Lecture No 14

Unit 3-Image Enhancement

Spatial Filters

Lecture By

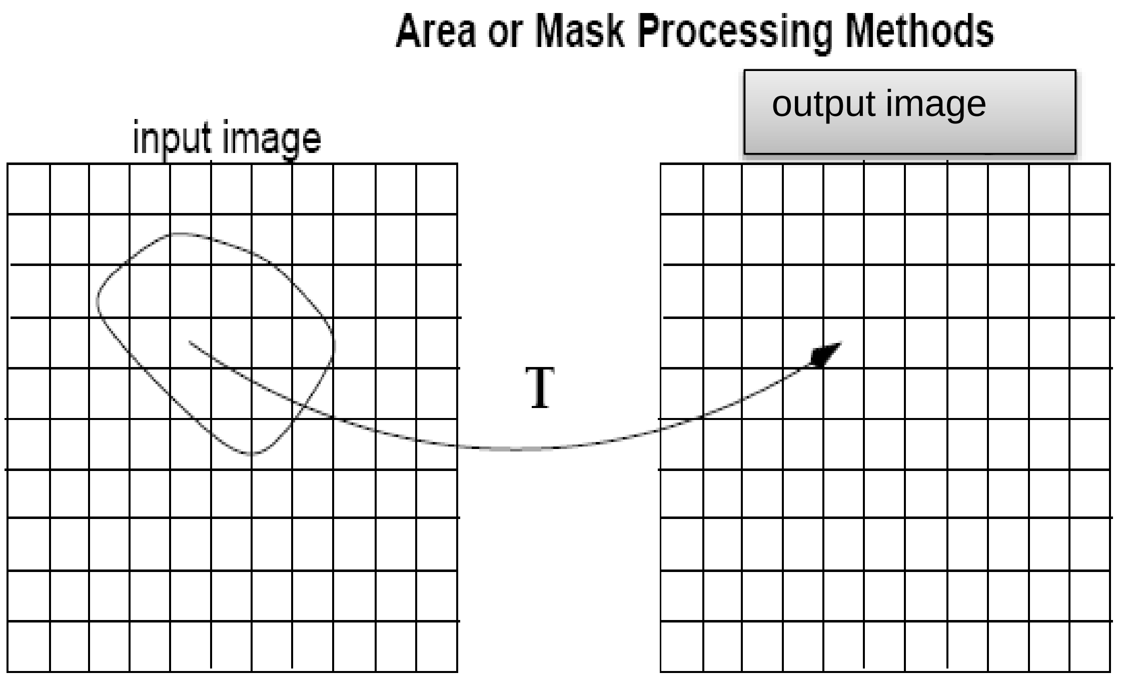
Kalyan Acharjya

kalyan5.blogspot.in

Disclaimer

- This PPT contents are copied from various publicly available sources and all images are copyright of the respective original owner.
- This is used for academic purpose only.

Spatial Filtering Methods
(or Mask Processing Methods)



$$g(x,y) = T[f(x,y)]$$

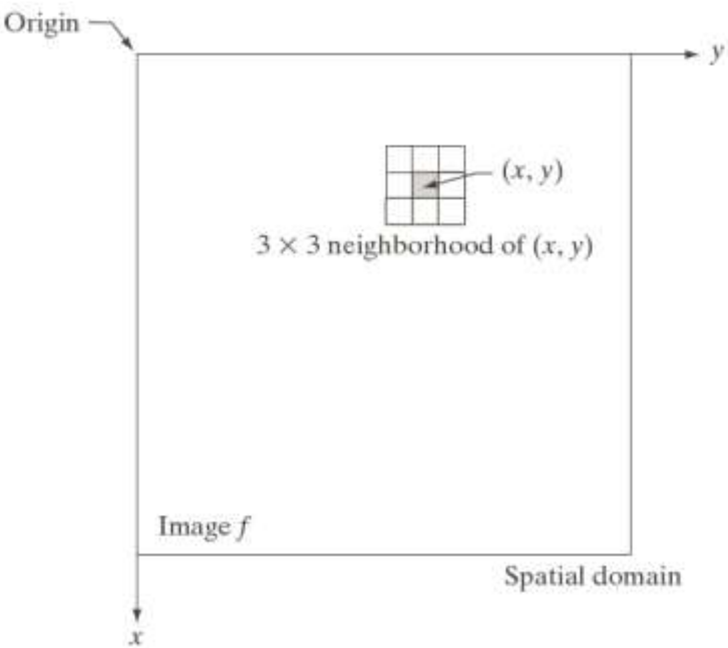
T operates on a neighborhood of pixels

Spatial Filtering (cont'd)

- Filters are classified as:
  - **Low-pass** (i.e., preserve low frequencies)
  - **High-pass** (i.e., preserve high frequencies)
  - **Band-pass** (i.e., preserve frequencies within a band)
  - **Band-reject** (i.e., reject frequencies within a band)

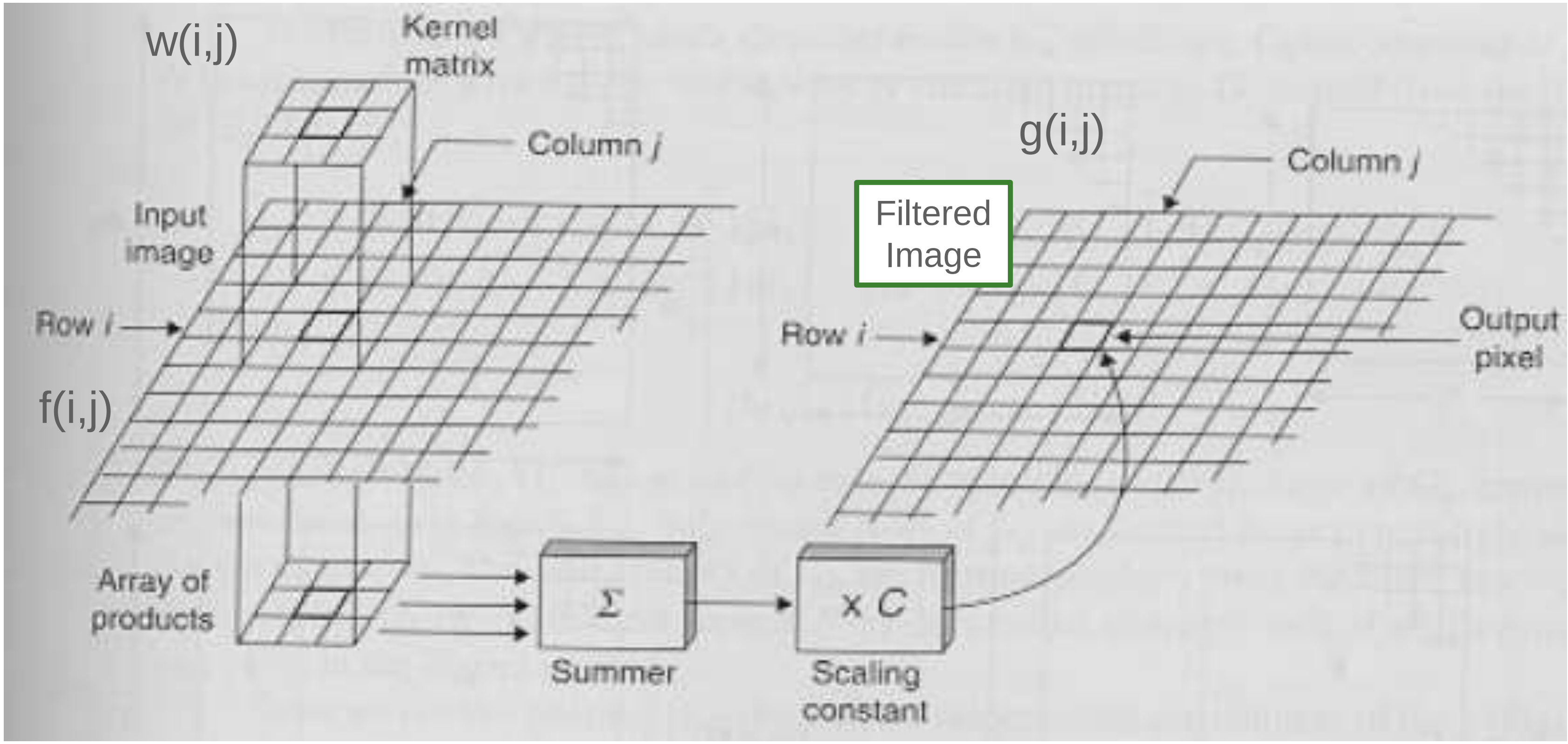
Spatial Filtering (cont'd)

- Need to define:
  - ✓ A neighborhood (or mask)
  - ✓ An Mask Operation



•Typically, the neighborhood is rectangular and its size is much smaller than that of f(x,y) - e.g., 3x3 or 5x5

Operation:



A filtered image is generated as the center of the mask moves to every pixel in the input image.

Spatial filtering - Operation

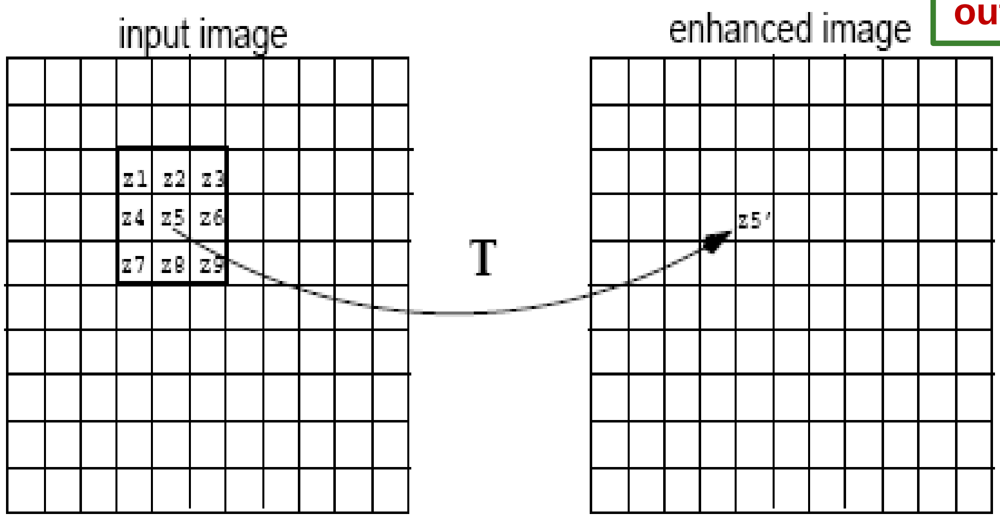
Example: weighted sum of input pixels.

```
z5' = R = w1z1 + w2z2 + ... + z9w9
```

mask weights:

w1	w2	w3
w4	w5	w6
w7	w8	w9

Area or Mask Processing Methods



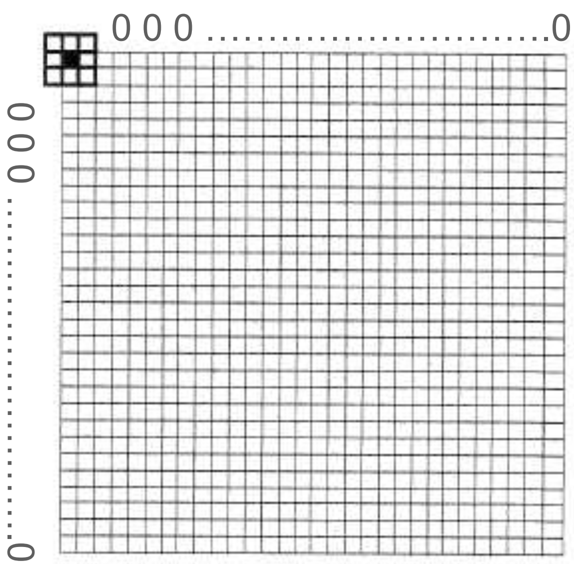
output image

$$g(x,y) = T[f(x,y)]$$

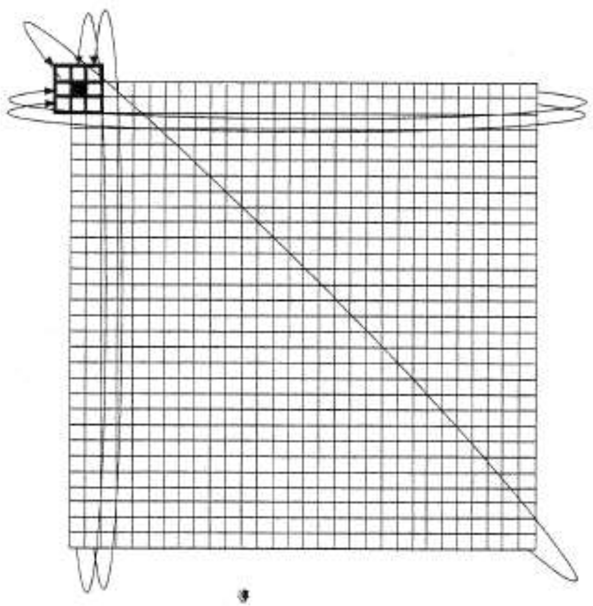
T operates on a neighborhood of pixels

Handling Pixels Close to Boundaries

Pad With Zeroes



or



Linear vs Non-Linear
Spatial Filtering Methods

- A filtering method is linear when the output is a weighted sum of the input pixels.

w1	w2	w3
w4	w5	w6
w7	w8	w9

$$z5' = R = w1z1 + w2z2 + \ldots + z9w9$$

- **Methods that do not satisfy the above property are called non-linear.**
  - e.g.,

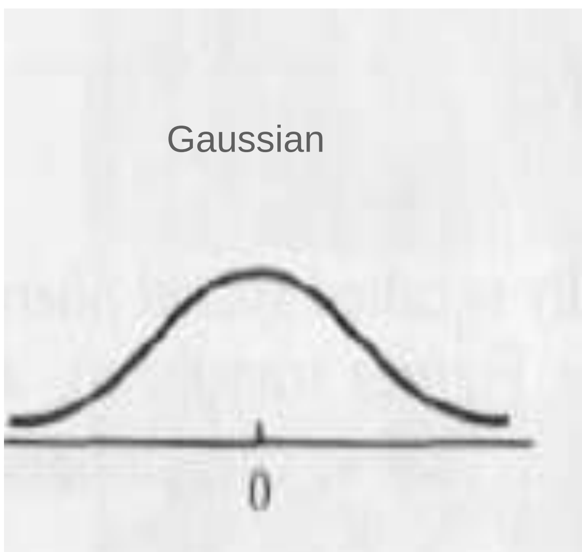
$$z_5' = max(z_k, k = 1, 2, \ldots, 9)$$

Spatial Filters

- We will mainly focus on two types of filters:
  - **Smoothing** (low-pass)
  - **Sharpening** (high-pass)

Smoothing Filters (low-pass)

- Useful for reducing noise and eliminating small details.
  - The elements of the mask must be **positive**.
  - Sum of mask elements is 1 (after normalization).



7×7 Gaussian mask

1	1	2	2	2	1	1
1	2	2	4	2	2	1
2	2	4	8	4	2	2
2	4	8	16	8	4	2
2	2	4	8	4	2	2
1	2	2	4	2	2	1
1	1	2	2	2	1	1

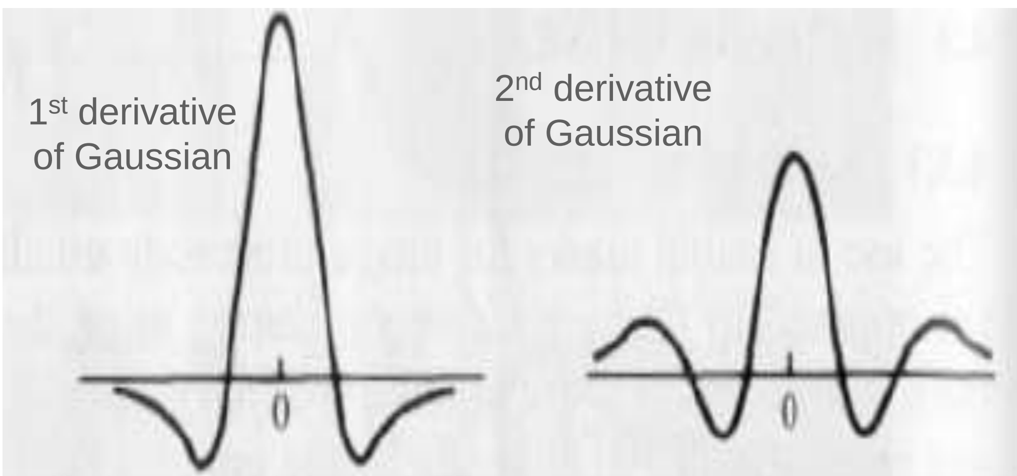
Smoothing filters – Example

Input Image

Smoothed Image

Sharpening Filters (high-pass)

- Useful for highlighting fine details.
  - The elements of the mask contain both **positive** and **negative** weights.
  - Sum of mask elements is 0.



mask

-1	-1	-1
-1	8	-1
-1	-1	-1

Sharpening Filters - Example

- **Warning**: the results of sharpening might contain negative values (i.e., re-map them to [0, 255])

Input Image

Sharpened Image

(for better visualization, the original image is added to the sharpened image)

Common Smoothing Filters

- Averaging
- Gaussian
- Median filtering (non-linear)

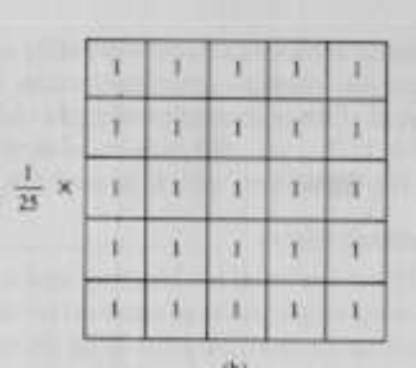
Smoothing Filters: Averaging Un-weighted

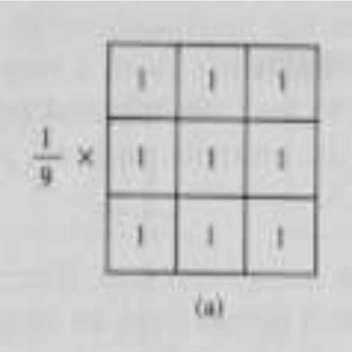
Input Image

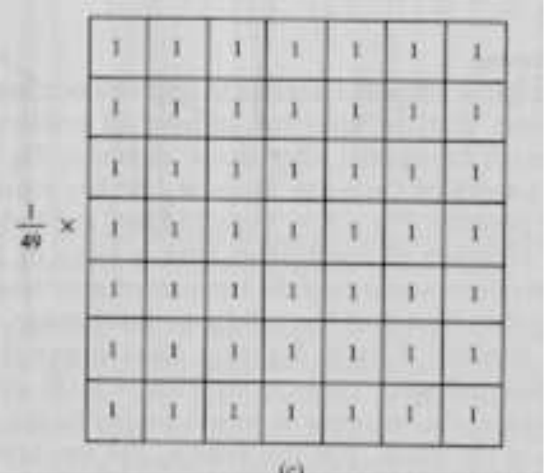
3	3	0	1	1
4	5	4	0	2
2	3	4	0	1
1	5	6	7	1
1	0	2	3	5
4	5	6	7	0

Mask

1	1	1
1	1	1
1	1	1



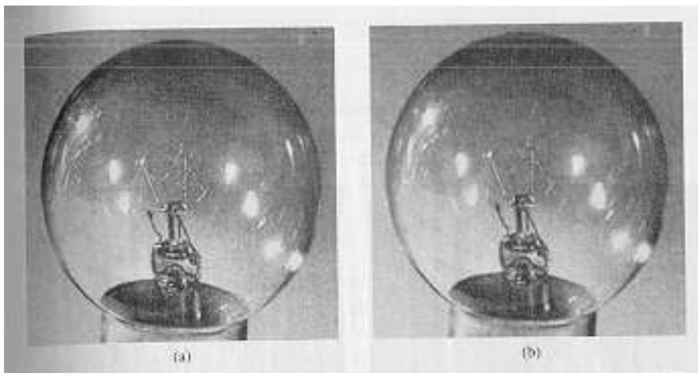




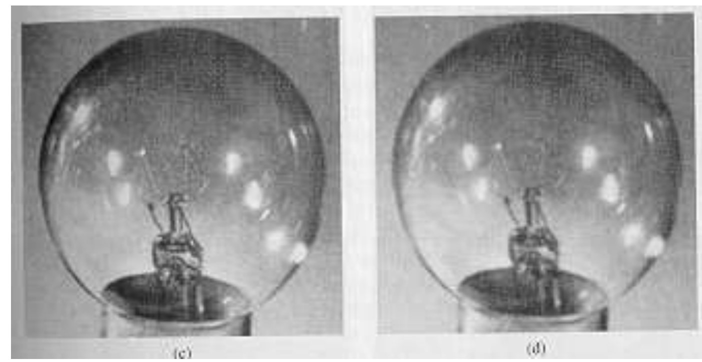
Smoothing Filters: Averaging (cont'd)

- Mask size determines the degree of smoothing (loss of detail).

original 3x3



5x5 7x7



15x15 25x25

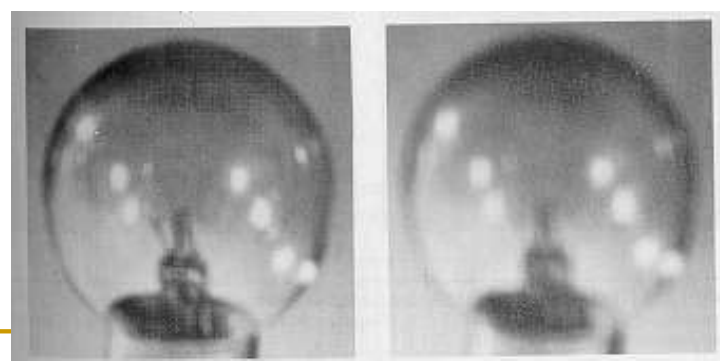


Image Smoothing

- This operation is equivalent to lowpass filtering.

Example of Image Blurring

Original Image

$$\frac{1}{N^2}\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}_{N\times N}$$

Avg. Mask

$N = 3$

$N = 5$

$N = 7$

$N = 11$

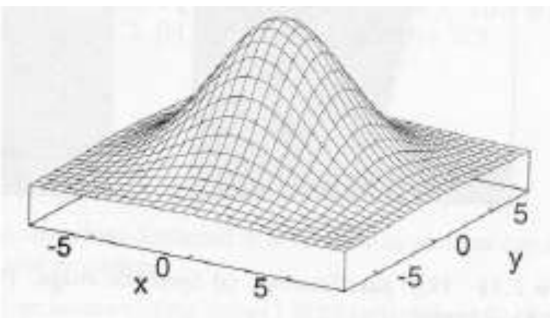
$N = 15$

$N = 21$

Smoothing filters: Gaussian

- The weights are samples of a 2D Gaussian function:

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{x^2 + y^2}{2\sigma^2}}$$



7 × 7 Gaussian mask

1	1	2	2	2	1	1
1	2	2	4	2	2	1
2	2	4	8	4	2	2
2	4	8	16	8	4	2
2	2	4	8	4	2	2
1	2	2	4	2	2	1
1	1	2	2	2	1	1

$\sigma = 1.4$

mask size is a function of **σ**: $height = width = 5\sigma$ (subtends 98.76% of the area)

Smoothing filters: Gaussian (cont'd)

• **σ** controls the amount of smoothing

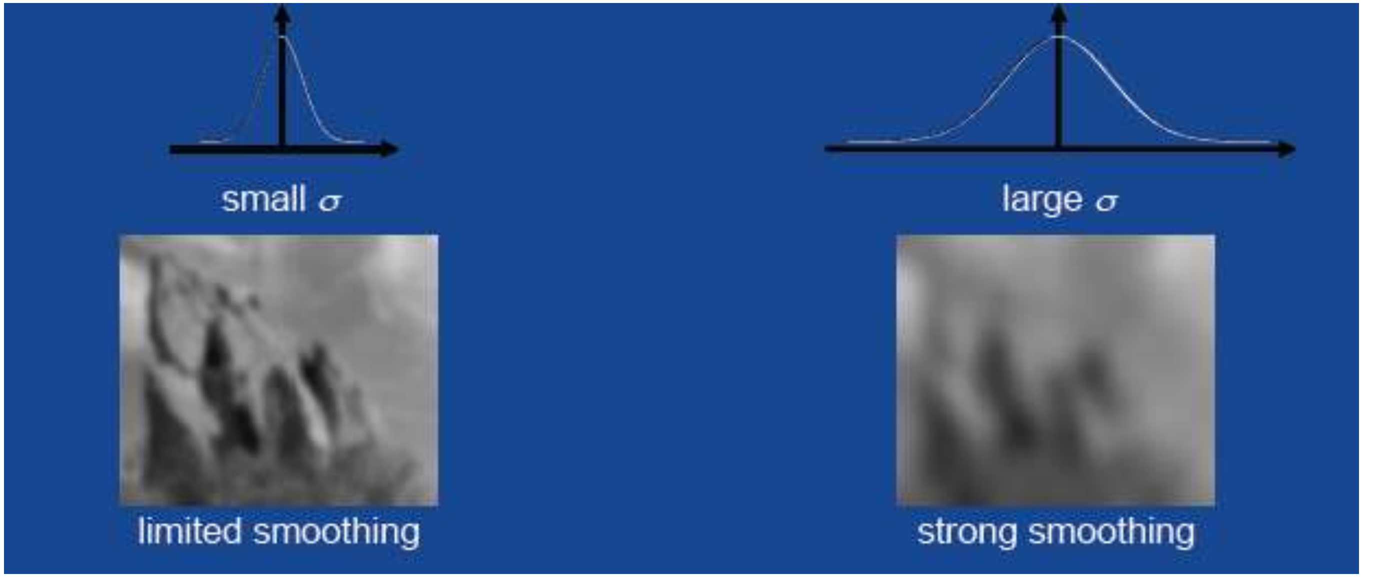
• As **σ** increases, more samples must be obtained to represent the Gaussian function accurately.

15 × 15 Gaussian mask

$\sigma = 3$

2	2	3	4	5	5	6	6	6	5	5	4	3	2	2
2	3	4	5	7	7	8	8	8	7	7	5	4	3	2
3	4	6	7	9	10	10	11	10	10	9	7	6	4	3
4	5	7	9	10	12	13	13	13	12	10	9	7	5	4
5	7	9	11	13	14	15	16	15	14	13	11	9	7	5
5	7	10	12	14	16	17	18	17	16	14	12	10	7	5
6	8	10	13	15	17	19	19	19	17	15	13	10	8	6
6	8	11	13	16	18	19	20	19	18	16	13	11	8	6
6	8	10	13	15	17	19	19	19	17	15	13	10	8	6
5	7	10	12	14	16	17	18	17	16	14	12	10	7	5
5	7	9	11	13	14	15	16	15	14	13	11	9	7	5
4	5	7	9	10	12	13	13	13	12	10	9	7	5	4
3	4	6	7	9	10	10	11	10	10	9	7	6	4	3
2	3	4	5	7	7	8	8	8	7	7	5	4	3	2
2	2	3	4	5	5	6	6	6	5	5	4	3	2	2

Smoothing filters: Gaussian (cont'd)



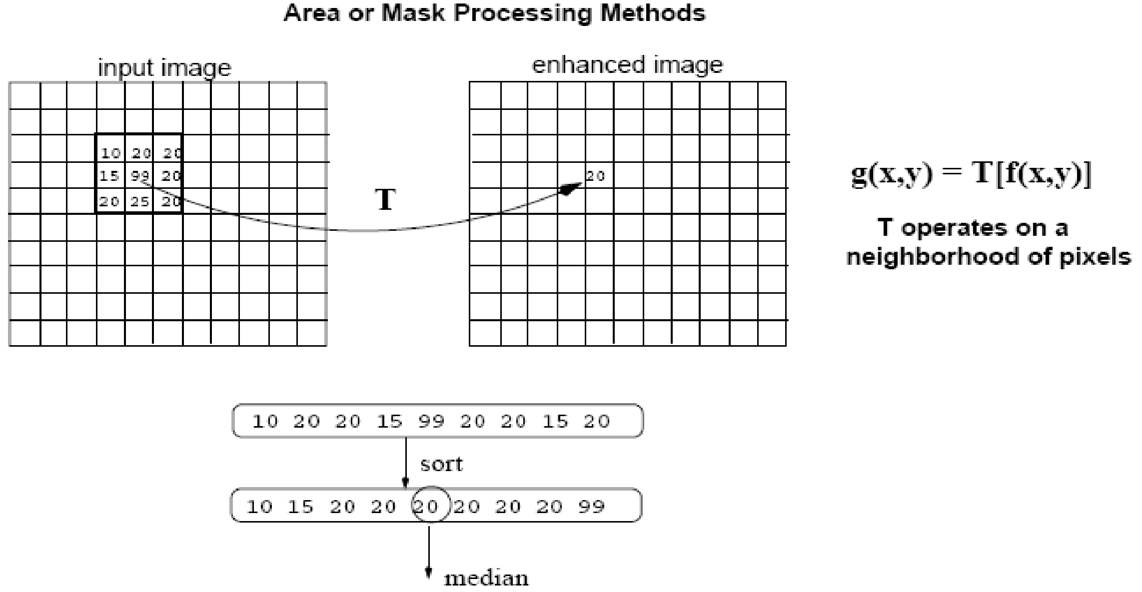
Averaging vs. Gaussian Smoothing

Averaging

Gaussian

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.

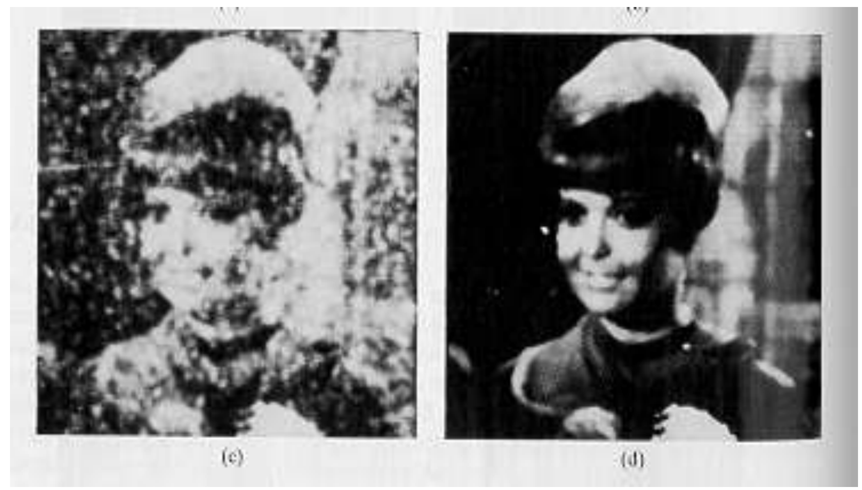


Smoothing Filters: Median Filtering (non-linear)

- Very effective for removing "salt and pepper" noise (i.e., random occurrences of black and white pixels).

Original Image   Image with Noise

Averaging   Median Filtering



Common Sharpening Filters

- Unsharp masking
- High Boost filter
- Gradient (1st derivative)
- Laplacian (2nd derivative)

Sharpening Filters: Un sharp Masking

- Obtain a sharp image by subtracting a low pass filtered (i.e., smoothed) image from the original image:

$$Highpass = Original - Lowpass$$

(after contrast enhancement)

Sharpening Filters: High Boost

- Image sharpening emphasizes edges but details are lost.
- **High boost filter**: Amplify input image, then subtract a Low pass image.

$$Highboost = A\ Original - Lowpass$$

$$= (A-1)\ Original + Original - Lowpass$$

$$= (A-1)\ Original + Highpass$$

(A-1)

Sharpening Filters: High Boost (cont'd)

- If **A=1**, we get unsharp masking.
- If **A>1**, part of the original image is **added back** to the high pass filtered image.

$$\text{High boost} = (A-1)\ Original + Highpass$$

One way to implement high boost filtering is using the masks below

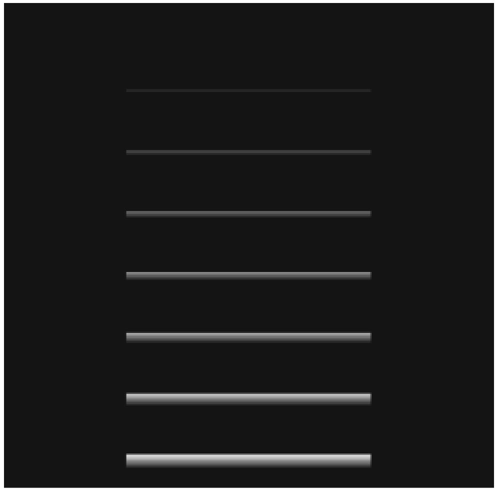
A>=1
w = 9A-1

-1	-1	-1
-1	w	-1
-1	-1	-1

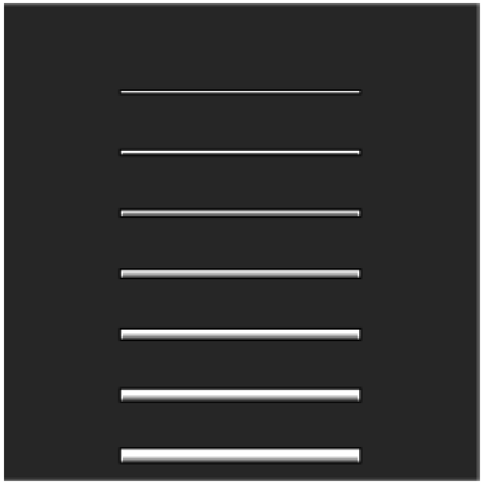
A=2
w = 17

-1	-1	-1
-1	17	-1
-1	-1	-1

Sharpening Filters: High Boost (cont'd)



A=1.4



A=1.9

Sharpening Filters: Derivatives

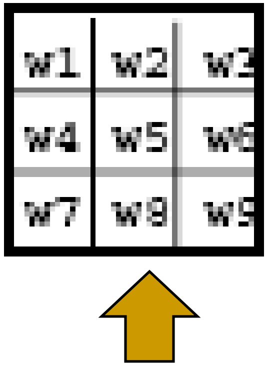
- Taking the derivative of an image results in sharpening the image.

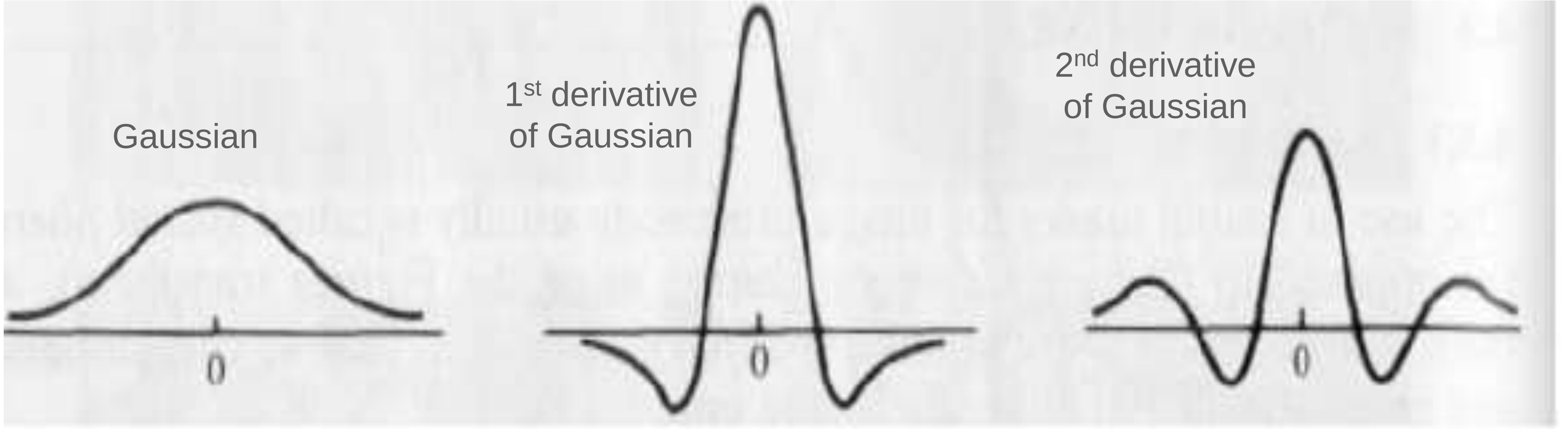
- The derivative of an image (i.e., 2D function) can be computed using the gradient.

$$grad(f) = \begin{pmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{pmatrix}$$

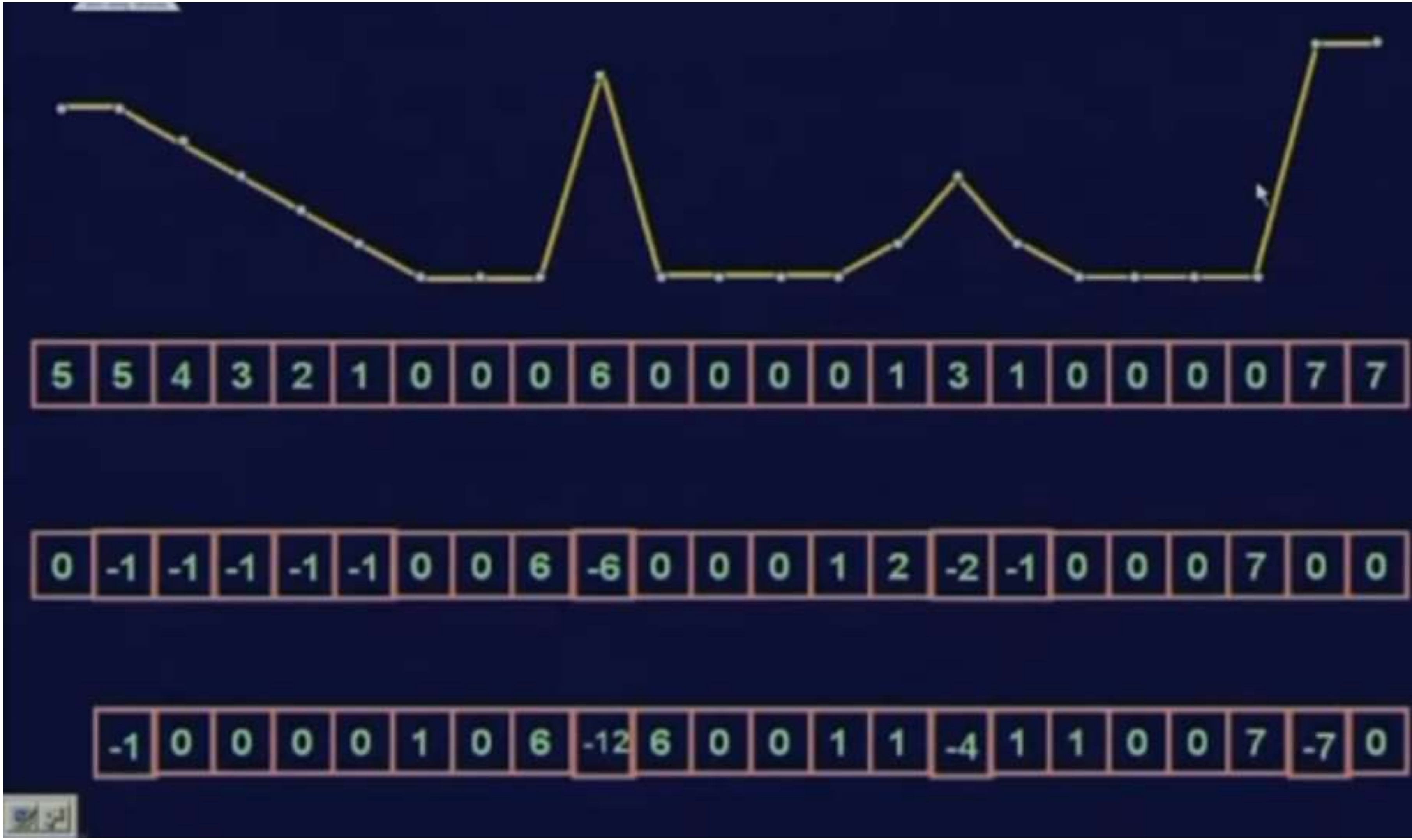
How do we choose the mask weights?

- Typically, by sampling certain functions:





First and Second Derivative

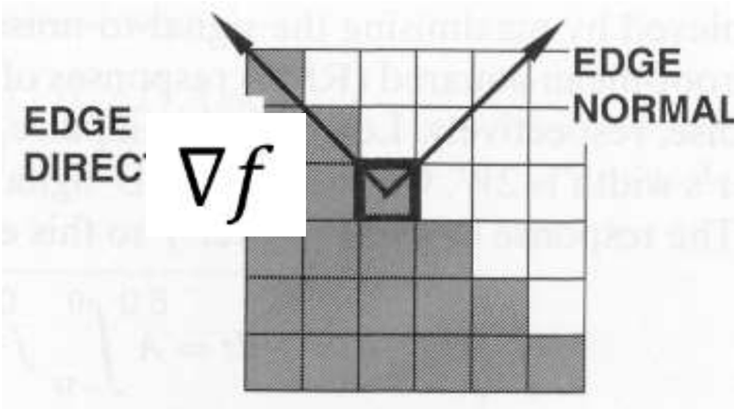


Gradient (cont'd)

- **Gradient magnitude:** provides information about edge strength.
- **Gradient direction:** perpendicular to the direction of the edge.

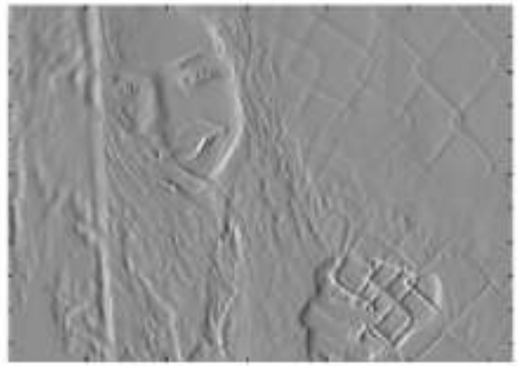
$$magnitude(grad(f)) = \sqrt{\frac{\partial f}{\partial x}^2 + \frac{\partial f}{\partial y}^2}$$

$$direction(grad(f)) = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$



Derivative Results and Laplacian:

$$\frac{\partial f}{\partial x}$$



$$\frac{\partial f}{\partial y}$$

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

Reminder: Assignment

Online Submission Due Date **10 Oct 2018**

1. Comments on Role of Digital Image Processing in Modern Imaging Based Medical Treatments.

2. Explain your view on Importance of Image Understanding in Recent Computer Vision Applications

Assignment Submission link is available at kalyan5.blogspot.in

OR

QR Code:

You can directly visit at http://bit.do/dipr_jnu

Any Questions?

Thank You