mask size =7 x 7)

➢ Maximum Filter

- The maximum filter selects the largest value within of pixel values and replace the center by the largest value.
- Using comparison, the maximum value can be obtained fast.(not necessary to sort)
- Using the 100^{th} percentile results in the so-called max filter
- It enhances bright areas of image

mask (3 x 3)

mask (7 x 7)

➢ Sharpening spatial filters

- Sharpening aims to highlight fine details (e.g., edges) in an image or enhance detail that has been blurred through errors or imperfect capturing devices.
- Image blurring can be achieved using averaging filters, and hence sharpening can be achieved by operators that invert averaging operators.
- In mathematics, averaging is equivalent to the concept of integration, and differentiation inverts integration. Thus, sharpening spatial filters can be represented by partial derivatives.

➢ Sharpening spatial filters

Partial derivatives of digital functions

The first order partial derivatives of the digital image $f(x,y)$ are:

$$\frac{\partial f}{\partial x} = f(x+1,y) - f(x,y) \quad \text{and} \quad \frac{\partial f}{\partial y} = f(x,y+1) - f(x,y)$$

The first derivative must be:

1) zero along flat segments (i.e. constant gray values).
2) non-zero at the outset of gray level step or ramp (edges or noise)
3) non-zero along segments of continuing changes (i.e. ramps).

➢ Sharpening spatial filters

The second order partial derivatives of the digital image *f(x,y)* are:

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

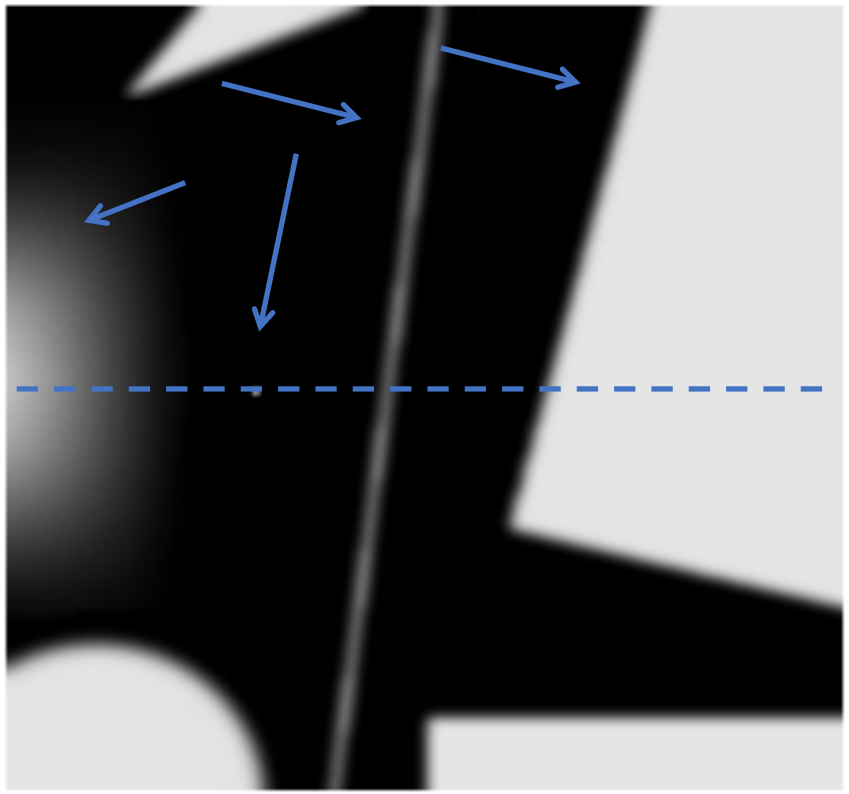
$$\frac{\partial^2 f}{\partial y^2} = f(x,y+1) + f(x,y-1) - 2f(x,y)$$

The second derivative must be:

1) zero along flat segments.
2) nonzero at the outset and end of a gray-level step or ramp;
3) zero along ramps

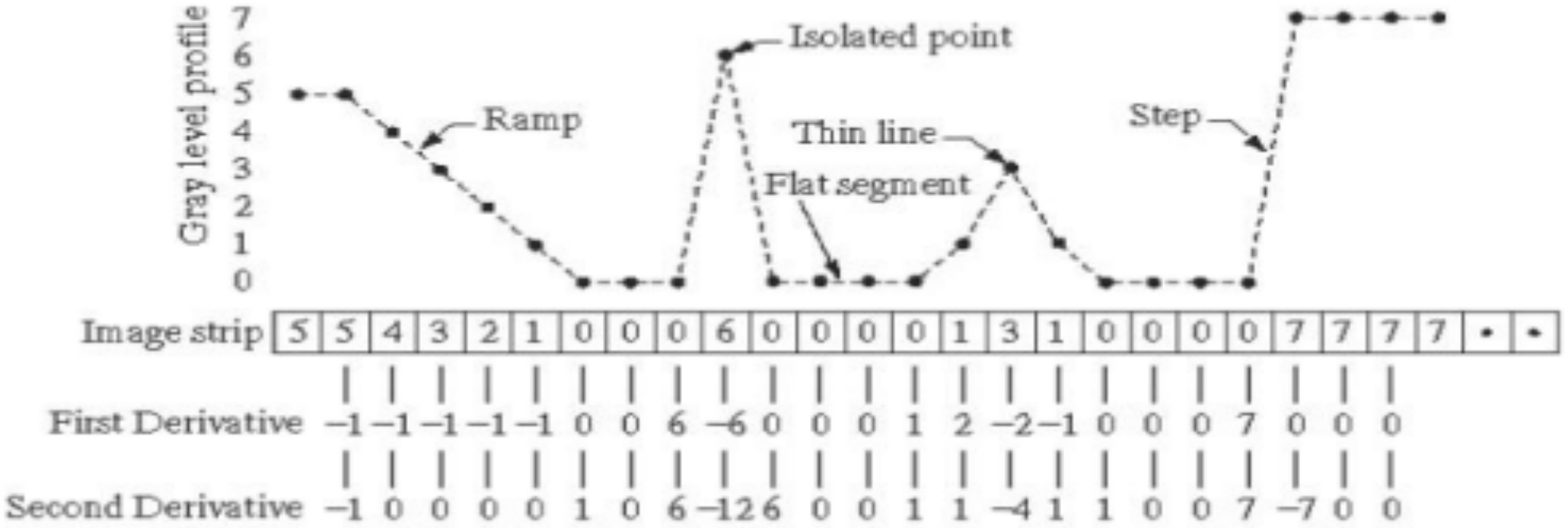
➢ Sharpening spatial filters

Consider the example below:



Horizontal intensity through center of the image

Example 5



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

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

➢ Comments

- We conclude that:

  1. 1st derivative detects thick edges while 2nd derivative detects thin edges.

  2. 2nd derivative has much stronger response at gray-level step than 1st derivative.

- Thus, we can expect a second-order derivative to enhance fine detail (thin lines, edges, including noise) much more than a first-order derivative.

➢ Code

Syntax

```
B = imsharpen(A)
B = imsharpen(A,Name,Value)
```

Description

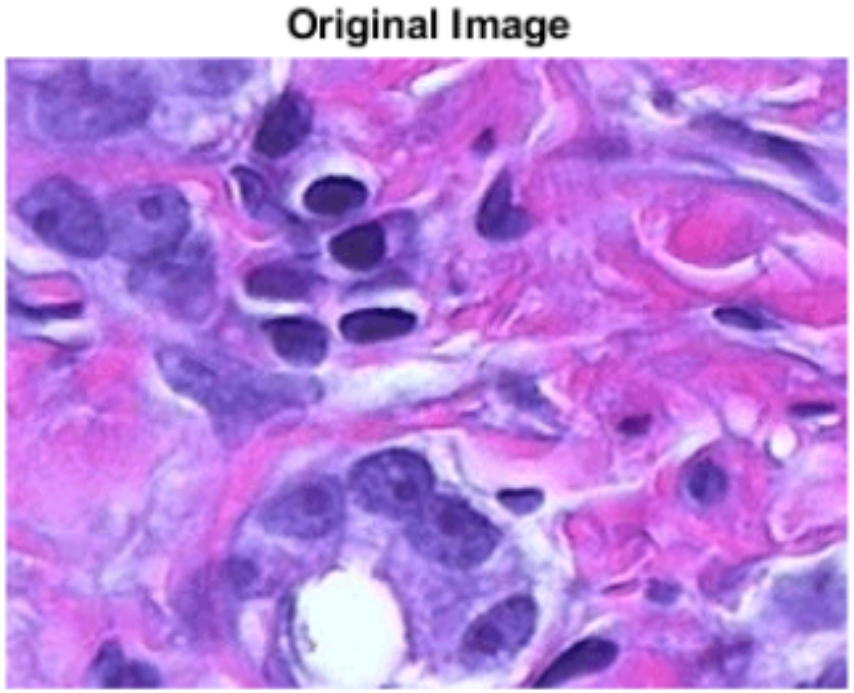
`B = imsharpen(A)` sharpens the grayscale or truecolor (RGB) image `A` by using the unsharp masking method.

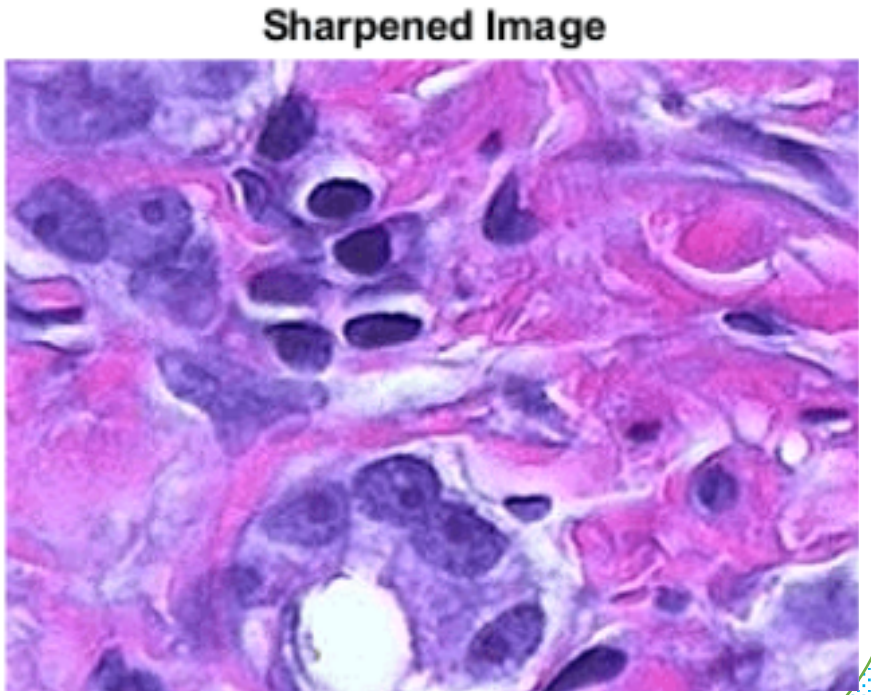
`B = imsharpen(A,Name,Value)` uses name-value arguments to control aspects of the unsharp masking.

➢ Code

```
a = imread('hestain.png');
imshow(a)
title('Original Image');
```

```
b = imsharpen(a);
figure, imshow(b)
title('Sharpened Image');
```

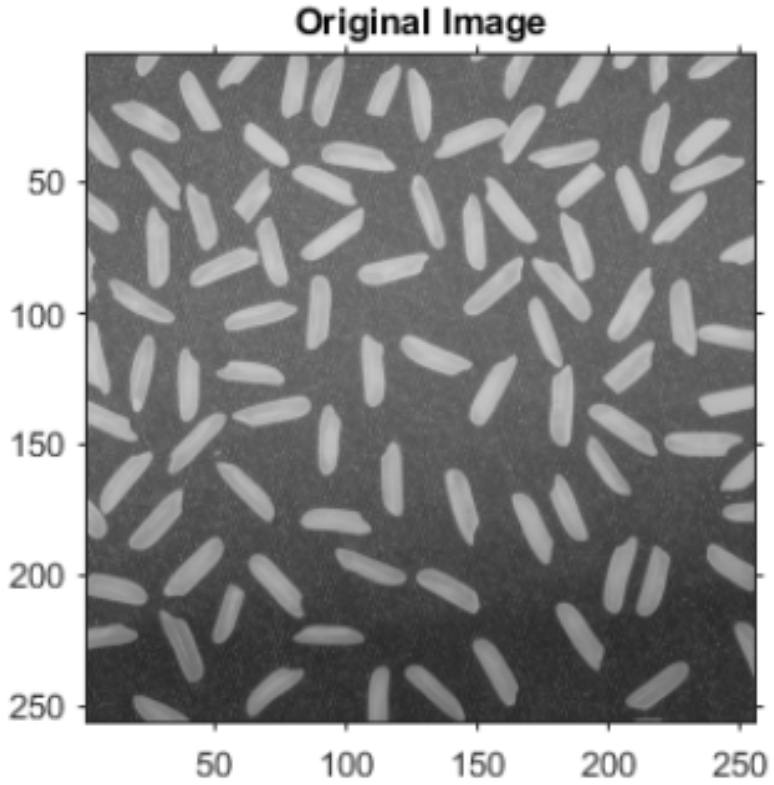




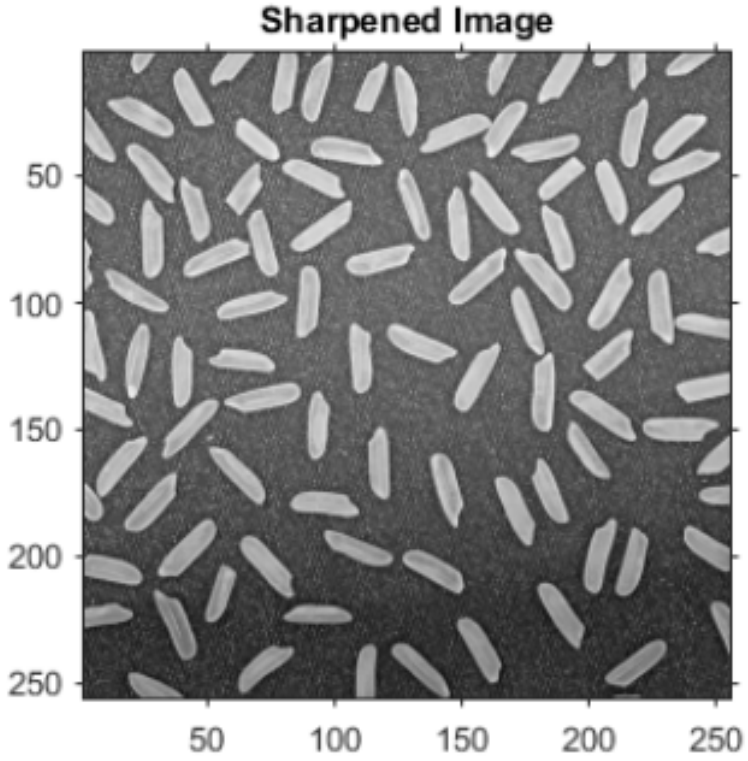
Sharpen the image using the ” imsharpen “ function and display it.

➢ Code

```
a = imread('rice.png');
imshow(a), title('Original Image');
```




```
b = imsharpen(a,'Radius',2,'Amount',1);
figure, imshow(b)
title('Sharpened Image');
```




Thank
you