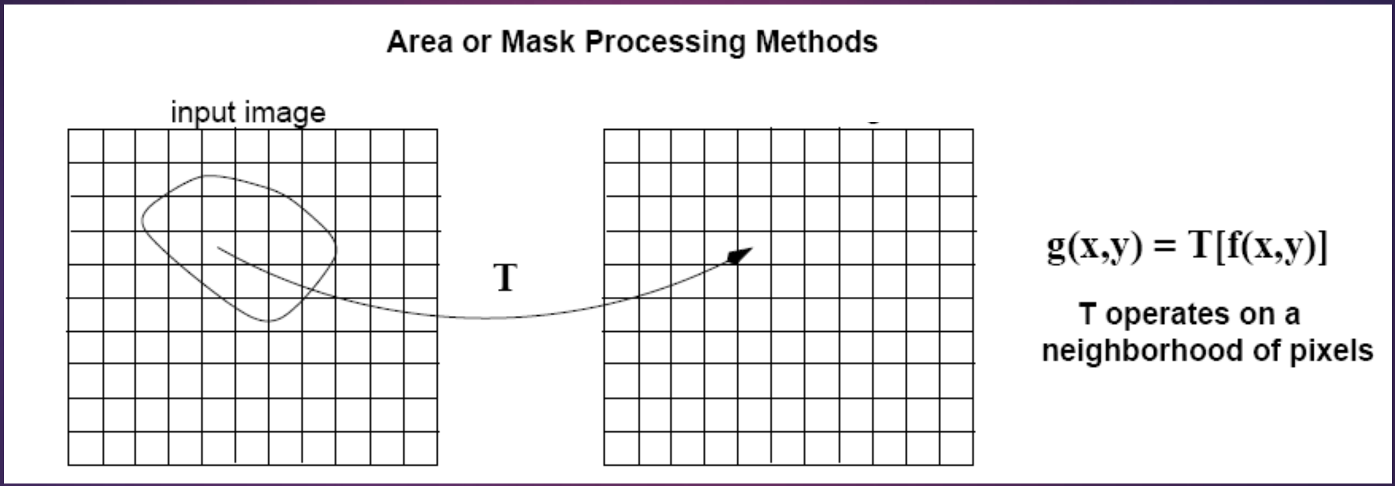
Spatial Filtering

Smoothening and Sharpening Filters

Spatial Filtering Methods (or Mask Processing Methods)



Spatial Filtering

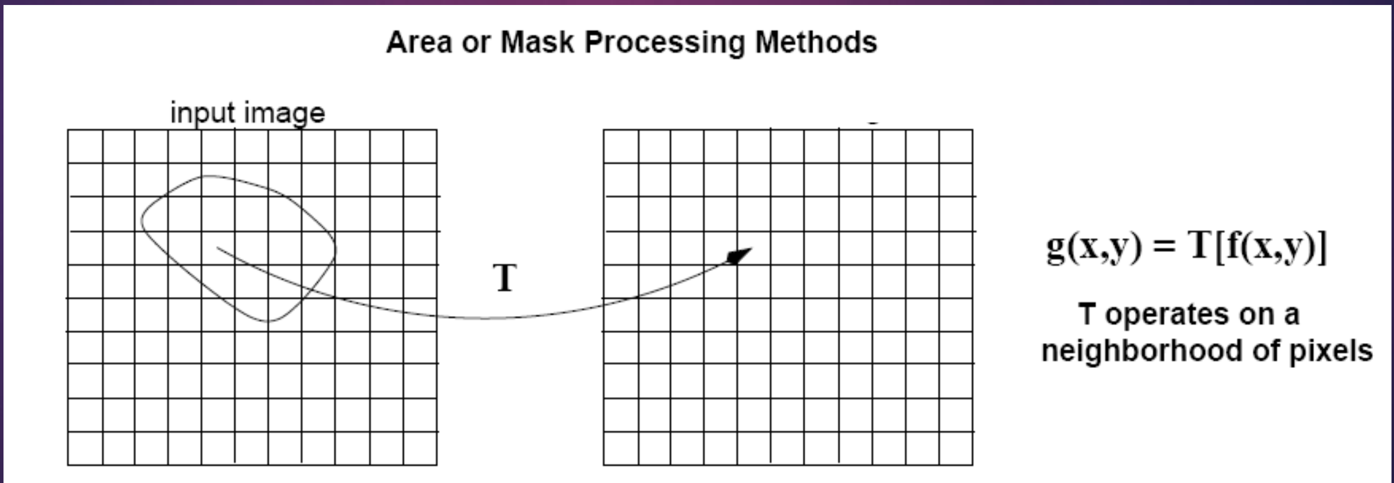
- The word “filtering” has been borrowed from the frequency domain.
- Filters are classified as:
  - –Low-pass (i.e., preserve low frequencies)
  - –High-pass (i.e., preserve high frequencies)
  - –Band-pass (i.e., preserve frequencies within a band)
  - –Band-reject (i.e., reject frequencies within a band)

Spatial Filtering (cont'd)

- Spatial filtering are defined by:

  A neighborhood

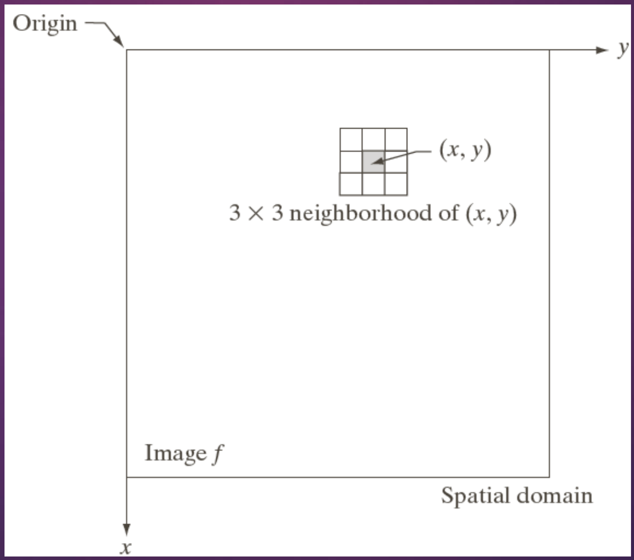
  An operation that is performed on the pixels inside the neighborhood



Spatial Filtering (cont'd)

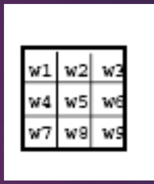
- Typically, the neighborhood is rectangular and its size is much smaller than that of *f(x,y)*

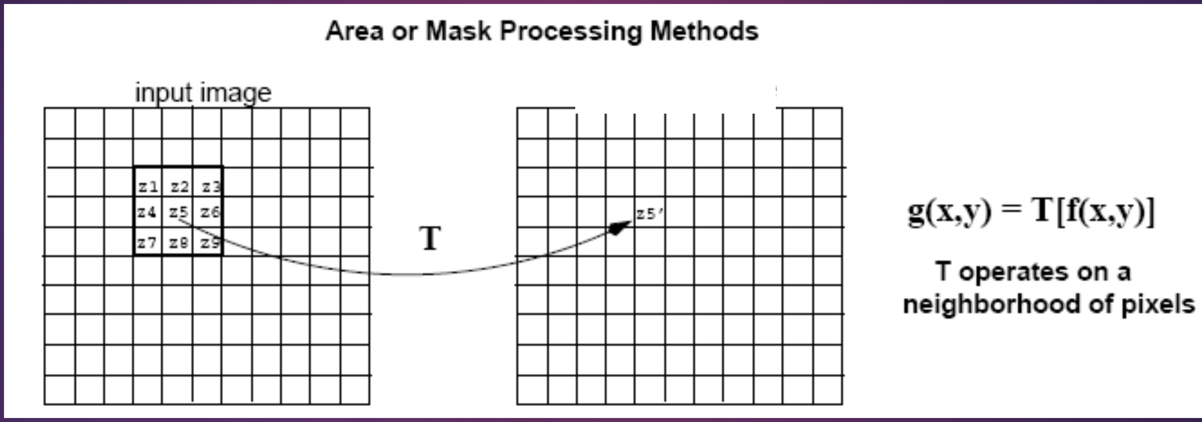
- e.g., 3x3 or 5x5



Spatial filtering (cont'd)

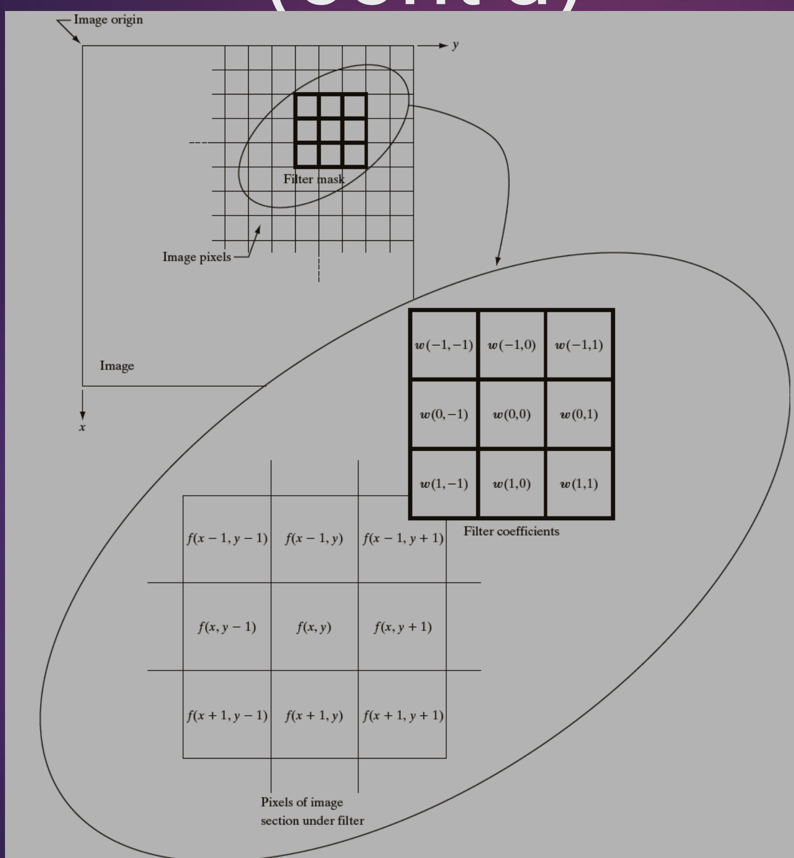
- Example: apply a "mask" of weights





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

Spatial filtering (cont'd)



Assume the origin of the mask is the center of the mask.

for a 3 x 3 mask:

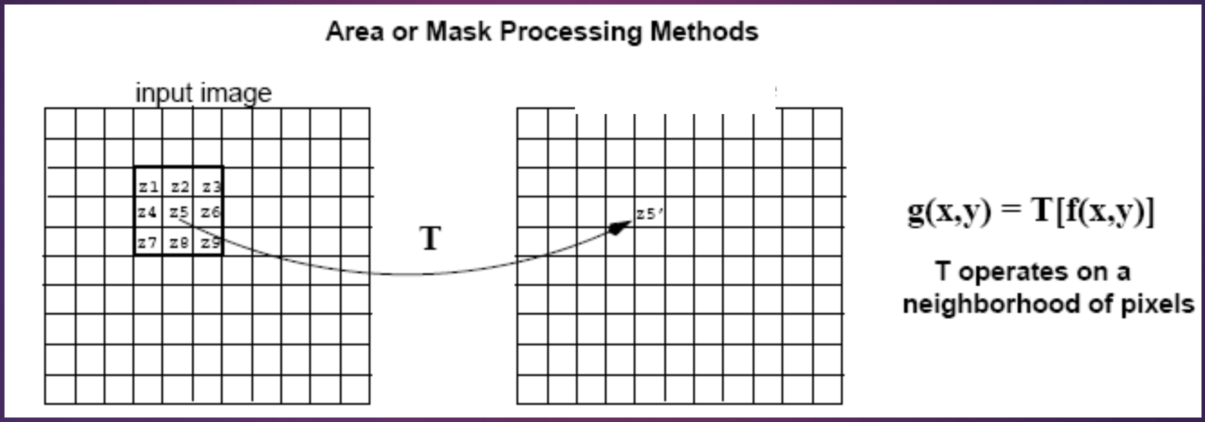
$$g(x, y) = \sum_{s=-1}^{1} \sum_{t=-1}^{1} w(s,t) f(x+s, y+t)$$

for a K x K mask:

$$g(x, y) = \sum_{s=-K/2}^{K/2} \sum_{t=-K/2}^{K/2} w(s, t) f(x+s, y+t)$$

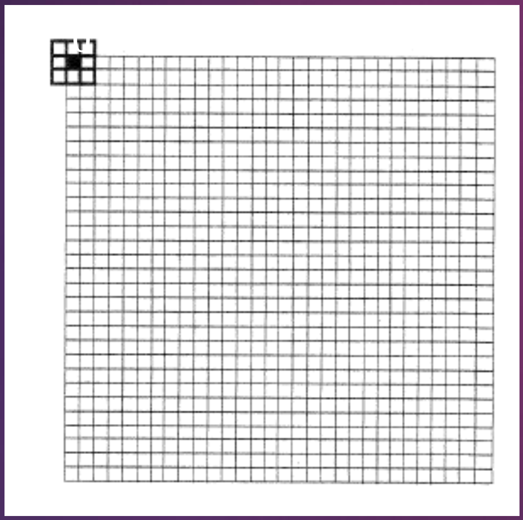
Spatial Filtering (cont'd)

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

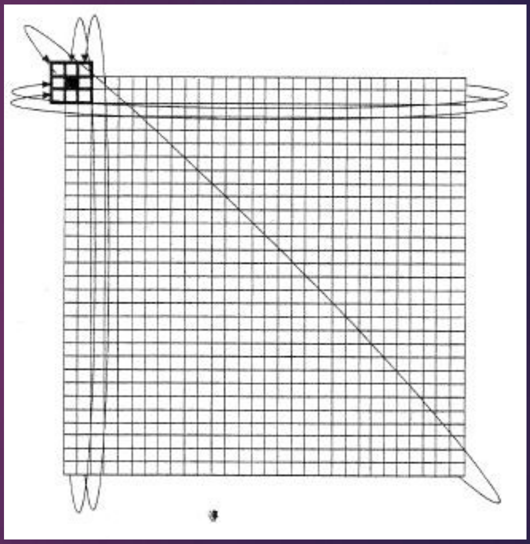


Handling Pixels Close to Boundaries

pad with zeroes

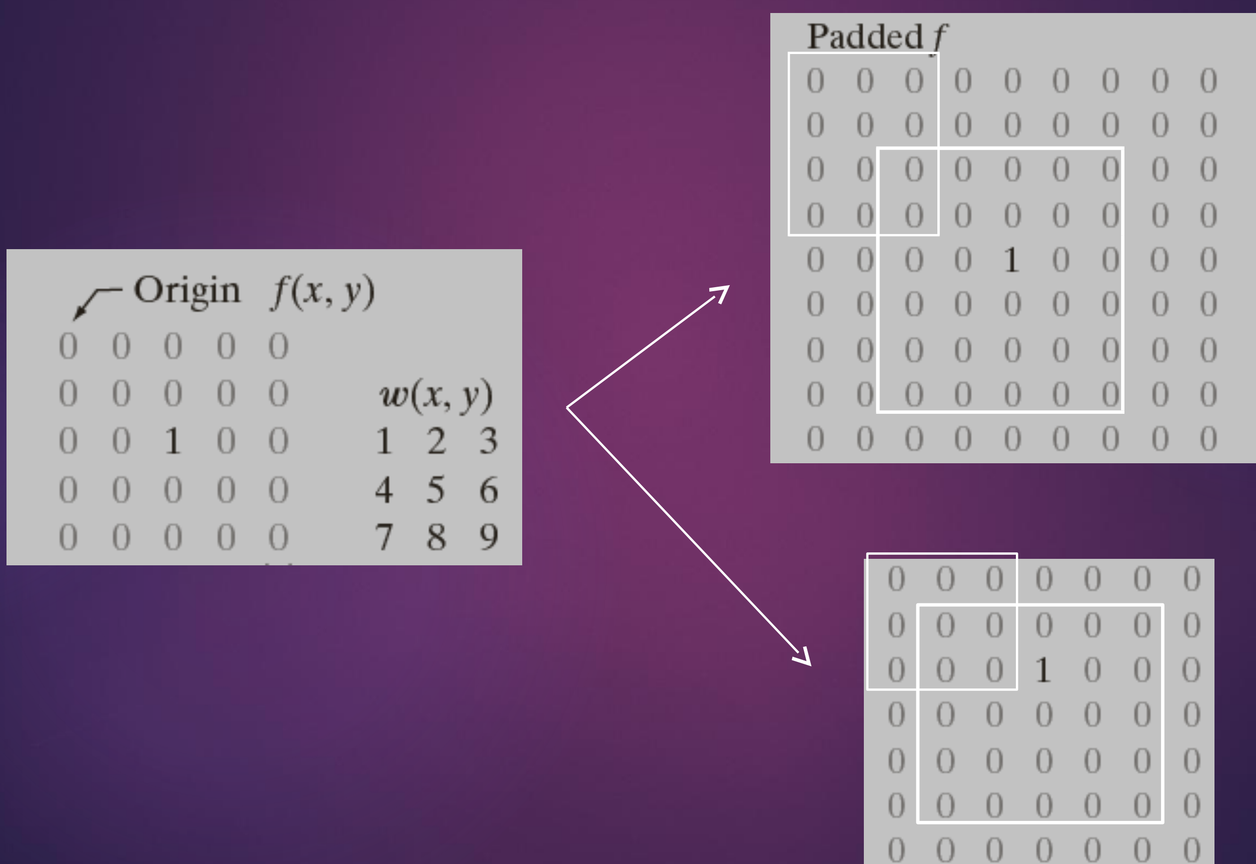


or



...
.0

Handling Pixels Close to Boundaries (cont'd)



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

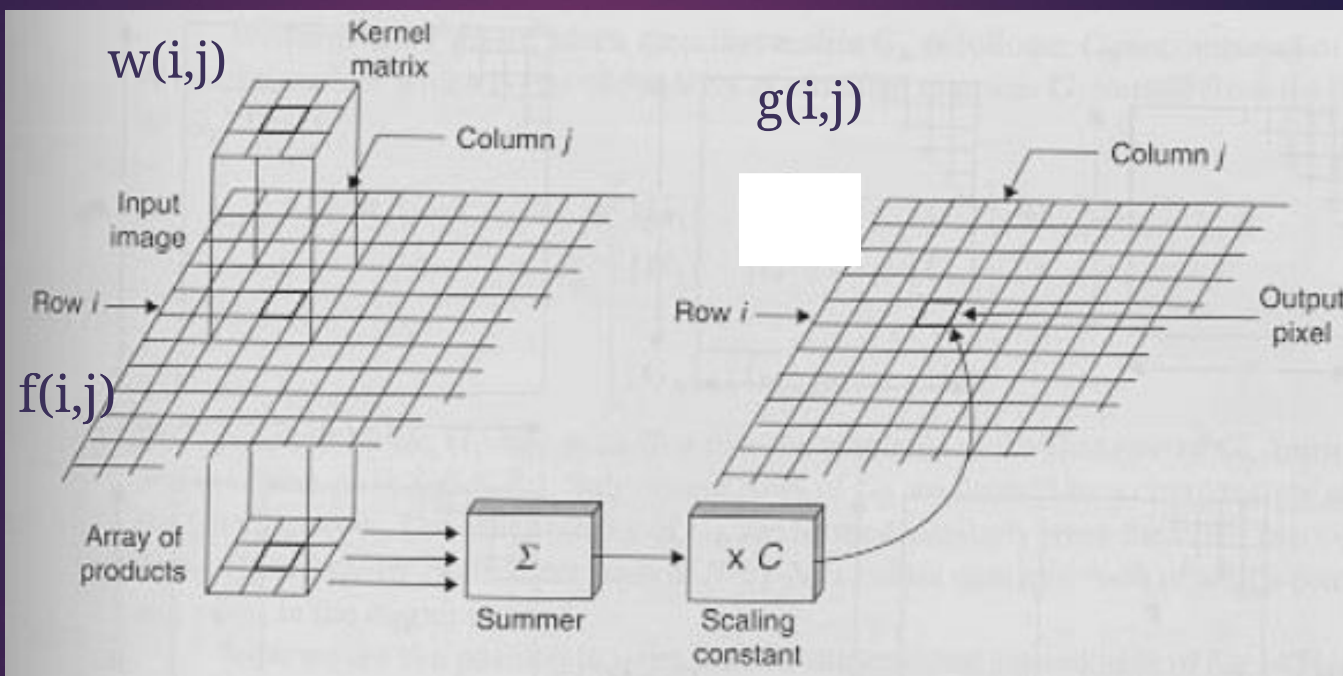
$$z5' = R = w1z1 + w2z2 + \ldots + z9w9$$

- Methods that do not satisfy the above property are called non-linear.
  - e.g., $z'_5 = max(z_k,\ k = 1, 2, \ldots, 9)$

Linear Spatial Filtering Methods

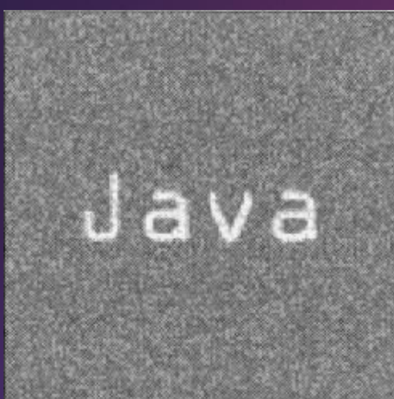
- Two main linear spatial filtering methods:
  - –Correlation
  - –Convolution

Correlation



$g(x, y) \top\ h(x, y) \cdot f(x, y)$

Correlation (cont'd)



Often used in applications where we need to measure the similarity between images or parts of images (e.g., pattern matching).

Convolution

- **Convolution is a mathematical way of combining two signals to form a third signal.**
- **It is the single most important technique in Digital Signal Processing.**
- **Using the strategy of impulse decomposition, systems are described by a signal called the *impulse response*.**
- **Convolution is important because it relates the three signals of interest: the input signal, the output signal, and the impulse response.**

Convolution

- Similar to correlation except that the mask is first <u>**flipped**</u> both horizontally and vertically.

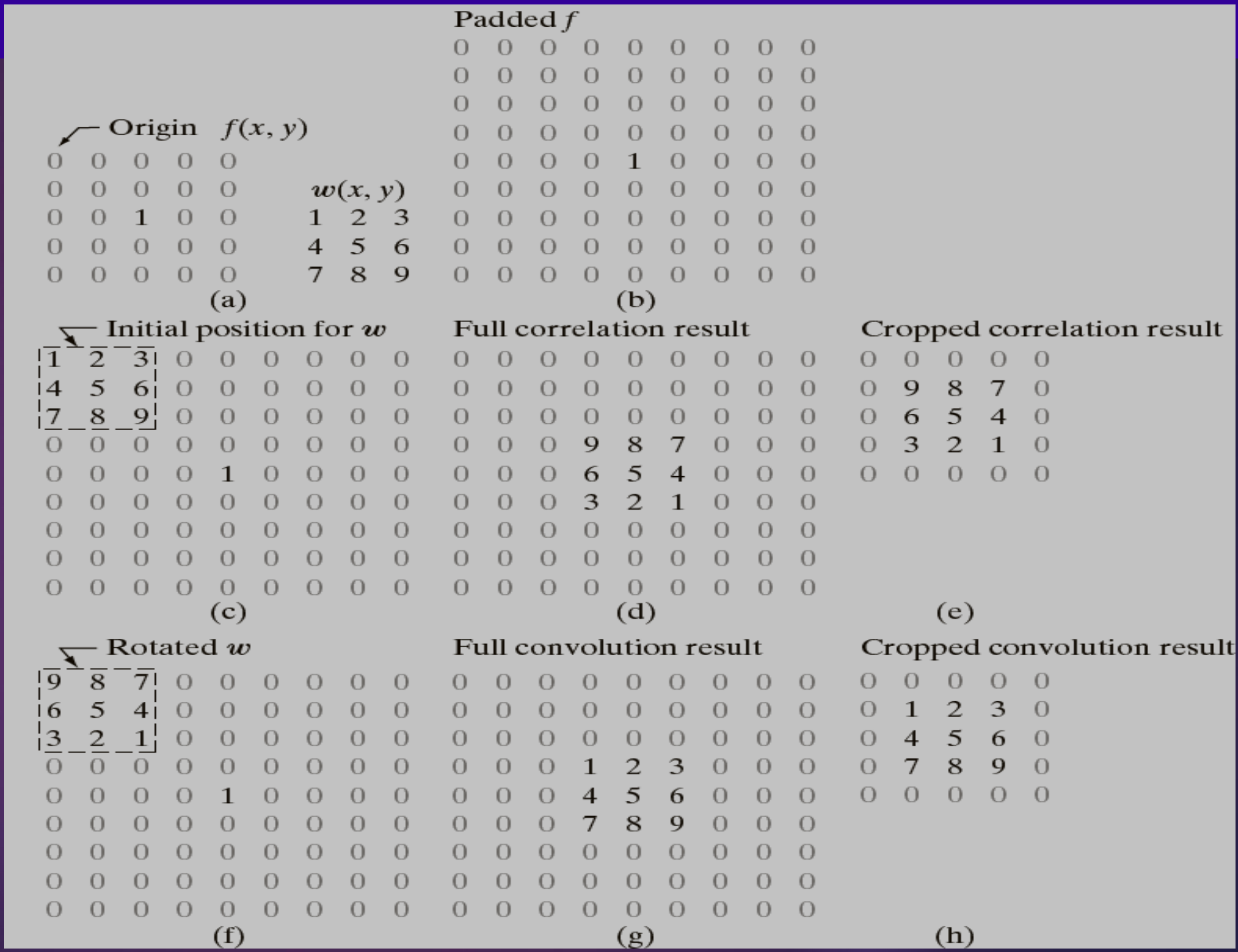
$$g(x,y) = w(x,y) * f(x,y) = \sum_{s=-K/2}^{K/2} \sum_{t=-K/2}^{K/2} w(s,t)\, f(x-s, y-t)$$

<u>Note</u>: if w(x,y) is symmetric, that is w(x,y)=w(-x,-y), then convolution is equivalent to correlation!

Example

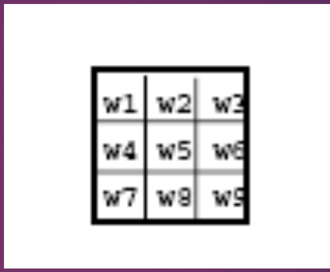
Correlation:

Convolution:



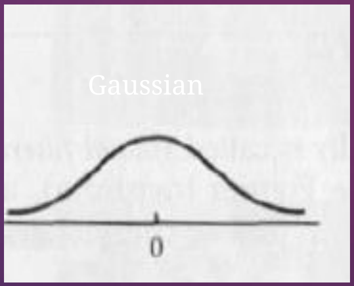
How do we choose the elements of a mask?

- Typically, by sampling certain functions.



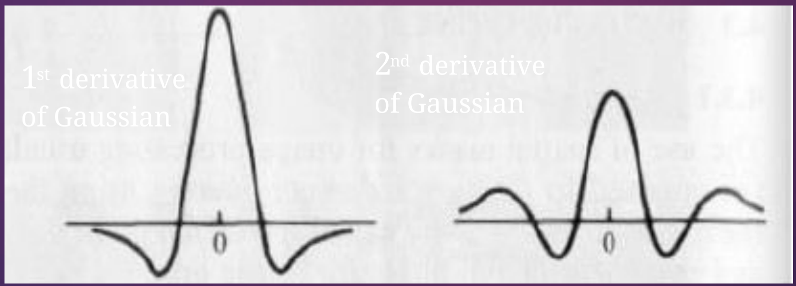
Filters

- Smoothing (i.e., low-pass filters)
  - –Reduce noise and small details.
  - –The elements of the mask must be positive.
  - –Sum of mask elements is 1



Filters

- Sharpening (i.e., high-pass filters)
  - –Highlight fine detail or enhance detail that has been blurred.
  - –The elements of the mask contain both positive and negative weights.
  - –Sum of the mask weights is 0



Smoothening Filters (Low Pass Filter)

- **Averaging Filter**
- **Median Filter**
- **Guassian Filter**

Linear Filter

- In linear filter, output values are the linear combinations of the pixels in the original image.
- Linear methods are far more amenable to mathematical analysis than are nonlinear ones, and are consequently far better understood.
- For example, if a linear filter is applied to the output from another linear filter, then the result is a third linear filter.

Linear Filters

- **Linear Filters**
- **GaussianFilter** — Gaussian and Gaussian derivatives filtering of images and arrays
- **DerivativeFilter** — general-order derivative filter
- **MeanFilter** ▪ **GradientFilter** ▪ **LaplacianFilter** ▪ **WienerFilter** ▪ **Moving Average**
- **LowpassFilter** ▪ **HighpassFilter** ▪ **BandpassFilte r** ▪ **BandstopFilter** ▪ **HilbertFilter** ▪ **Differentiator Filter**
- **ListConvolve**, **ListCorrelate** — convolve, correlate with any kernel

Non Linear Filters

- **Median Filter**
- **Min Filter**
- **Max Filter**
- **MeanShift Filter**
- **Entropy Filter**
- **Corner Filter**
- **Ridge Filter**
- **Kuwahara Filter**
- **Bilateral Filter**

Smoothing Filters: Averaging (Low-pass filtering)

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

(a)

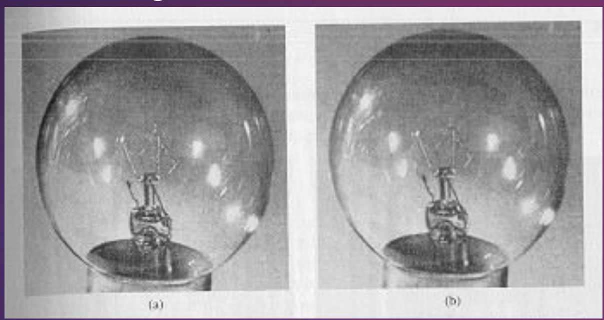
$$\frac{1}{25} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\frac{1}{49} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

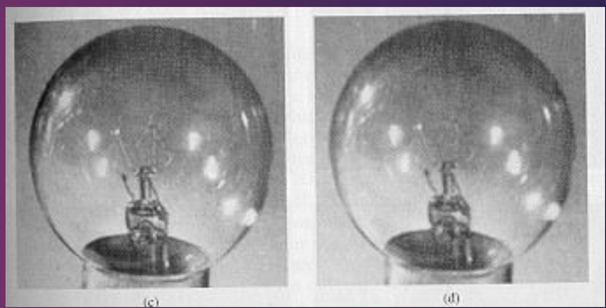
Smoothing Filters: Averaging (cont'd)

- Mask size determines the degree of smoothing and loss of detail.

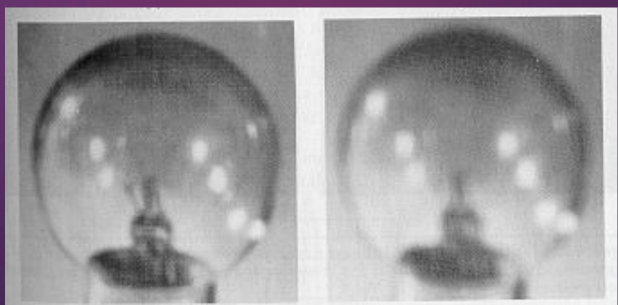
original 3x3



5x5 7x7



15x15 25x25

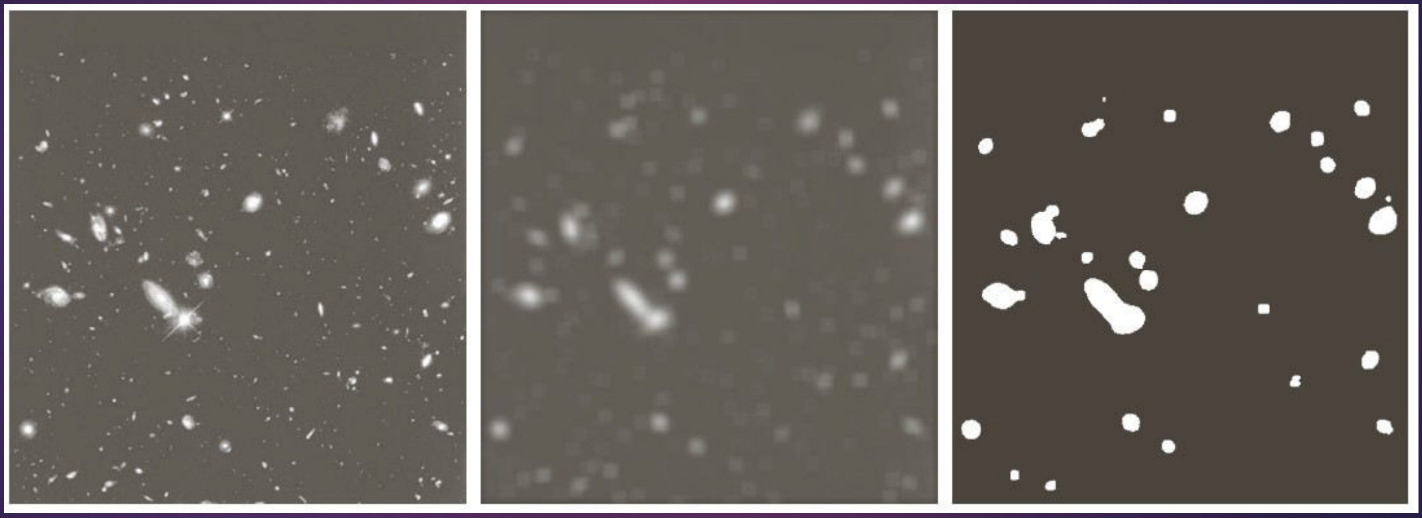


Smoothing Filters: Averaging (cont'd)

Example: extract, largest, brightest objects

15 x 15 averaging

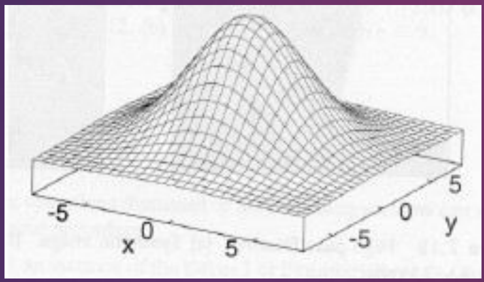
image thresholding



Smoothing filters: Gaussian

- The weights are samples of the 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

σ = 1.4

mask size: $height = width = 5\sigma$ (subtends 98.76% of the area)

Smoothing filters: Gaussian (cont'd)

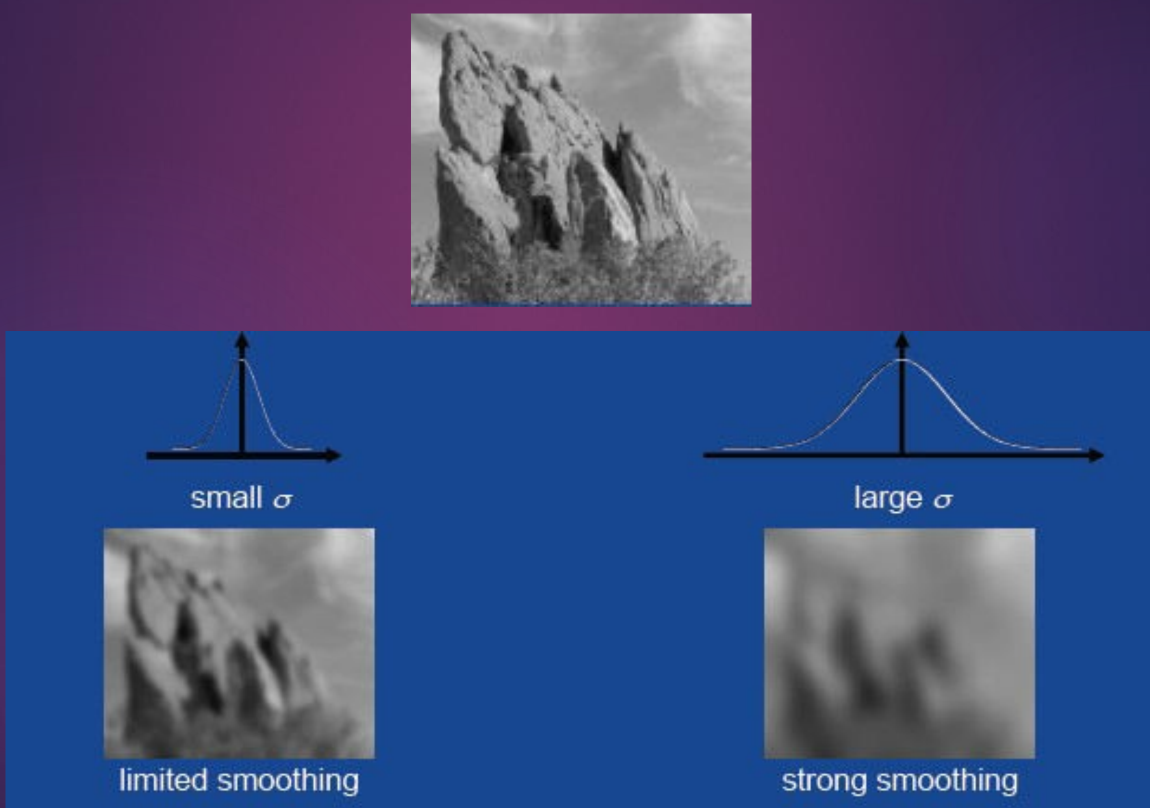
- As **σ** increases, more samples must be obtained to represent the Gaussian function accurately.
- Therefore, **σ** controls the amount of smoothing

σ = 3

15 × 15 Gaussian mask

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 (non-linear)

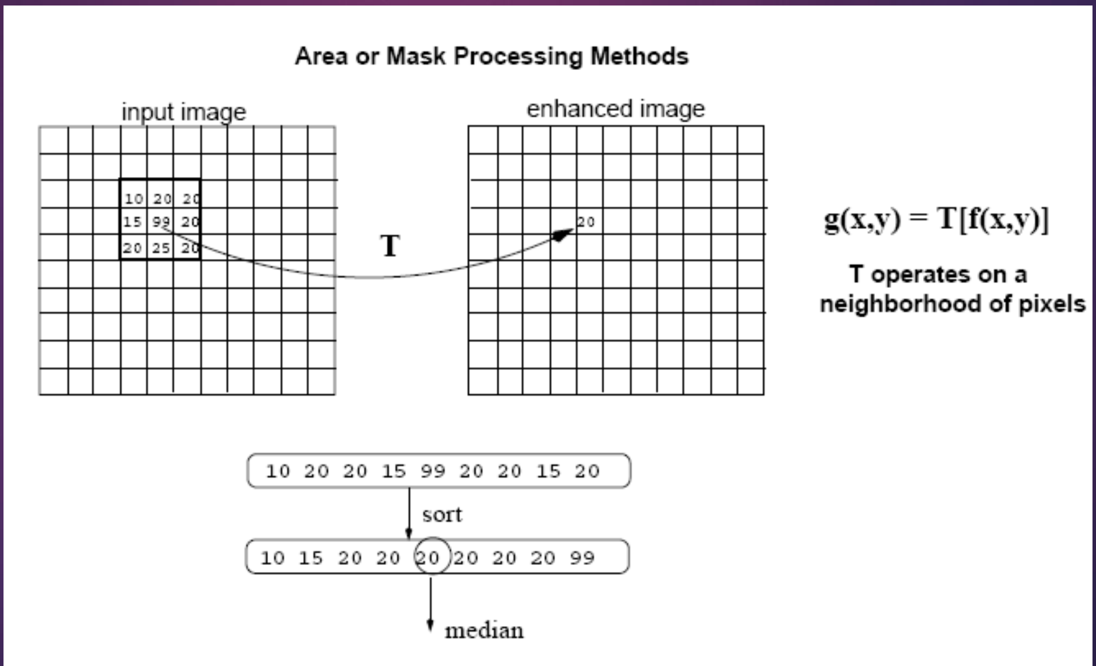
- Very effective for removing “salt and pepper” noise (i.e., random occurrences of black and white pixels).

averaging

median filtering

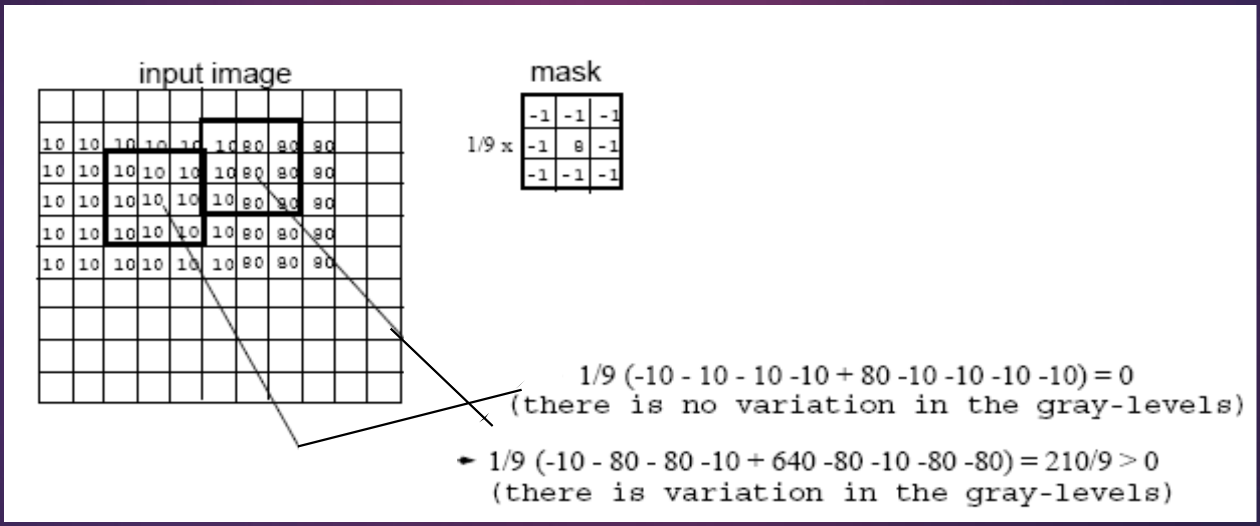
Smoothing Filters: Median Filtering (cont'd)

- Replace each pixel by the median in a neighborhood around the pixel.



Sharpening Filters (High Pass filtering)

- Useful for emphasizing transitions in image intensity (e.g., edges).



Sharpening Filters (cont'd)

- Note that the response of high-pass filtering might be negative.
- Values must be re-mapped to [0, 255]

original image

sharpened images

Unsharp Masking

- Idea: Sharpen images using low-pass filters!   How?
- 1. Blur the original image
- 2. Subtract the blurred image from the original
- 3. Add the previous result to the original.

Sharpening Filters: Unsharp Masking

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

$$\textit{Highpass} = \textit{Original} - \textit{Lowpass}$$

Sharpening Filters: High Boost

- Image sharpening emphasizes edges but details (i.e., low frequency components) might be lost.
- **High boost filter**: amplify input image, then subtract a lowpass image.

$$
\begin{aligned}
Highboost &= A\ Original - Lowpass \\
&= (A-1)\ Original + Original - Lowpass \\
&= (A-1)\ Original + Highpass
\end{aligned}
$$

(A-1)

Sharpening Filters: Unsharp Masking (cont'd)

- If **A=1**, we get a high pass filter

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

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: Unsharp Masking (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 can be computed using the gradient.

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

Sharpening Filters: Derivatives (cont'd)

- The gradient is a **vector** which has magnitude and direction:

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

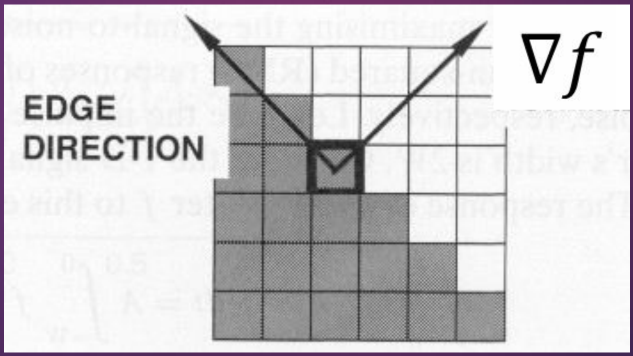
$$direction(grad(f)) = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

or $\left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right|$

(approximation)

Sharpening Filters: Derivatives (cont'd)

- **Magnitude:** provides information about edge strength.
- **Direction:** perpendicular to the direction of the edge.



Sharpening Filters: Gradient Computation

- Approximate gradient using finite differences.

$$f'(x) = \lim_{h->0} \frac{f(x+h)-f(x)}{h} \approx f(x+1)-f(x) \quad (h=1)$$

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

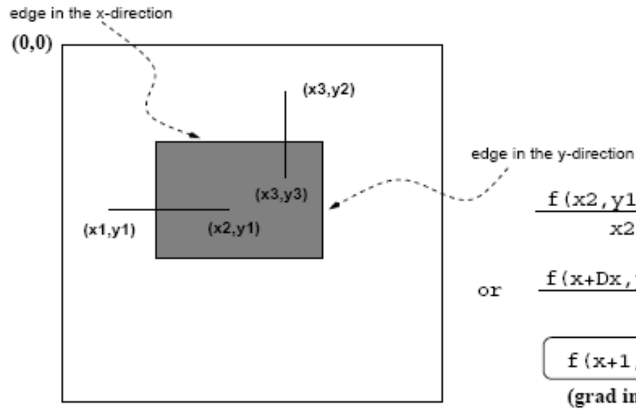
$$\frac{\partial f}{\partial y} = \frac{f(x, y+\Delta y)-f(x,y)}{-\Delta y} = f(x,y) - f(x,y+1), \quad (\Delta y=1)$$

Sharpening Filters: Gradient Computation (cont'd)

$$\frac{f(x3,y2) - f(x3,y3)}{y2 - y3}$$

or $$\frac{f(x,y+Dy) - f(x,y)}{-Dy} = \boxed{f(x,y) - f(x,y+1)}$$ **(grad in the y-direction)**

(y3=y2+Dy, y2=y, x3=x, Dy=1)



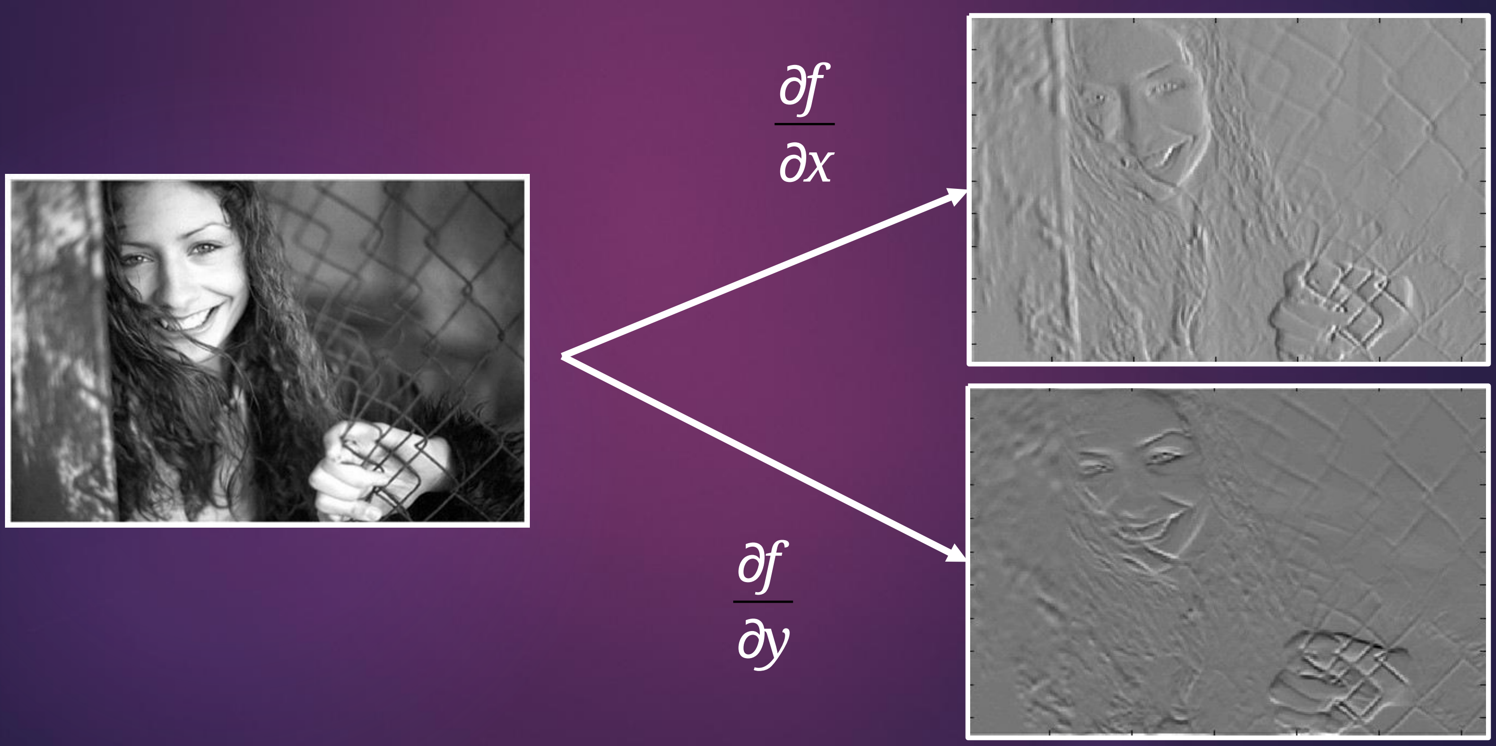
$$\frac{f(x2,y1) - f(x1,y1)}{x2 - x1}$$

or $$\frac{f(x+Dx,y) - f(x,y)}{Dx} = \boxed{f(x+1,y) - f(x,y)}$$

(grad in the x-direction)

(x2=x+Dx, x1=x, y1=y2=y, Dx=1)

Example



Sharpening Filters: Gradient Computation (cont'd)

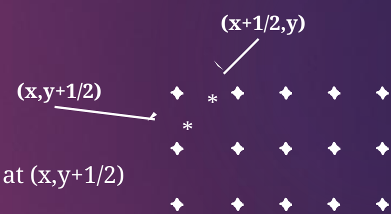
- We can implementa $\frac{\partial f}{\partial x}$ using t $\frac{\partial f}{\partial y}$ masks:

-1	1

1
-1

$\frac{\partial f}{\partial x}$ good approximation at (x+1/2,y)

$\frac{\partial f}{\partial y}$ good approximation at (x,y+1/2)



- Example: approximate gradient at z_5

Z1	Z2	Z3
Z4	Z5	Z6
Z7	Z8	Z9

$$\frac{\partial f}{\partial x} = z_6 - z_5$$

$$\frac{\partial f}{\partial y} = z_5 - z_8$$

$$mag(grad(f)) = \sqrt{(z_6 - z_5)^2 + (z_5 - z_8)^2}$$

Sharpening Filters: Gradient Computation (cont'd)

- A different approximation of the gradient:

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

good approximation

(x+1/2,y+1/2)

- We can implement $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ using the following masks:

1	0
0	–1

0	1
–1	0

(b) Roberts

Sharpening Filters: Gradient Computation (cont'd)

- Example: approximate gradient at z_5

Z1	Z2	Z3
Z4	Z5	Z6
Z7	Z8	Z9

$$\frac{\partial f}{\partial x} = z_5 - z_9$$

$$\frac{\partial f}{\partial y} = z_6 - z_8$$

$$mag(grad(f)) = \sqrt{(z_5 - z_9)^2 + (z_6 - z_8)^2}$$

- Other approximations

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

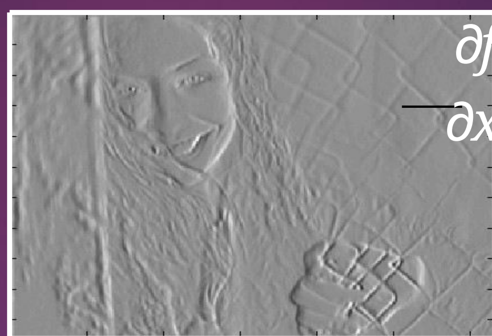
(c) Prewitt

−1	−2	−1
0	0	0
1	2	1

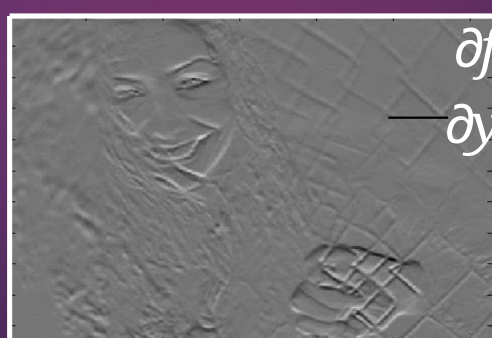
−1	0	1
−2	0	2
−1	0	1

Sobel

Example



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



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

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

Sharpening Filters: Laplacian

The Laplacian (2nd derivative)is defined as:

$$\nabla^2 = \nabla \cdot \nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

(dot product)

Approximate derivatives:

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

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

$$\nabla^2 f = -4f(i, j) + f(i, j+1) + f(i, j-1) + f(i+1, j) + f(i-1, j)$$

Sharpening Filters: Laplacian (cont'd)

Laplacian Mask

0	0	0
1	-2	1
0	0	0

+

0	1	0
0	-2	0
0	1	0

=

0	1	0
1	-4	1
0	1	0

5	5	5	5	5	5
5	5	5	5	5	5
5	5	10	10	10	10
5	5	10	10	10	10
5	5	5	10	10	10
5	5	5	5	10	10

detect zero-crossings

-	-	-	-	-	-
-	0	-5	-5	-5	-
-	-5	10	5	5	-
-	-5	10	0	0	-
-	0	-10	10	0	-
-	-	-	-	-	-

Sharpening Filters: Laplacian (cont'd)

Laplacian

Sobel