# Spatial Filters

# Filtering

- It's a technique with which certain frequency components can be chosen or rejected.

- **Ex:** Low pass filters allows low frequencies and suppresses High frequency components.

- Filters in image processing can be categorized into following three categories:

  1. Convolution-based filters.
  2. Order-statistics(rank) filters.
  3. Hybrid filters.

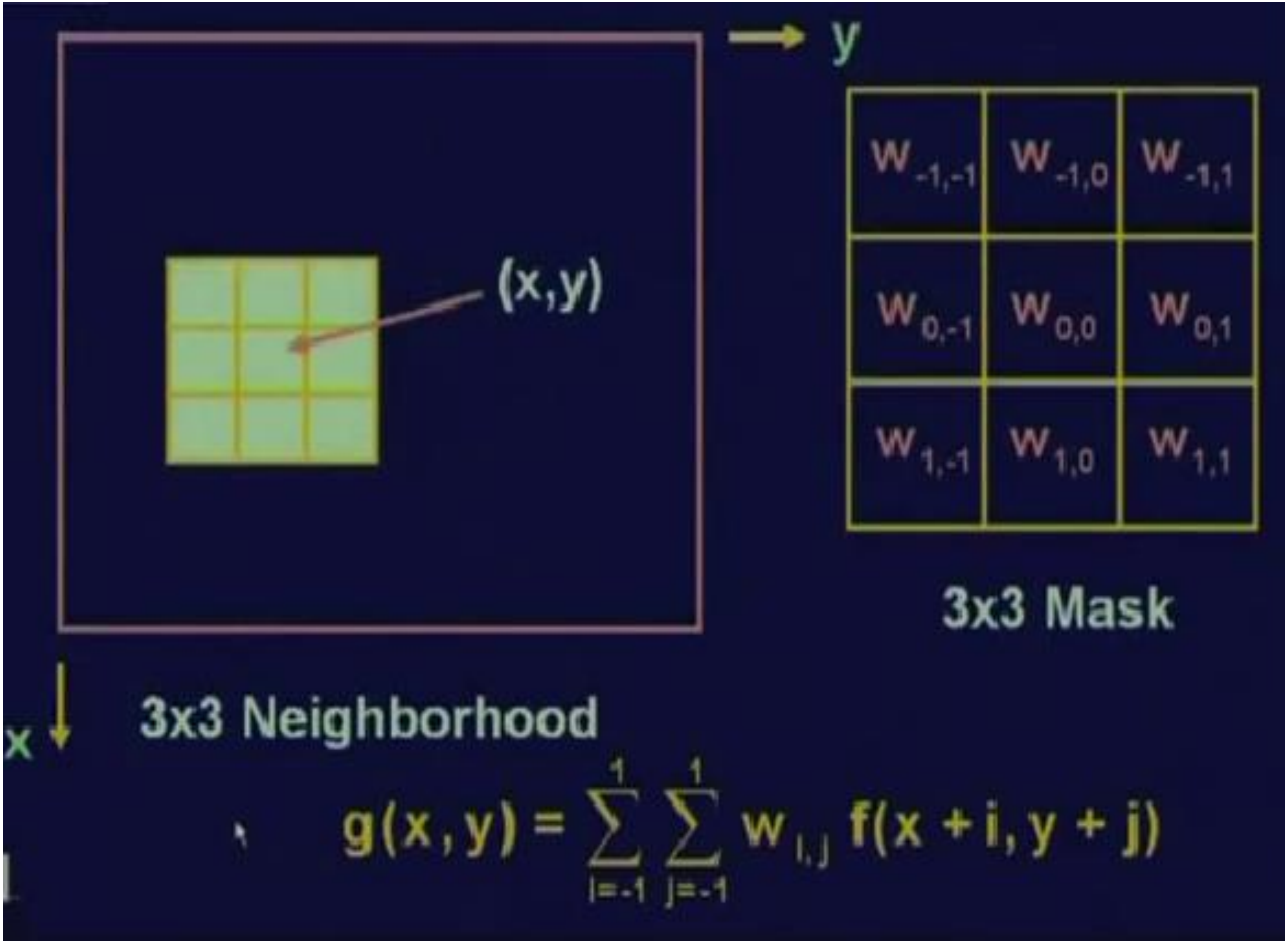
# Spatial Filters

## 1.Convolution-based filters

**(**shift-multiply-sum operations)**:**

- It uses spatial masks also known as kernels, templates and windows.
- The spatial mask is convolved with the given image to achieve the required smoothing or

# Mask Processing



# Mask Processing

- Based on the type of operation to be done the value of ‘Wi’ is chosen.(sharpening, averaging, etc.,)
- Appropriately the type of mask is chosen 3X3 or 5X5 or 7X7

# Neighborhood Operations

The value assigned to a pixel is a function of its gray label and the gray labels of its neighbors.

| | | |
|---|---|---|
| $Z_1$ | $Z_2$ | $Z_3$ |
| $Z_4$ | $Z_5$ | $Z_6$ |
| $Z_7$ | $Z_8$ | $Z_9$ |

| | | |
|---|---|---|
| $W_{-1,-1}$ | $W_{-1,0}$ | $W_{-1,1}$ |
| $W_{0,-1}$ | $W_{0,0}$ | $W_{0,1}$ |
| $W_{1,-1}$ | $W_{1,0}$ | $W_{1,1}$ |

3x3 Mask

$$Z = 1/9\ (Z_1+Z_2+Z_3+\ldots\ldots+Z_9) = \text{Average}$$

# Spatial Filters

## 2.Order-statistics(rank) filters:

- They do not use convolution techniques.
- They simply arrange the pixels that are under the mask in a desired order.
- Ex: Median filter.
- They are mainly used for image restoration.

# Order Statistic Filter

Let us assume a mask of size 3 × 3. The pixel values are arranged in ascending order $I_1 \leq I_2 \leq \ldots \leq I_9$ based on the grey scale value. The order determines the value that should replace the central pixel.

| | | |
|---|---|---|
| 12 | 13 | 15 |
| 17 | 14 | 18 |
| 19 | 20 | 21 |

Using the application of order statistics, this is arranged as

$$\{12, 13, 14, 15, 17, 18, 19, 20, 21\}$$

Minimum Median Maximum

# Spatial Filters

## 3.Hybrid filters:

- Uses the concepts of ranking and convolution.
- Unsharp masking is a good example of hybrid filters.

# Image Smoothing Filter

- A Smoothing filter is a linear filter.
- It creates an image with a smooth appearance by blurring the image and by removing noise such as Gaussian noise.

# Image Smoothing Filter

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

Fig. 5.18 Sample 3 × 3 2D spatial mask

# Image Smoothing Filter

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

Fig. 5.18 Sample 3 × 3 2D spatial mask

**Example:**

If the image is given as below, the above mask leads to the computation of the average of the neighborhood as: 1/9[1+2+3+5+4+6+7+9] = 1/9[45] = 5

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

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 5 | 5 | 6 |
| 7 | 8 | 9 |

(b) (c)

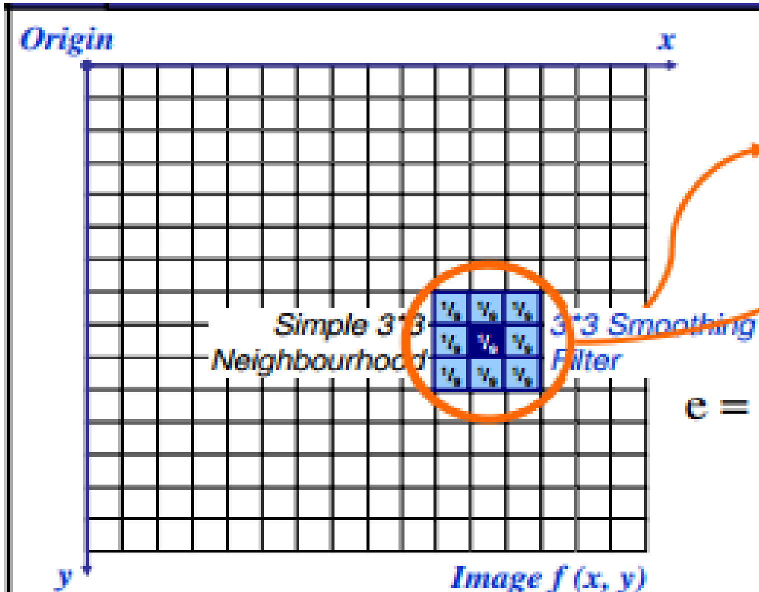
(b) Original image (c) Filtering with 3 × 3 mask

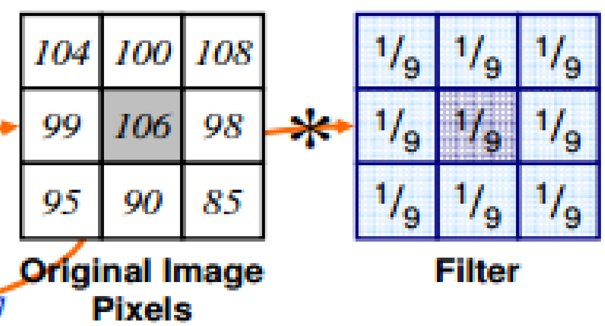
# Variable Weights

$$\frac{1}{10}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad \frac{1}{4}\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad \frac{1}{8}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Fig. 5.21 Sample masks with variable weights

# Example – Smoothing Operation




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

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

# Image Smoothing Example

The image at the top left is an original image of size 500*500 pixels

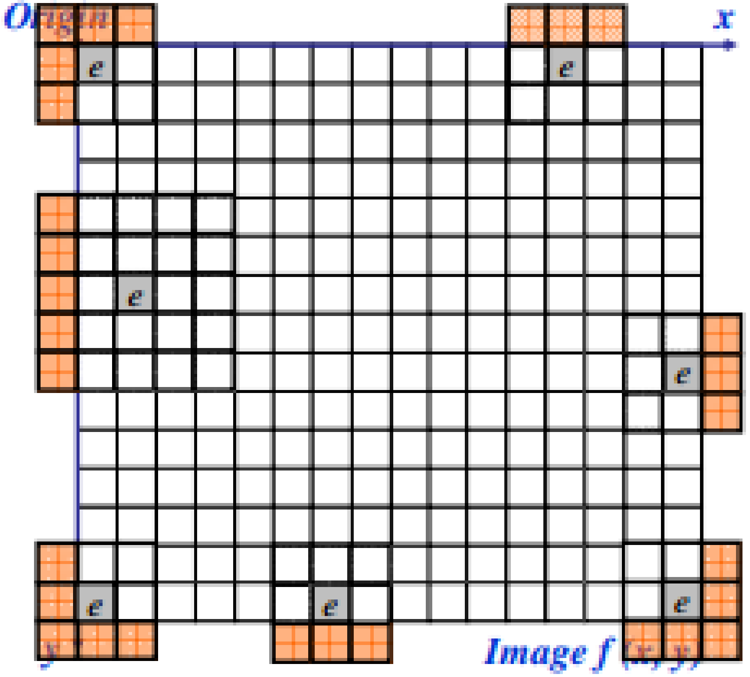
The subsequent images show the image after filtering with an averaging filter of increasing sizes

- 3, 5, 9, 15 and 35

Notice how detail begins to disappear

# Strange Things Along the Edges

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



# Possible Solutions

- Here are a few approaches to dealing with  the missing edge pixels:
- – Omit missing pixels
  - Only works with some filters
  - Can add extra code and slow down processing
- – Pad the image
  - Typically with either all white or all black pixels
- – Replicate border pixels
- – Truncate the image
- – Allow pixels wrap around the image
  - • Can cause some strange image artifacts

# Apply mean, mode and median filter

| | | | | |
|---|---|---|---|---|
| 10 | 20 | 30 | 40 | 50 |
| 12 | 13 | 12 | 11 | 12 |
| 12 | 11 | 23 | 34 | 45 |
| 11 | 22 | 33 | 30 | 30 |
| 44 | 55 | 66 | 77 | 88 |

**Mean**

| | | | | |
|---|---|---|---|---|
| 6 | 9 | 13 | 12 | 10 |
| 8 | 13 | 18 | 17 | 14 |
| 15 | 23 | 32 | 28 | 23 |
| 13 | 19 | 26 | 22 | 18 |
| 11 | 17 | 23 | 21 | 17 |

**Mode**

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 12 | 11 | 12 | 0 |
| 0 | 12 | 11 | 12 | 0 |
| 0 | 11 | 11 | 30 | 0 |
| 0 | 0 | 0 | 0 | 0 |

**Median**

| | | | | |
|---|---|---|---|---|
| 0 | 12 | 12 | 12 | 0 |
| 11 | 12 | 20 | 30 | 12 |
| 11 | 12 | 22 | 30 | 12 |
| 11 | 23 | 33 | 34 | 30 |
| 0 | 22 | 30 | 30 | 0 |

| | | | | |
|---|---|---|---|---|
| 0 | 12 | 12 | 12 | 0 |
| 11 | 12 | 20 | 30 | 12 |
| 11 | 12 | 22 | 30 | 12 |
| 11 | 23 | 33 | 34 | 30 |
| 0 | 22 | 30 | 30 | 0 |

| | | | | |
|---|---|---|---|---|
| 6.1111 | 10.7778 | 14.0000 | 17.2222 | 12.5556 |
| 8.6667 | 15.8889 | 21.5556 | 28.5556 | 21.3333 |
| 9.0000 | 16.5556 | 21.0000 | 25.5556 | 18.0000 |
| 17.2222 | 30.7778 | 39.0000 | 47.3333 | 33.7778 |
| 14.6667 | 25.6667 | 31.4444 | 36.0000 | 25.0000 |

# Gaussian Filters

- Gaussian filters choose the weight of the mask according to the shape of the Gaussian function.
- The value of the center pixel is greater and as the distance between the pixel and the center pixel increases, the mask weight decreases.
- Useful in smoothing as well as removing noises of normal distribution types.

# Gaussian Filters

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

(a) (b)

Fig. 5.22 Gaussian filters (a) Gaussian low-pass 3 × 3 filter (b) Gaussian low-pass 5 × 5 filter

# Directional Smoothing

- Most of the filters are isotropic.(Effect of filter is same in all the directions).
- In image processing applications, it may be necessary to select only certain features, in particular direction. Such filters are called anisotropic filters.
- Directional smoothing filters are useful in reducing the effect of edges from blurring by excessive smoothing.

# Directional Smoothing - Procedure

1. The spatial average is calculated in several directions.
2. The direction that is associated with the minimum is detected and is used as part of the convolution process to replace the centre pixel.

# Conservative smoothing

1. For a centre pixel, find the pixel values of its 8-neighborhood.
2. Find the maximum and minimum pixel values.
3. Compare the value of the centre pixel with the maximum and minimum values.

   (a) If the value of the centre pixel > maximum value, set the centre pixel value to the maximum value.

   (b) If the value of the centre pixel < minimum value, set the centre pixel value to the minimum value.

   (c) Otherwise, retain the centre pixel value as it is.

# Sharpening filters

- It enhances the details of an image.
- High frequency components have detailed information in the form of edges and boundaries.
- Edges are significant local intensity variations that exists between two different regions.
- Sharpening algorithms are used to separate object outline. So, they are also called as edge enhancement or edge crispening algorithm.

# Sharpening filters

- High pass and Laplacian filters extracts edges.
- All contrast information is lost.
- This affects the quality of the image.

# Sharpening filters

- Using derivatives.
  - First order derivative filter.
  - Second order derivative filter.

# Sharpening filters

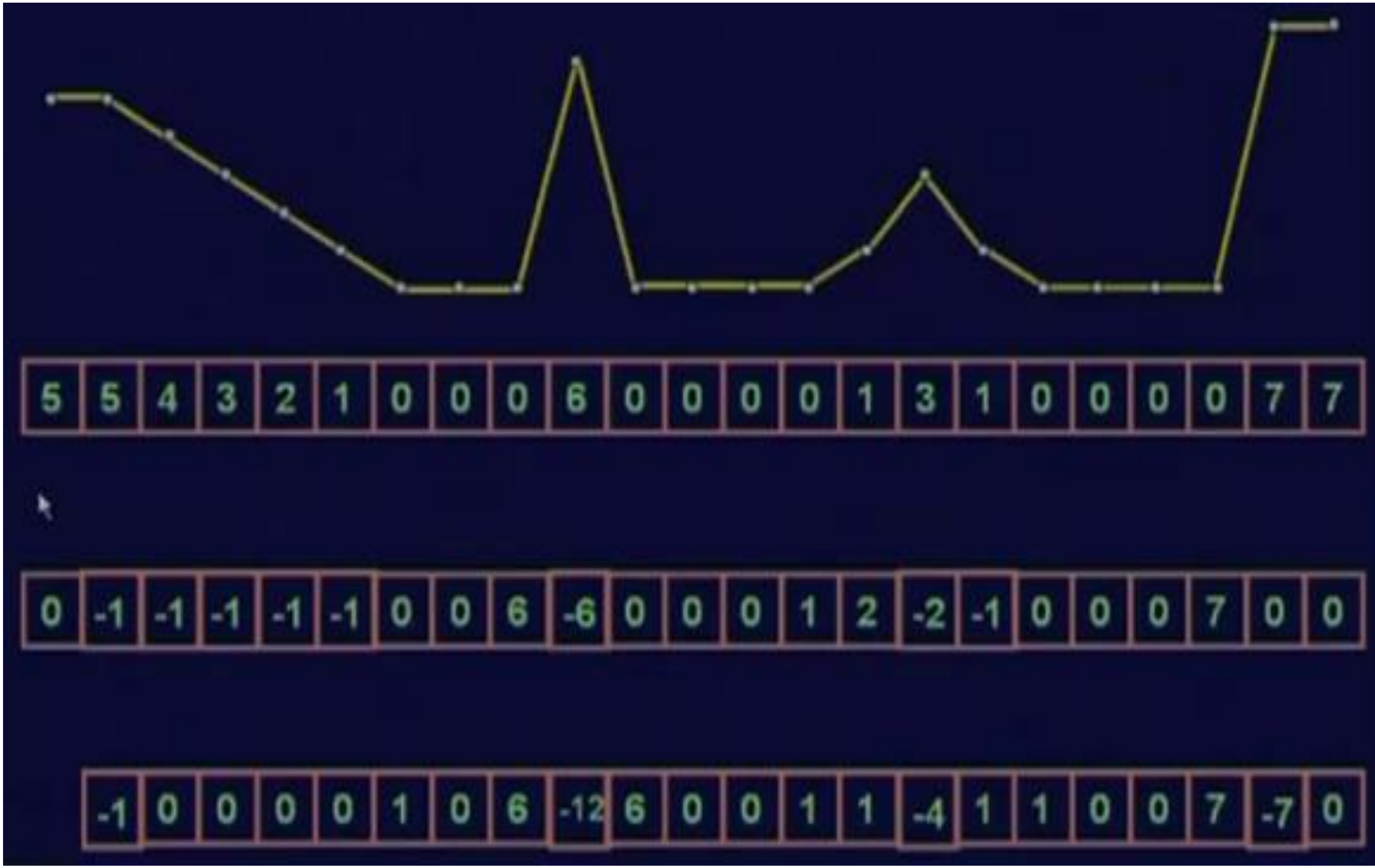
## First order derivative filter

- Must be zero in areas of constant gray level
- Non zero at the onset of a gray level step or ramp
- Non zero along ramps

## Second order derivative filter

- Zero in flat areas
- Non zero at onset and end of a gray level step or ramp
- Zero along ramps of constant slope

# Sharpening filters



# Sharpening filters

- First order derivative generally produce thicker edges in an image
- Second order derivatives give stronger response to fine details such as thin lines and isolated points
- First order derivative have stronger respose to gray level step
- Second order derivative produce a double response at step edges

→ Second order derivatives are better suited for image enhancement

# Sharpening filters

Laplacian Operator

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

1-D

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

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

2-D

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

# Sharpening filters

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

$$= \left[ f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) \right] - 4f(x, y)$$

# Sharpening filters (Laplacian Mask)

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

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | - 8 | 1 |
| 1 | 1 | 1 |

# Sharpening filters (Laplacian Mask)

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

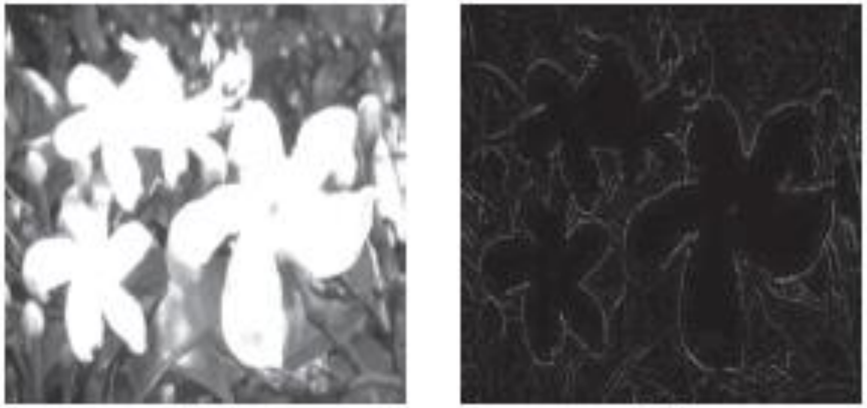
| | | |
|---|---|---|
| -1 | -1 | -1 |
| -1 | 8 | -1 |
| -1 | -1 | -1 |

# Sharpening Filters

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & +4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad \begin{pmatrix} -1 & -1 & -1 \\ -1 & +8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Fig. 5.24 Four sample Laplacian masks



(a) (b)

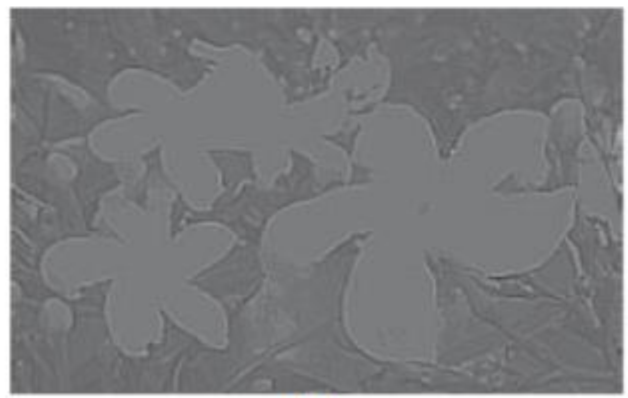
Fig. 5.25 Image sharpening spatial filters (a) Original image (b) Laplacian high-pass filter result

# High-Boost Filter

$$\text{High-boost image} = (A)\,(\text{Original}) - (\text{Low-pass})$$

$$= (A-1)\,(\text{Original}) + (\text{Original} - \text{Low-pass})$$

$$= (A-1)\,(\text{Original}) + (\text{High-pass})$$



(b)

(b) Result of a high-boost filter

# Unsharp Masking

The procedure for implementing an unsharp mask is as follows:

1. Read the image.
2. Blur the image using any image smoothing filters. This stage requires a convolution based smoothing filter. Let the smooth or blurred image be $\bar{f}(x, y)$.
3. Let the mask = original image – $\bar{f}(x, y)$.

   Subtracting the blurred version from the original image results in an image where there is a visible emphasis in edges.
4. Add to the original image the weighted portion of the mask, to restore some of the lost visual information.

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

**If k=1 → unsharp masking**
**If k>1 → High – Boost filtering**