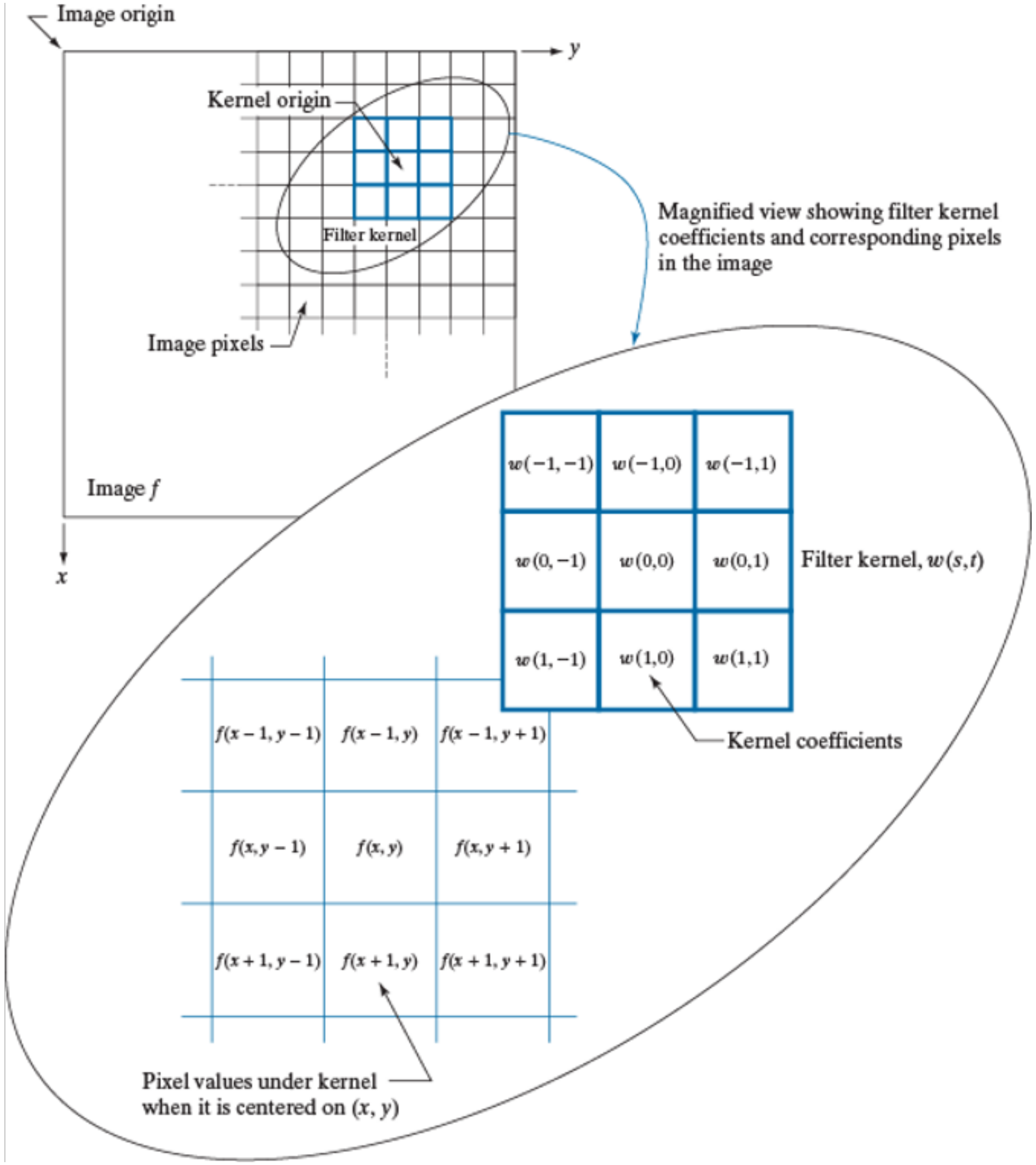
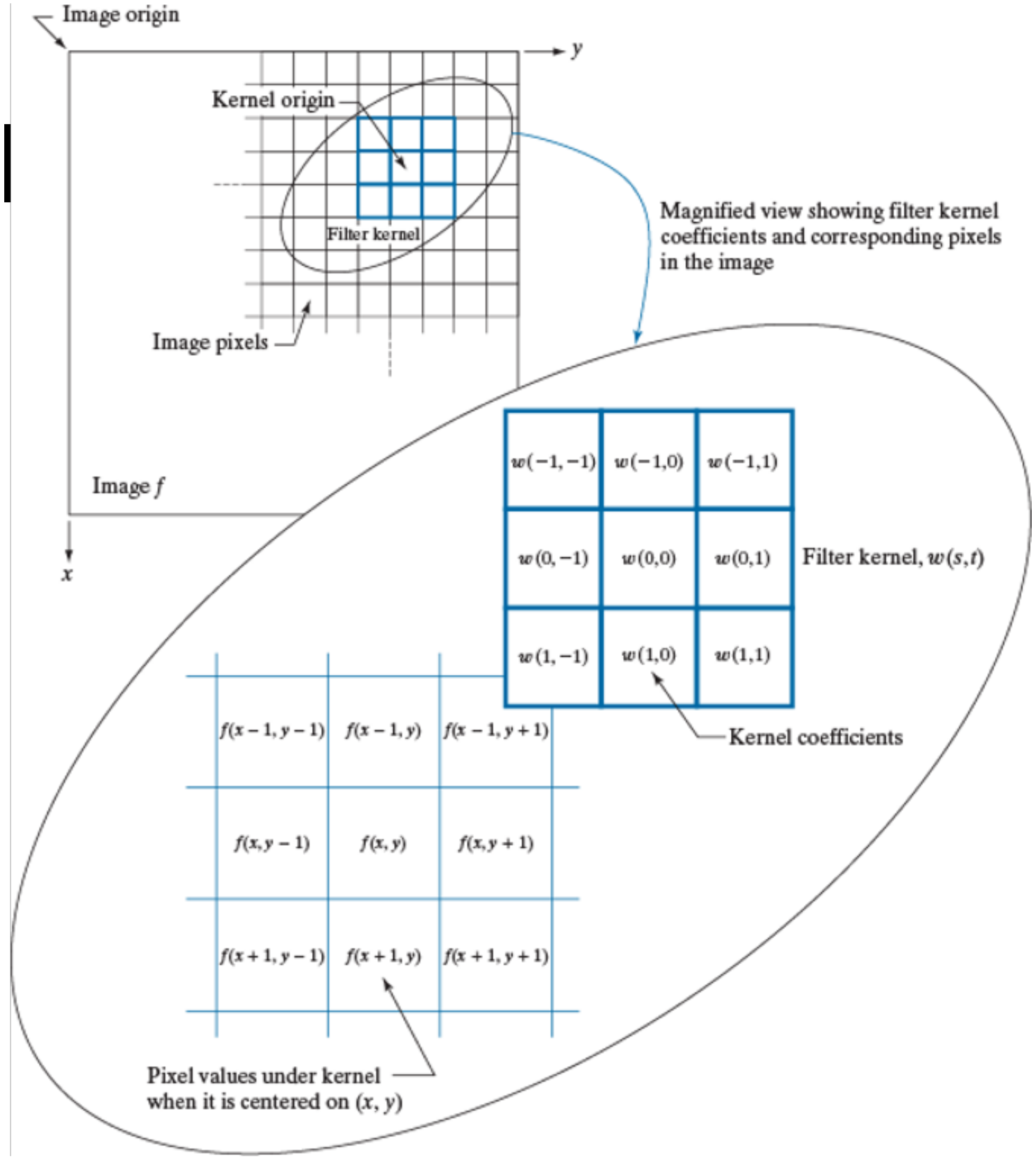
3.4 Fundamentals of Spatial Filtering

- Filter, Mask, Kernel, Template, Window
- Coefficients
- Linear Filtering vs Nonlinear Filtering (e.g., median filtering)



3.4 Fundamental

- Filter, Mask, Kernel, Template, Window
- Coefficients
- Linear Filtering vs Nonlinear Filtering (e.g., median filtering)



$$g(x,y) = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + \ldots$$
$$+ w(0,0)f(x,y) + \ldots + w(1,1)f(x+1,y+1)$$

3.4 Fundamentals of Spatial Filtering - Linear Spatial Filtering

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

- The center coefficient of the kernel, w(0, 0), aligns with the pixel at location (x, y).

- For a kernel of size m x n, we assume that m = 2a + 1 and n = 2b + 1, where a and b are nonnegative integers.

3.4 Fundamentals of Spatial Filtering - Linear Spatial Filtering

One of the simplest spatial filtering for smoothing operation - Averaging Filter

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

Box Filter

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

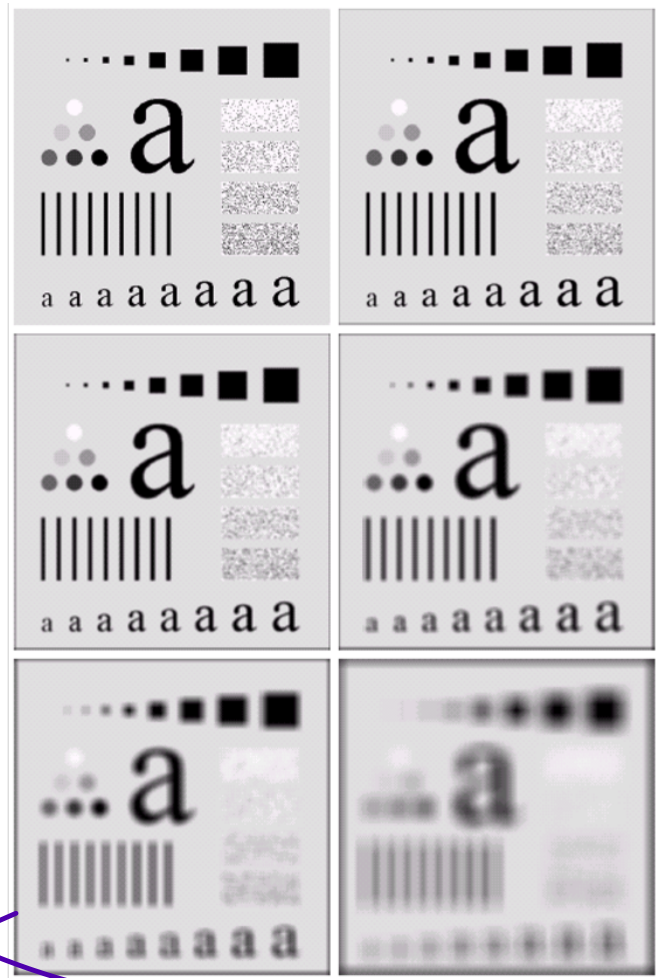
Weighted Average

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

3.4 Fundamentals of Spatial Filtering - Linear Spatial Filtering

One of the simplest spatial filtering for smoothing operation - Averaging Filter

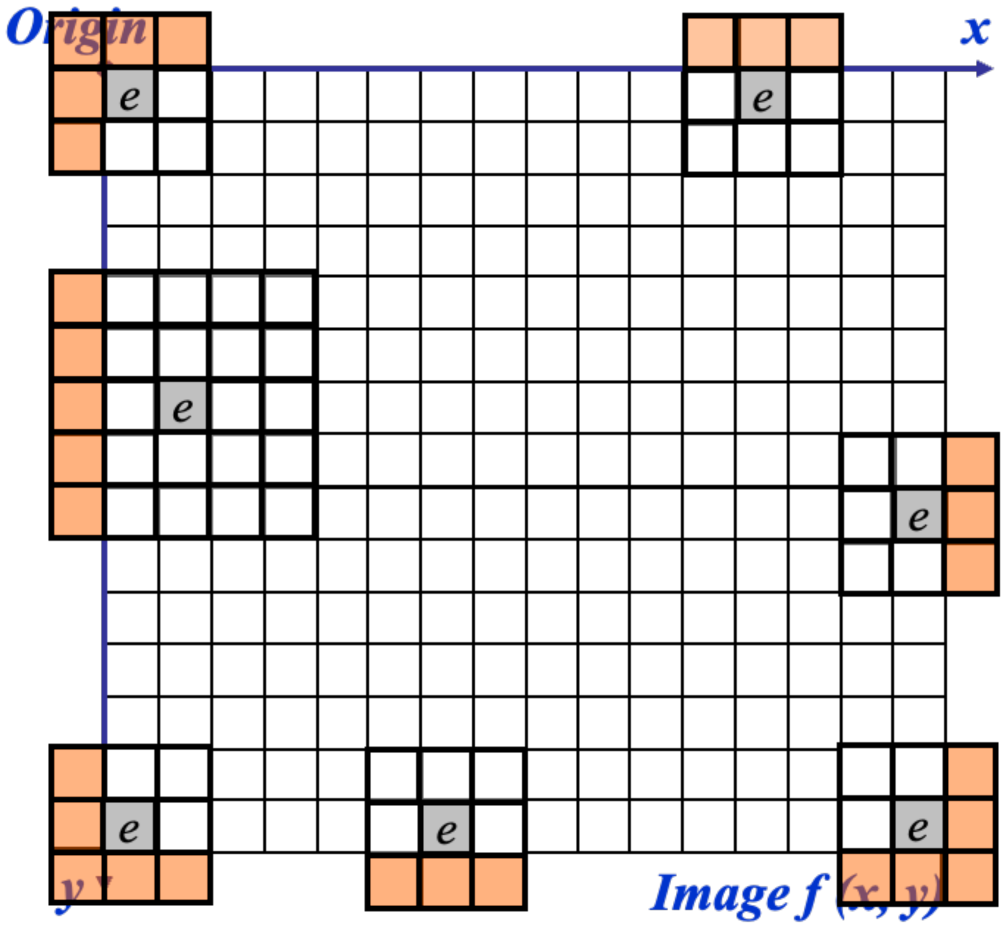
- original image of size 500*500 pixels
- filtering with an averaging filter of increasing sizes – 3, 5, 9, 15 and 35



Zero padding for edges

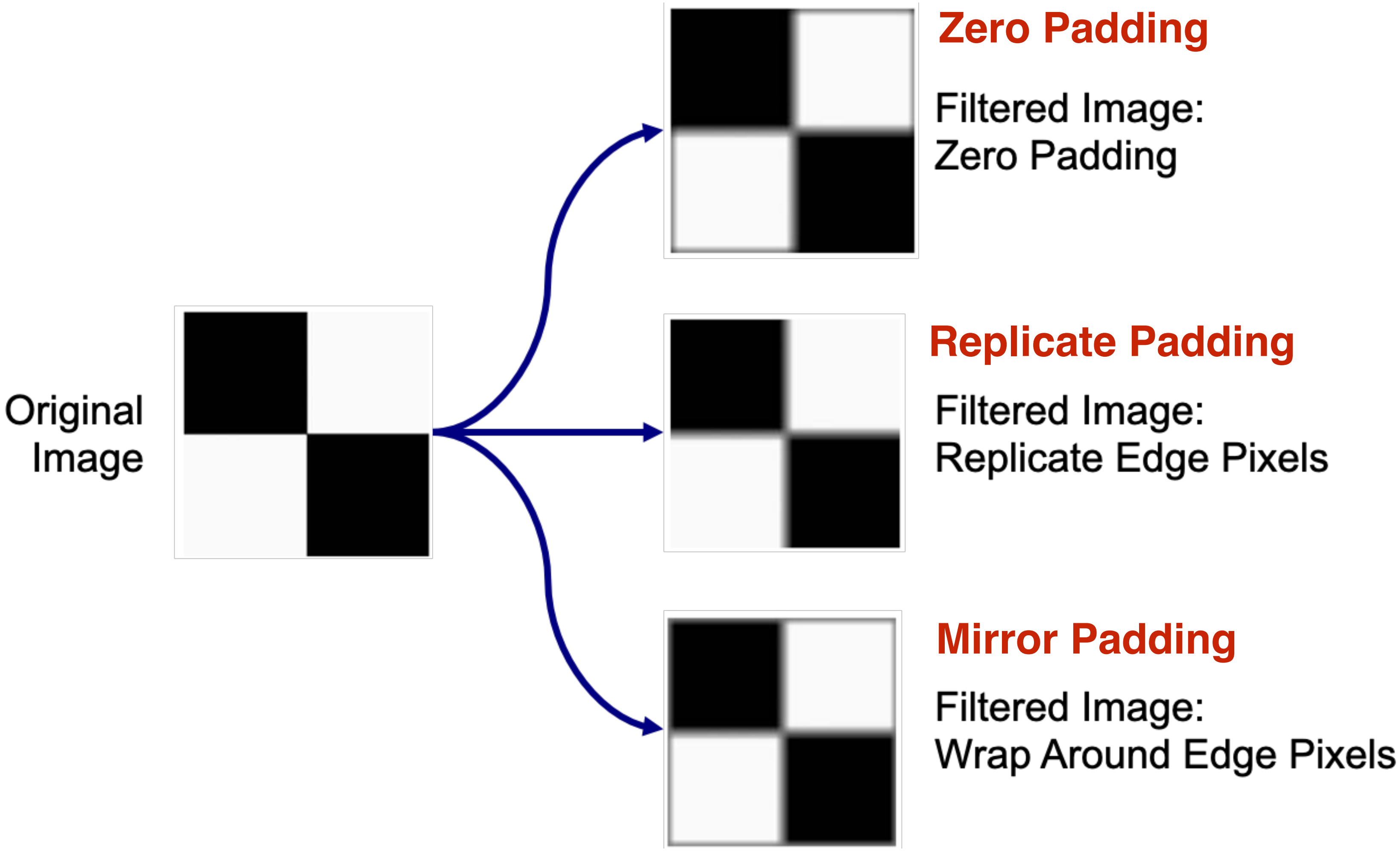
3.4 Fundamentals of Spatial Filtering - Spatial Filtering at the Edges

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



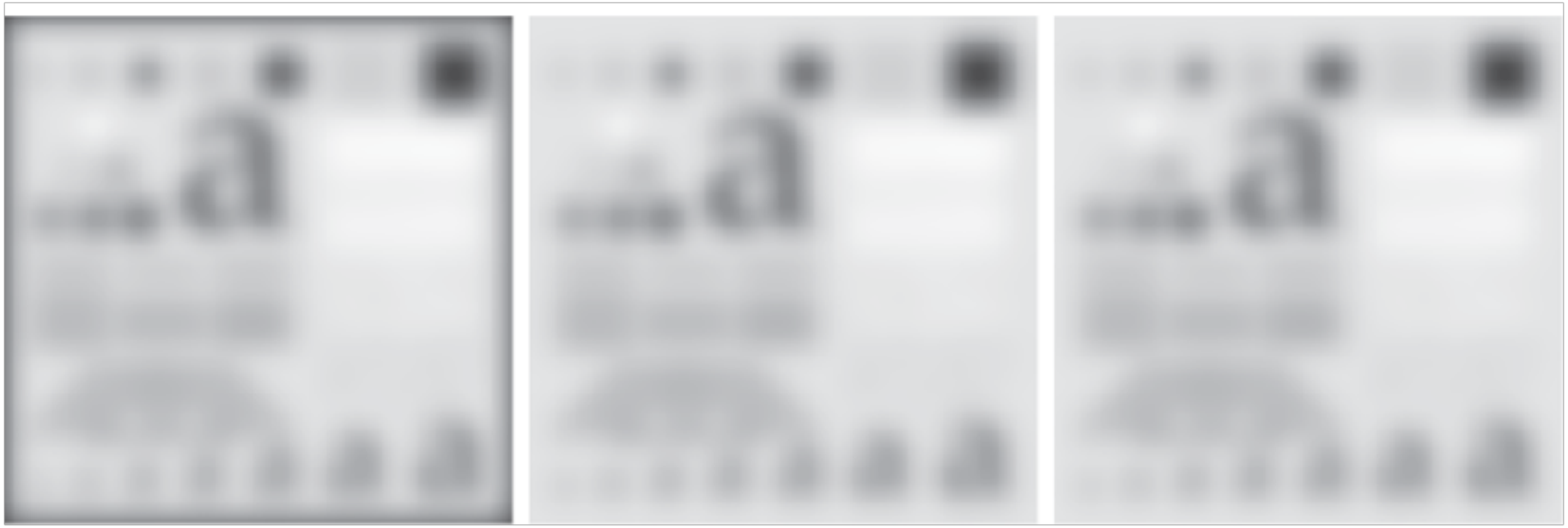
credit of this slide: C. Nikou

3.4 Fundamentals of Spatial Filtering
- Spatial Filtering at the Edges



credit of this slide: C. Nikou

3.4 Fundamentals of Spatial Filtering - Spatial Filtering at the Edges



Zero Padding | **Mirror Padding** | **Replicate Padding**

Mirror Padding: useful when the areas near the border of the image are constant

Replicate Padding: more applicable when the areas near the border contain image details

3.4 Fundamentals of Spatial Filtering - Spatial Correlation and Convolution

- The filtering we have been talking about so far is referred to as **correlation** with the filter itself referred to as the **correlation kernel**

- The mechanics of spatial **convolution** are the same, **except that the correlation kernel is rotated by 180°**.

- For symmetric filters it makes no difference.

credit of this slide: C. Nikou

3.4 Fundamentals of Spatial Filtering - Spatial Correlation and Convolution

Correlation

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

Convolution

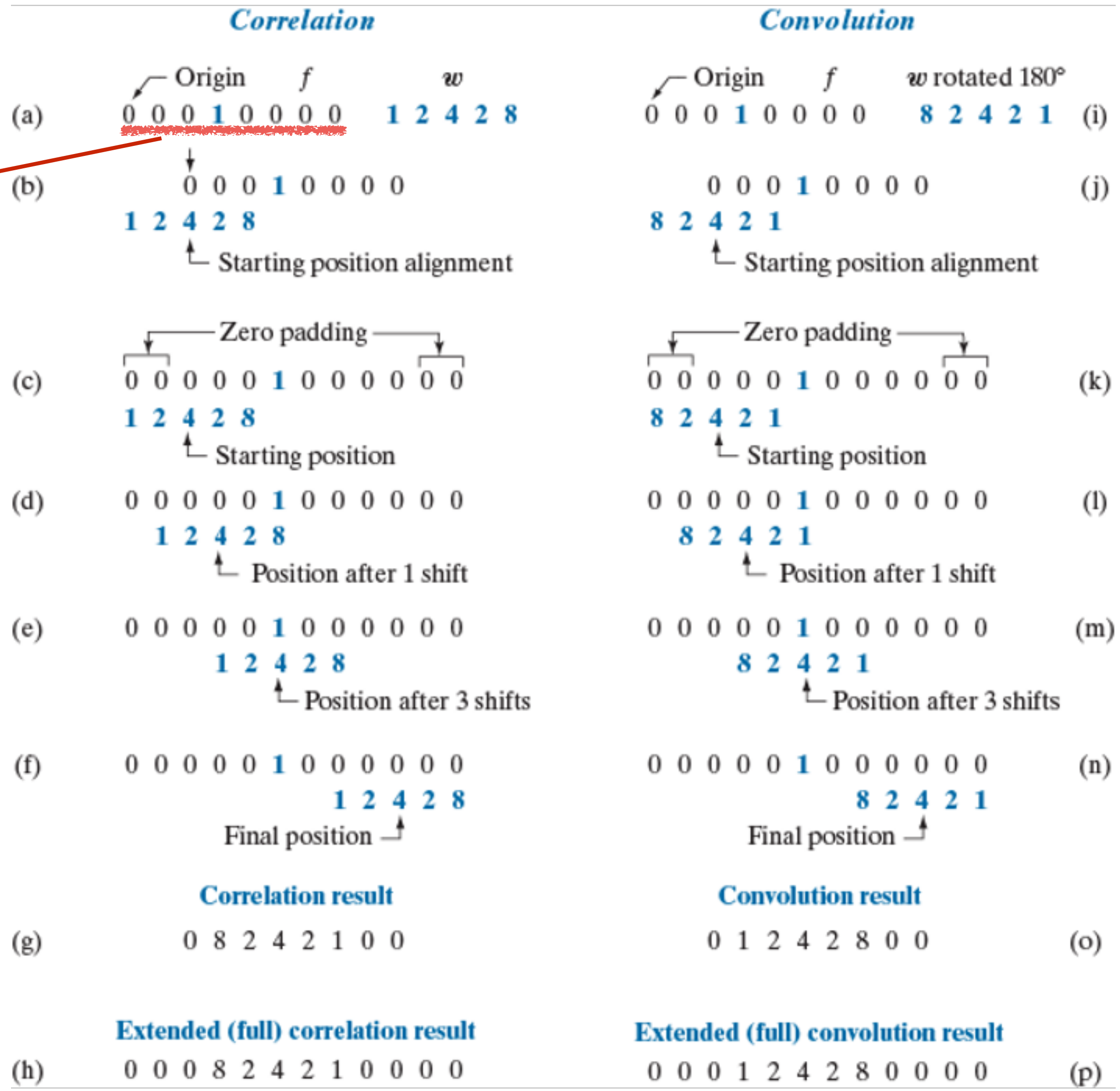
$$(w \bigstar f)(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t)$$

3.4 Fundamentals of Spatial Filtering - Spatial Correlation and Convolution

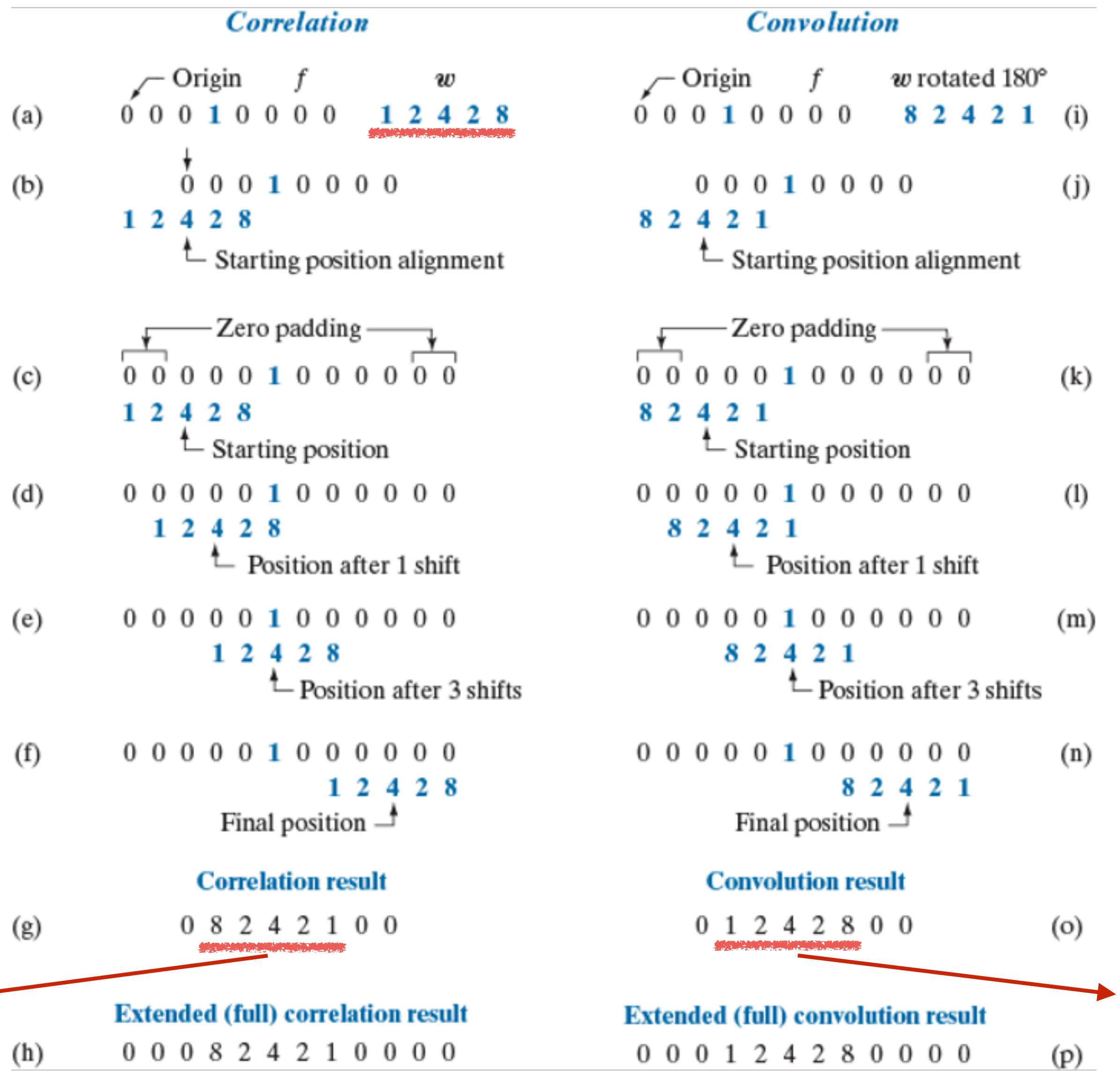
discrete unit impulse
- A function that contains a single 1 with the rest being 0's

discrete impulse of strength (amplitude) A

$$\delta(x-x_0, y-y_0) = \begin{cases} A & \text{if } x = x_0 \text{ and } y = y_0 \\ 0 & \text{otherwise} \end{cases}$$



3.4 Fundamentals of Spatial Filtering
- Spatial Correlation and Convolution



Correlation

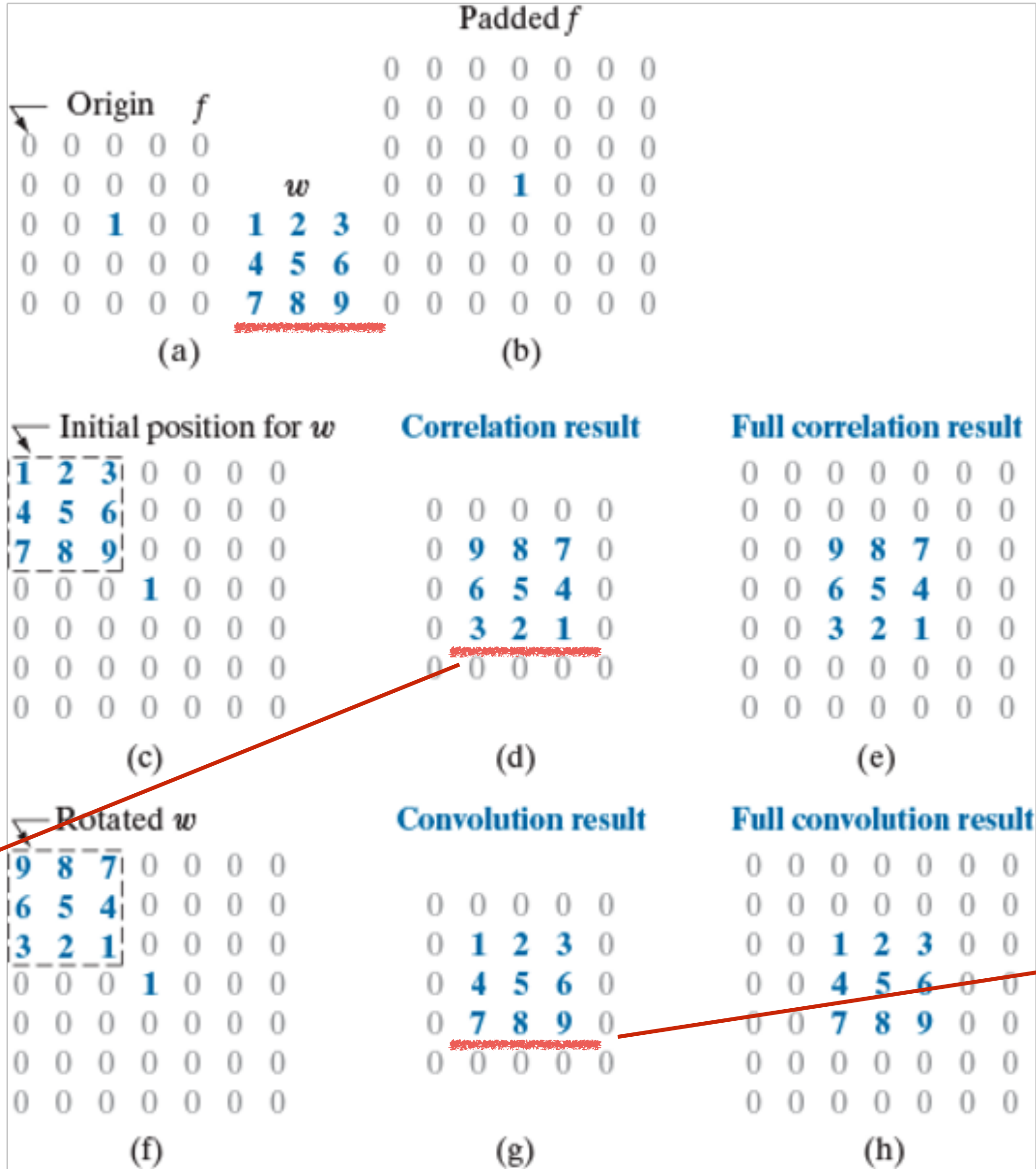
rotated version of the kernel at the location of the impulse

Convolution

exactly copy of the kernel at the location of the impulse

3.4 Fundamentals of Spatial Filtering - Spatial Correlation and Convolution

$\delta(x-2, y-2)$



Correlation

rotated version of the kernel at the location of the impulse

Convolution

exactly copy of the kernel at the location of the impulse

3.4 Fundamentals of Spatial Filtering - Spatial Correlation and Convolution

Correlation

rotated version of the kernel at the location of the impulse

Convolution

exactly copy of the kernel at the location of the impulse

TABLE 3.5 Some fundamental properties of convolution and correlation. A dash means that the property does not hold.

Property	Convolution	Correlation
Commutative	$f \star g = g \star f$	—
Associative	$f \star (g \star h) = (f \star g) \star h$	—
Distributive	$f \star (g + h) = (f \star g) + (f \star h)$	$f \stackrel{\star}{\smile}(g + h) = (f \stackrel{\star}{\smile} g) + (f \stackrel{\star}{\smile} h)$

That's why convolution is more widely used.

- When talking about filtering and kernels, you are likely to encounter the terms ***convolution filter***, ***convolution mask***, or ***convolution kernel*** to denote filter kernels.
- In this book, when we use the term *linear spatial filtering*, we mean *convolving a kernel with an image*. **Not** to imply necessarily that the kernel is used for convolution.

3.4 Fundamentals of Spatial Filtering - Separable Filter Kernels

TABLE 3.5
Some fundamental properties of convolution and correlation. A dash means that the property does not hold.

Property	**Convolution**	**Correlation**
Commutative	$f \star g = g \star f$	—
Associative	$f \star (g \star h) = (f \star g) \star h$	—
Distributive	$f \star (g + h) = (f \star g) + (f \star h)$	$f \,☆\, (g + h) = (f \,☆\, g) + (f \,☆\, h)$

$$\underbrace{w}_{m \times n \text{ kernel}} = \underbrace{\mathbf{v}}_{m \times 1 \text{ vector}} \underbrace{\mathbf{w}^T}_{1 \times n \text{ vector}}$$

$$w \star f = (w_1 \star w_2) \star f = (w_2 \star w_1) \star f = w_2 \star (w_1 \star f) = (w_1 \star f) \star w_2$$

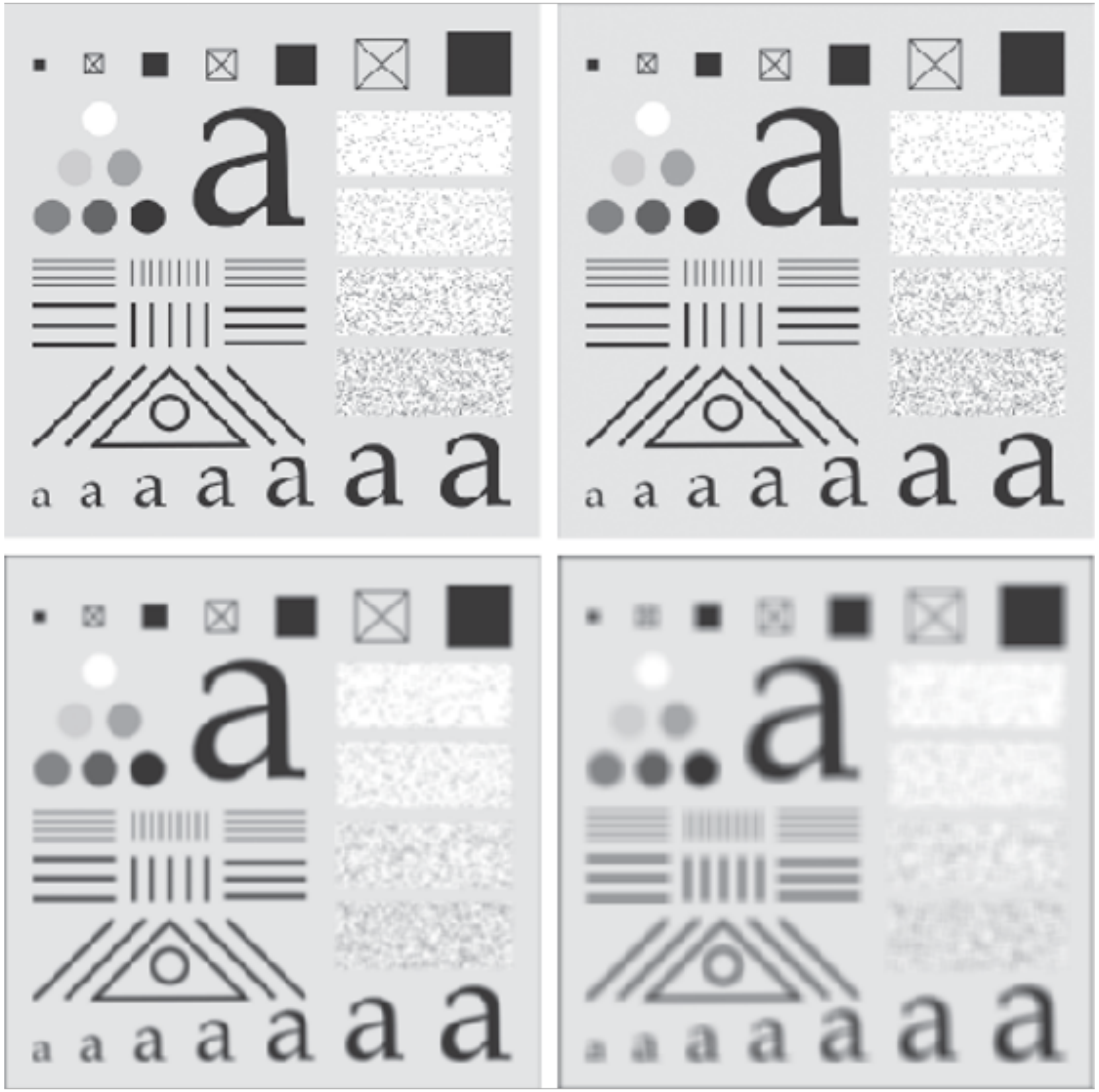
Image size *M* x *N* (including padding)

Directly calculate: **O(*MNmn*)**
If separable kernel: **O(*MN*(*m*+*n*))** (*MNm* + *MNn*)

3.5 Smoothing (Lowpass) Spatial Filters - Box Filter Kernels

a b
c d

FIGURE 3.33
(a) Test pattern of size 1024×1024 pixels.
(b)-(d) Results of lowpass filtering with box kernels of sizes 3×3, 11×11, and 21×21, respectively.



3x3

11x11

21x21

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

Box Filter

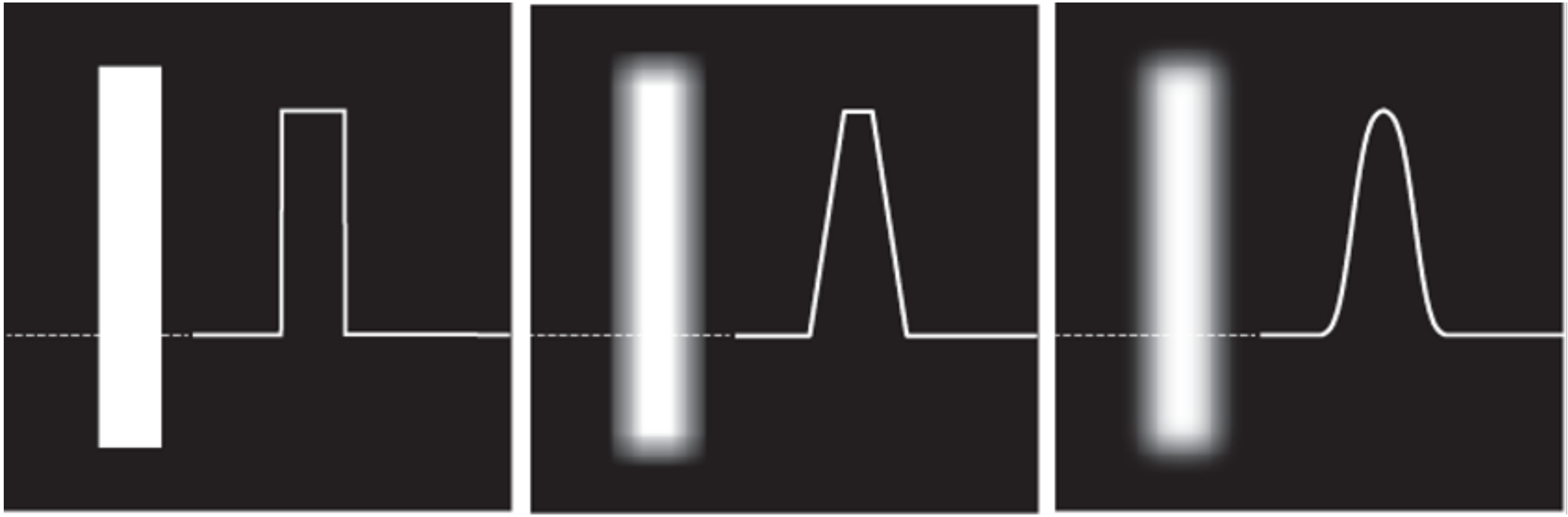
3.5 Smoothing (Lowpass) Spatial Filters - Box Filter Kernels

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

Box Filter

- Limitation of box filters:
  - poor choices in many applications, ex. defocused lens
  - box filters favor blurring along perpendicular directions

3.5 Smoothing (Lowpass) Spatial Filters - Box Filter Kernels



Box kernel
71x71

Gaussian kernel
151x151,
with $K = 1$, $\sigma = 25$

hard transitions at the onset and end of the ramp

significantly smoother results around the edge transitions

3.5 Smoothing (Lowpass) Spatial Filters - Lowpass Gaussian Filter Kernels

Gaussian Kernel:

$$w(s,t) = G(s,t) = Ke^{-\frac{s^2 + t^2}{2\sigma^2}}$$

the **only** *circularly symmetric* kernels that are also **separable**.

(also called *isotropic*, meaning their response is independent of orientation)

Let $r = [s^2 + t^2]^{1/2}$

$$G(r) = Ke^{-\frac{r^2}{2\sigma^2}}$$

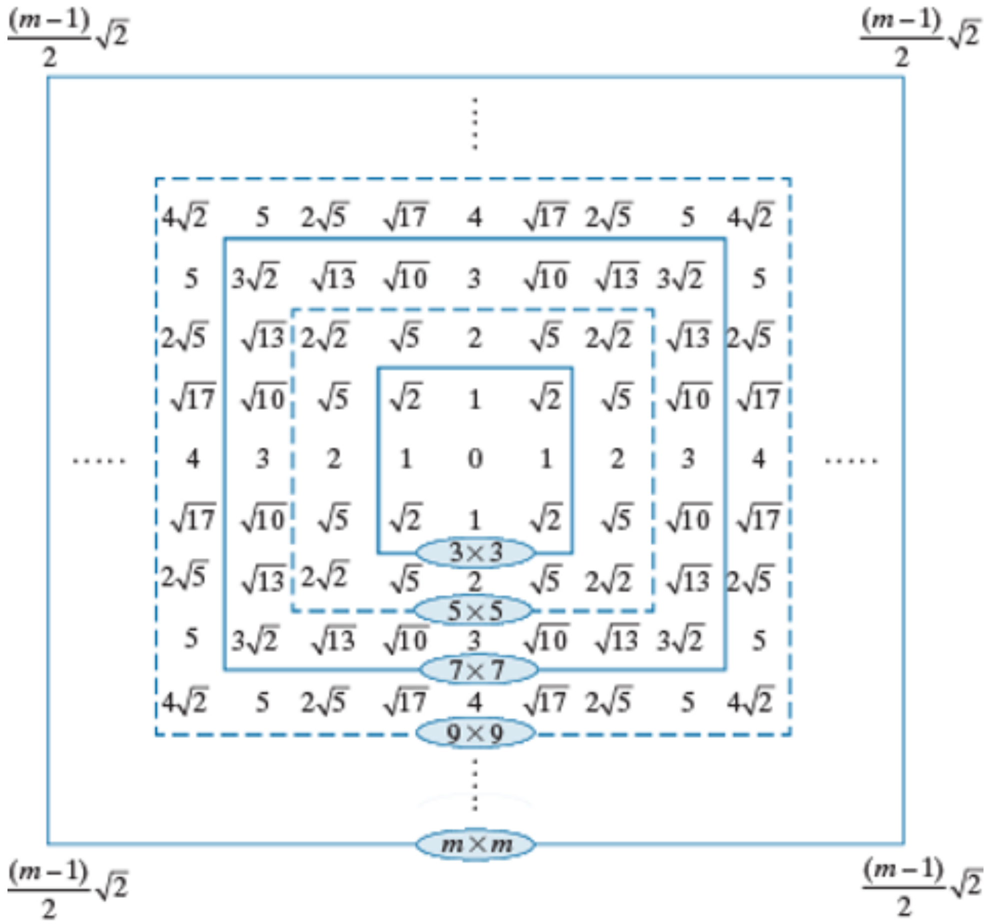
3.5 Smoothing (Lowpass) Spatial Filters - Lowpass Gaussian Filter Kernels

Let $r = [s^2 + t^2]^{1/2}$

$$G(r) = Ke^{-\frac{r^2}{2\sigma^2}}$$

distance r

$$r_{\max}^2 = \left[\frac{(m-1)}{2}\sqrt{2}\right]^2 = \frac{(m-1)^2}{2}$$

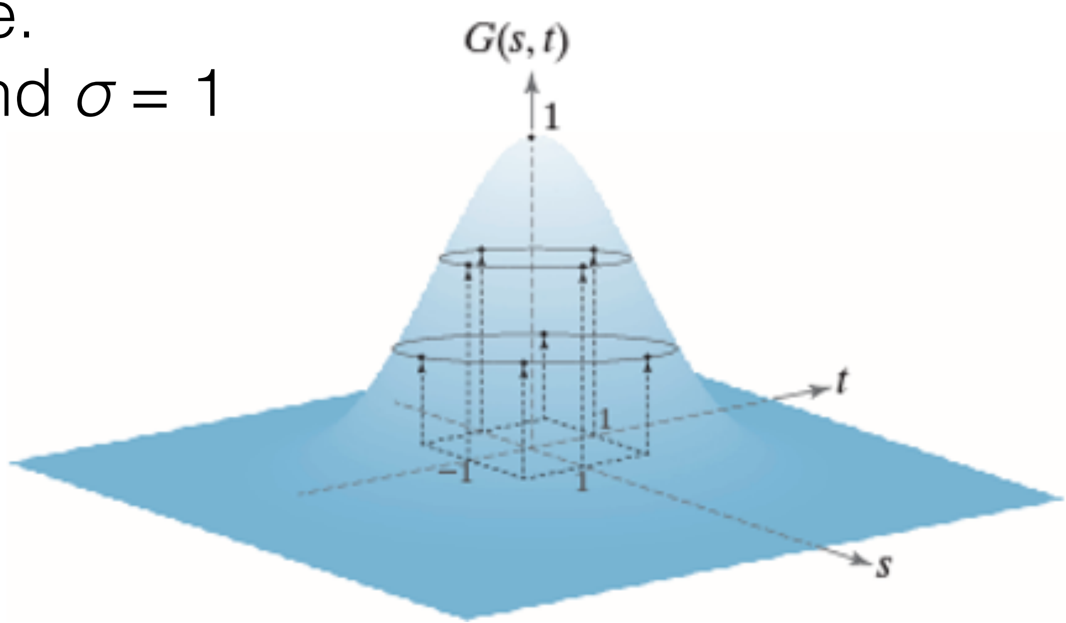


3.5 Smoothing (Lowpass) Spatial Filters - Lowpass Gaussian Filter Kernels

Let $r = [s^2 + t^2]^{1/2}$

$$G(r) = Ke^{-\frac{r^2}{2\sigma^2}}$$

Example.
$K = 1$ and $\sigma = 1$



$\frac{1}{4.8976} \times$

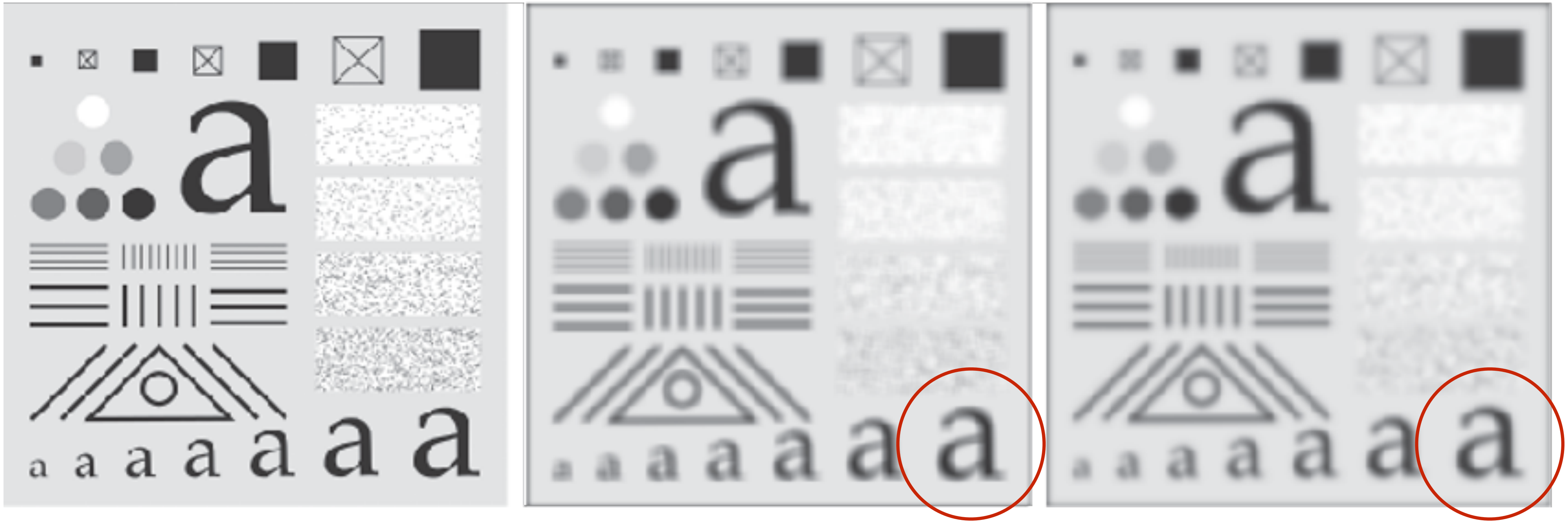
0.3679	0.6065	0.3679
0.6065	1.0000	0.6065
0.3679	0.6065	0.3679

3.5 Smoothing (Lowpass) Spatial Filters - Lowpass Gaussian Filter Kernels

Gaussian Kernel
size 21x21
$K = 1$ and $\sigma = 3.5$

Gaussian Kernel
size 43x43
$K = 1$ and $\sigma = 7$

3.5 Smoothing (Lowpass) Spatial Filters - Lowpass Gaussian Filter Kernels



Comparison

Box filter
size 21x21

Gaussian Kernel
size 43x43
$K = 1$ and $\sigma = 7$

much smoother around the edges

3.5 Smoothing (Lowpass) Spatial Filters - Lowpass Gaussian Filter Kernels

Some important **properties** of Gaussian kernels:

- **Separable**
- The meaningful **largest kernel size** for image processing is **6σ x 6σ** (e.g., a 43 x 43 kernel if $\sigma = 7$)
  - because the values of a Gaussian function at a distance larger than 3σ from the mean are small enough that they can be ignored.

Gaussian Kernel size 43x43, $\sigma = 7$

Gaussian Kernel size 85x85, $\sigma = 7$

Difference image

3.5 Smoothing (Lowpass) Spatial Filters - Lowpass Gaussian Filter Kernels

Some important **properties** of Gaussian kernels:

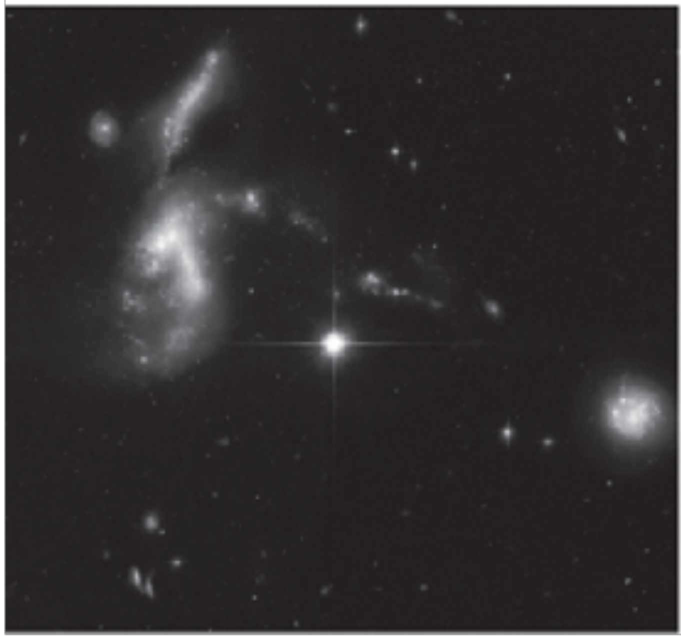
- **Separable**
- The meaningful **largest kernel size** for image processing is **6*σ* x 6*σ*** (e.g., a 43 x 43 kernel if $\sigma = 7$)
  - because the values of a Gaussian function at a distance larger than 3σ from the mean are small enough that they can be ignored.
- The **product** and **convolution** of two Gaussians are Gaussian functions also.

TABLE 3.6 Mean and standard deviation of the product (×) and convolution (★) of two 1-D Gaussian functions, f and g. These results generalize directly to the product and convolution of more than two 1-D Gaussian functions (see Problem 3.25).

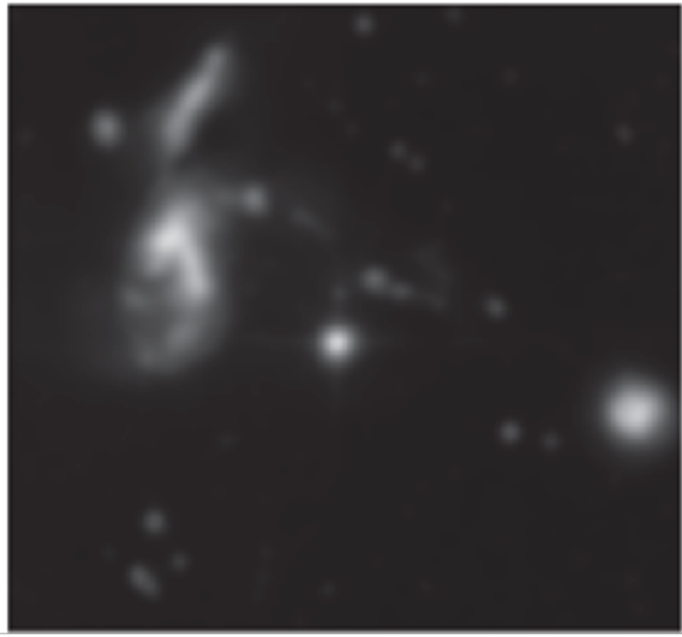
	f	g	$f \times g$	$f \star g$
Mean	m_f	m_g	$m_{f\times g} = \dfrac{m_f\sigma_g^2 + m_g\sigma_f^2}{\sigma_f^2 + \sigma_g^2}$	$m_{f\star g} = m_f + m_g$
Standard deviation	σ_f	σ_g	$\sigma_{f\times g} = \sqrt{\dfrac{\sigma_f^2\sigma_g^2}{\sigma_f^2 + \sigma_g^2}}$	$\sigma_{f\star g} = \sqrt{\sigma_f^2 + \sigma_g^2}$

3.5 Smoothing (Lowpass) Spatial Filters
- Lowpass Gaussian Filter Kernels

Example of applications for lowpass filtering: **region extraction**



Hubble Telescope image

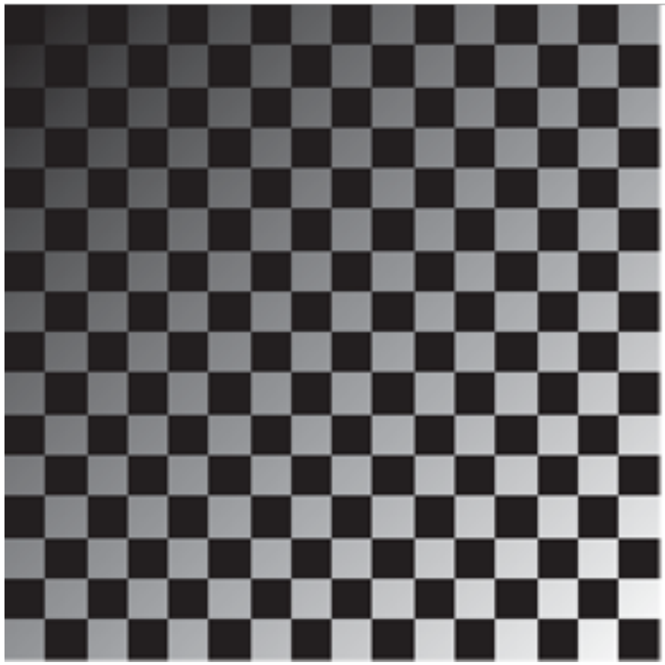


lowpass filtering with a Gaussian kernel

thresholding the filtered image

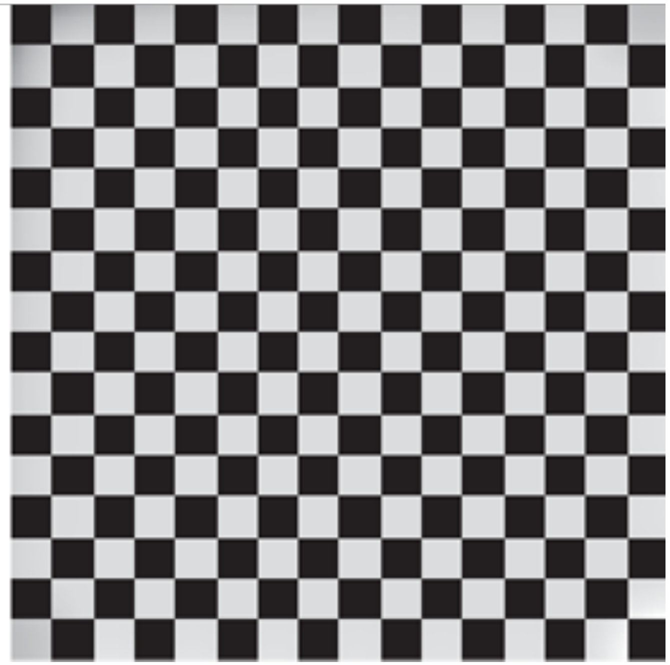
3.5 Smoothing (Lowpass) Spatial Filters - Lowpass Gaussian Filter Kernels

Example of applications for lowpass filtering: **shading correction**



(a) Image shaded

(b) Estimate of the shading patterns obtained using lowpass filtering



Result of dividing (a) by (b)

3.5 Smoothing (Lowpass) Spatial Filters - Order-Statistic (non-linear) Filters

Median Filter

-- the 50^{th} percentile of a ranked set of numbers

-- effective for reducing *impulse noise,*
or *salt-and-pepper noise*

Max Filter

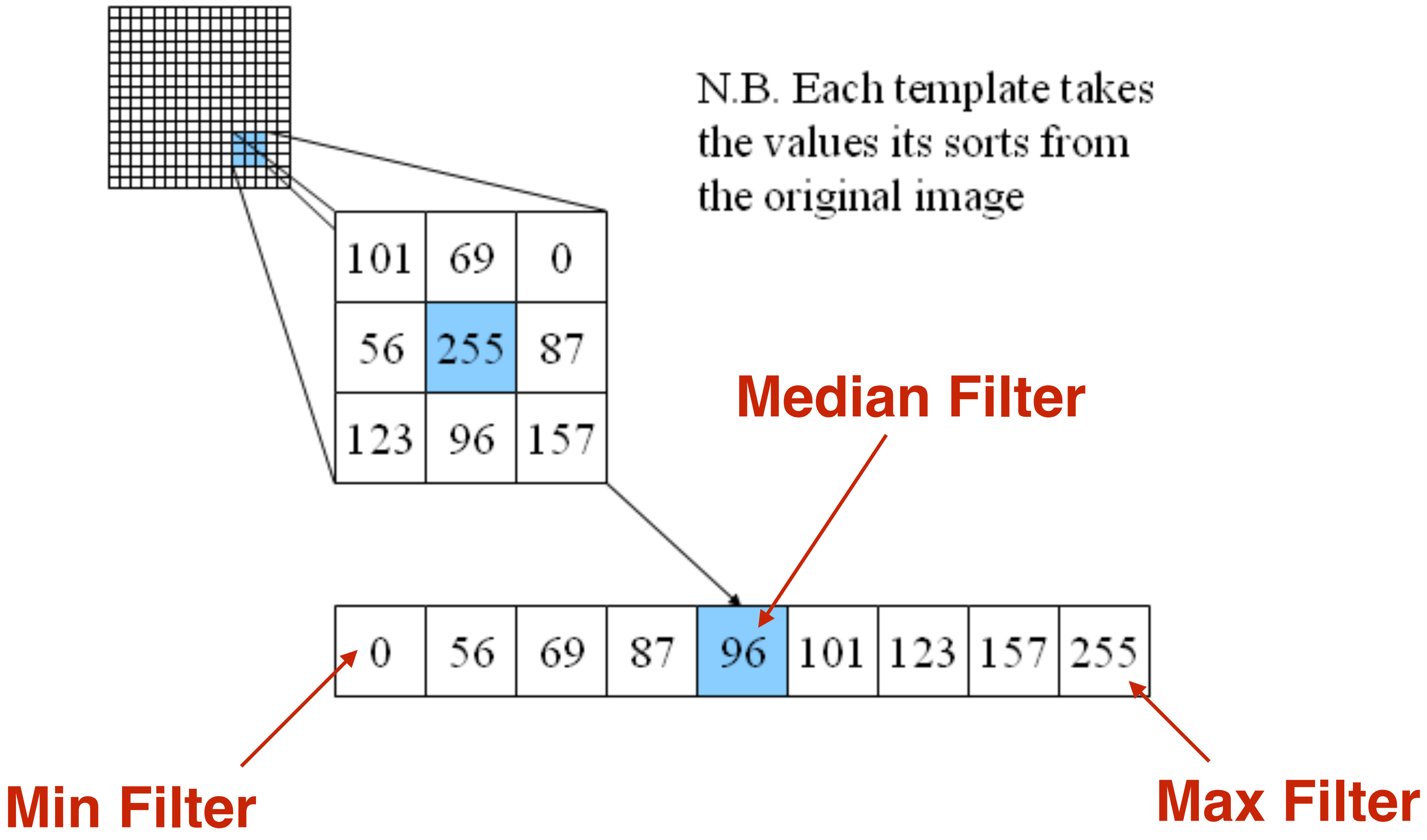
-- the 100^{th} percentile filter

Min Filter

-- the 0^{th} percentile filter

credit of this slide: Y. P. Hung

3.5 Smoothing (Lowpass) Spatial Filters - Order-Statistic (non-linear) Filters



source: http://www.southampton.ac.uk/~msn/book/new_demo/median/

3.5 Smoothing (Lowpass) Spatial Filters - Order-Statistic (non-linear) Filters

Median Filter

-- the 50^{th} percentile of a ranked set of numbers

-- effective for reducing *impulse noise,*
or *salt-and-pepper noise*

Max Filter

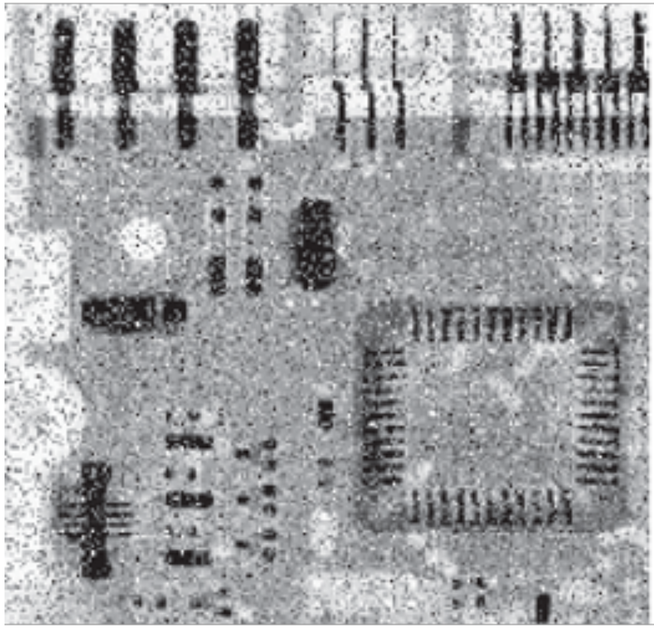
-- the 100^{th} percentile filter

Min Filter

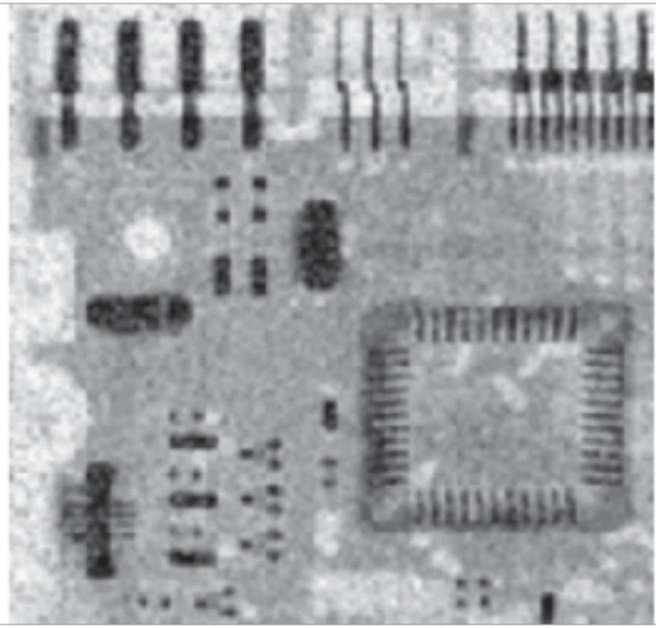
-- the 0^{th} percentile filter

credit of this slide: Y. P. Hung

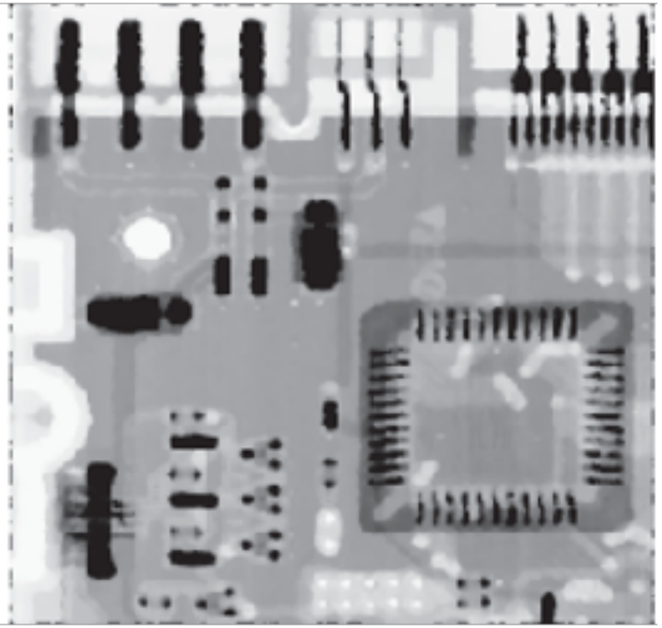
3.5 Smoothing (Lowpass) Spatial Filters
- Order-Statistic (non-linear) Filters



Salt-and-pepper noise image



19 x 19 Gaussian lowpass filter, $\sigma = 3$



7x7 median filter

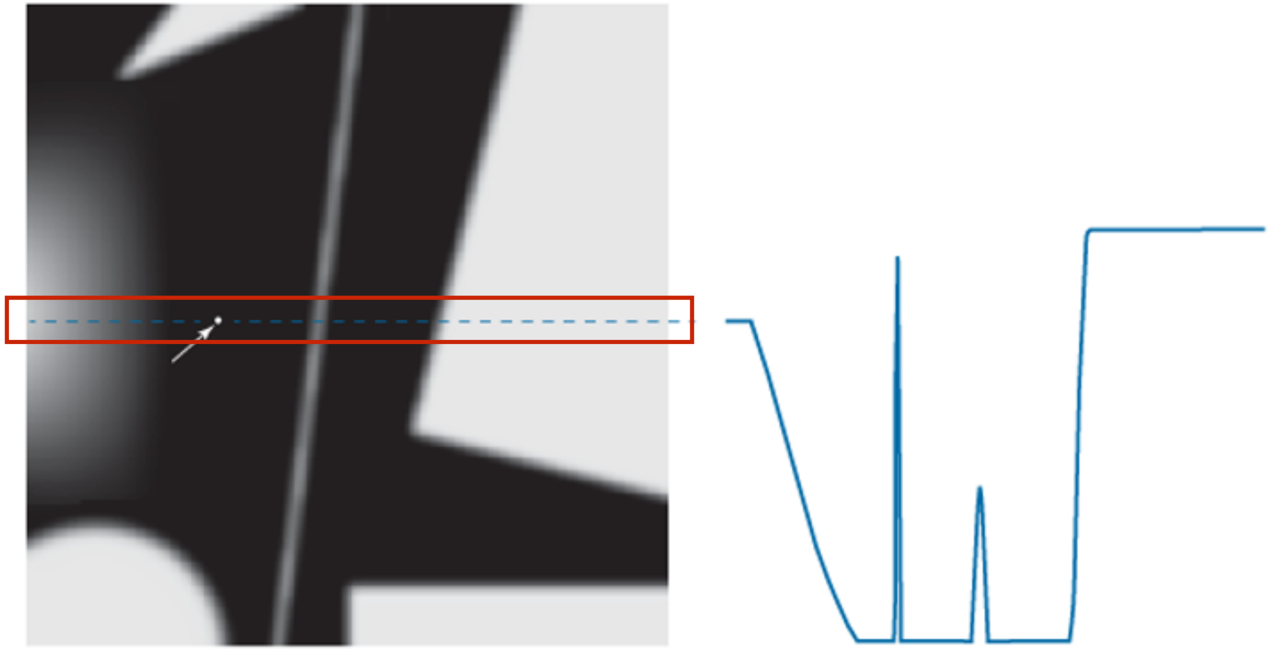
3.6 Sharpening (Highpass) Spatial Filters - Foundation

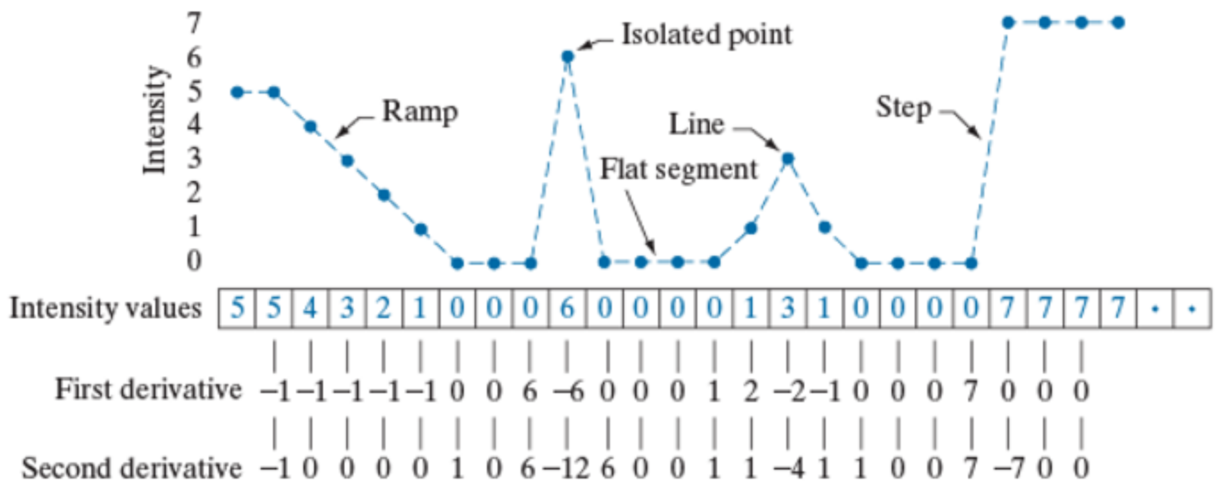
First Derivative:

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

Second Derivative:

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

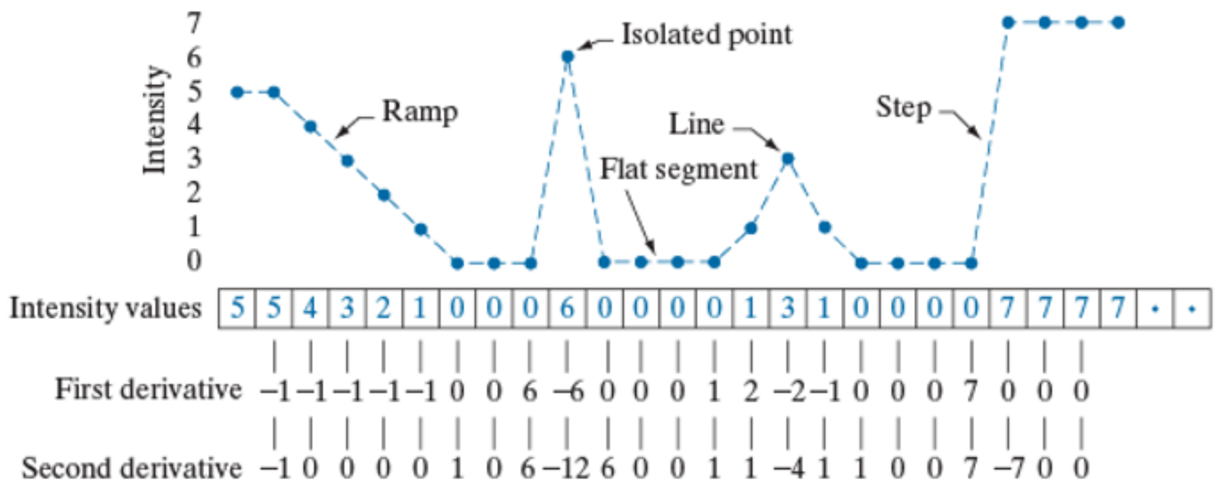




3.6 Sharpening (Highpass) Spatial Filters - Foundation

- First derivative:
  - Zero in areas of constant intensity
  - Nonzero at the onset of an intensity step or ramp
  - Nonzero along ramps

- Second derivative:
  - Zero in areas of constant intensity
  - Nonzero at the onset and end of an intensity step or ramp
  - Zero along ramps of constant slope



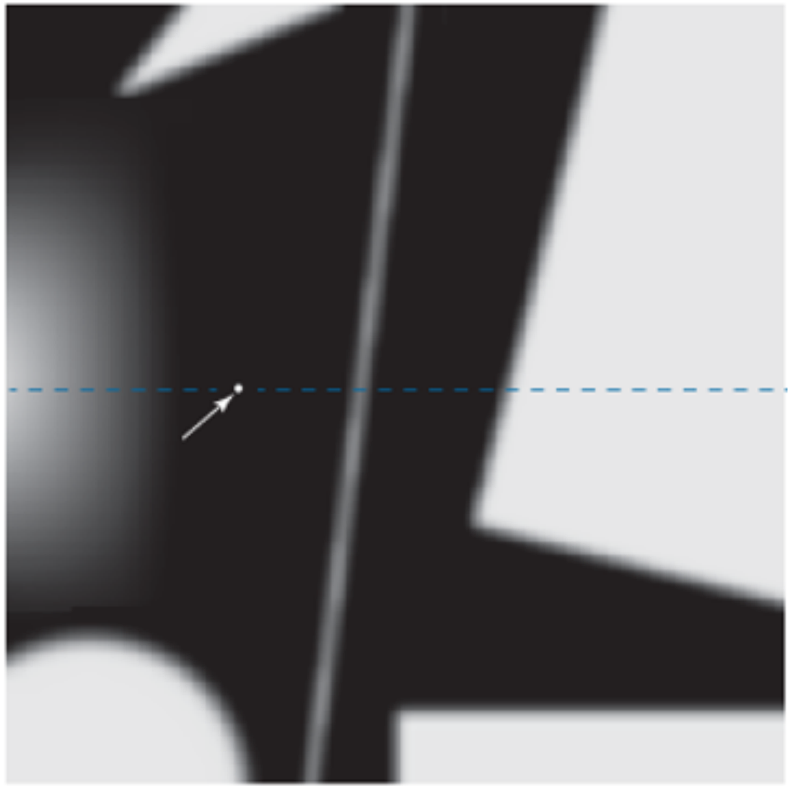
for edge detection

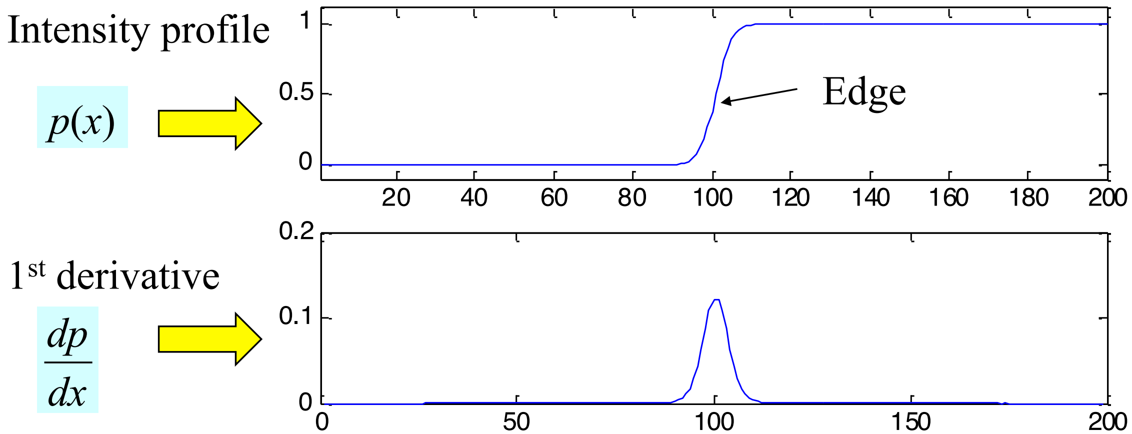
for sharpening

3.6 Sharpening (Highpass) Spatial Filters - Foundation

First derivative for edge detection

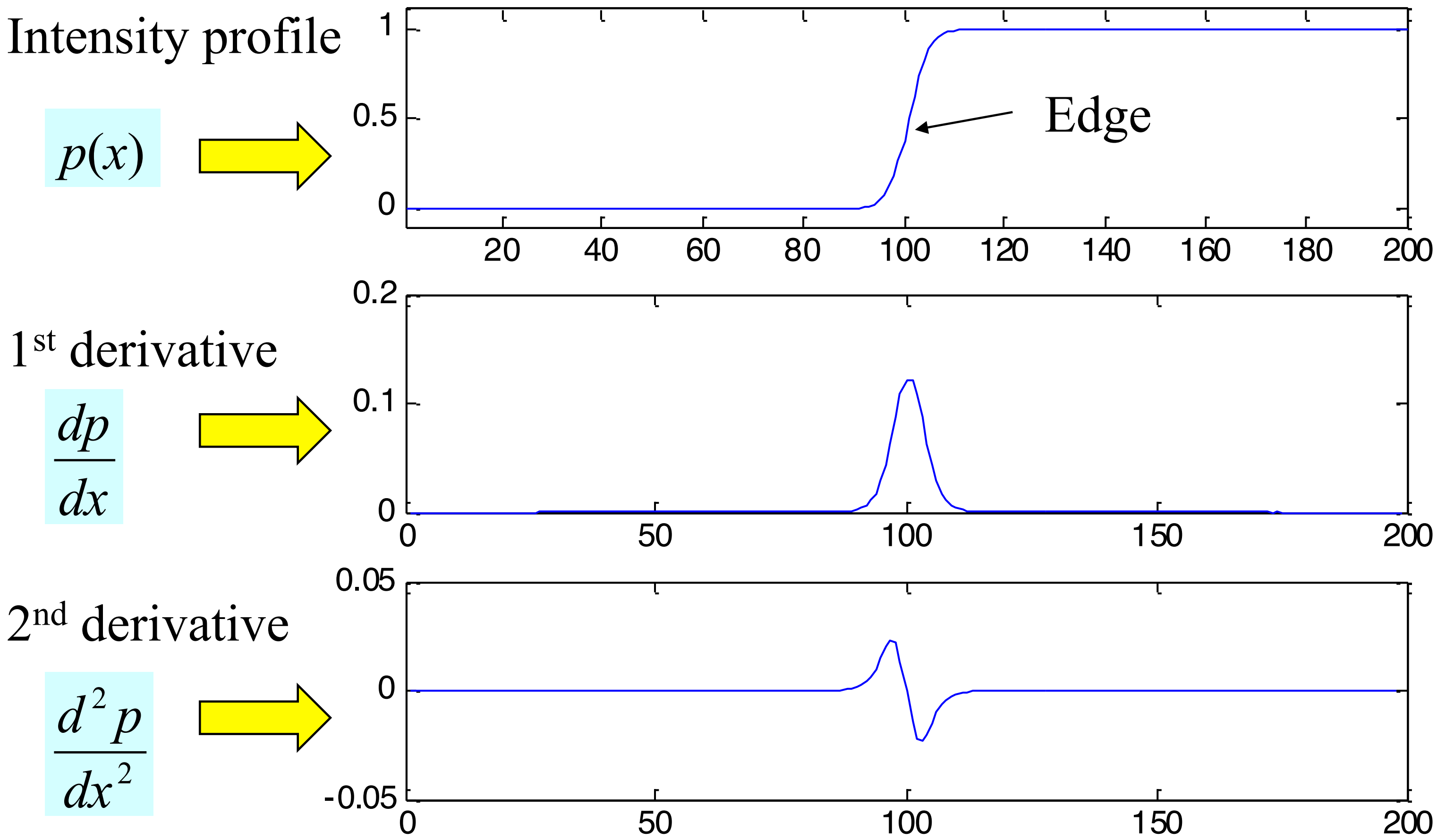
What is edge?





3.6 Sharpening (Highpass) Spatial Filters - Foundation

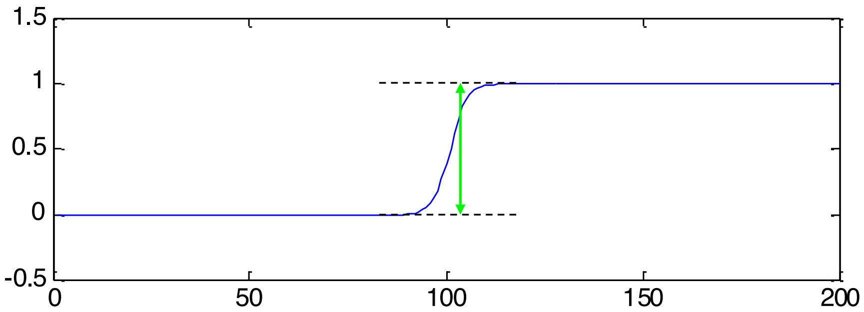
Second derivative for sharpening

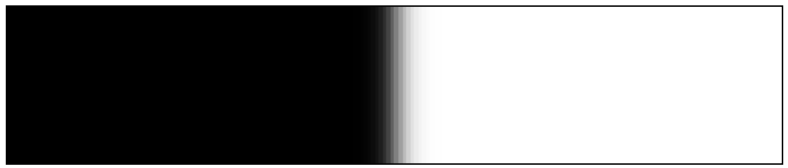


3.6 Sharpening (Highpass) Spatial Filters - Foundation

Second derivative for sharpening ➜ **Laplacian**

$p(x)$

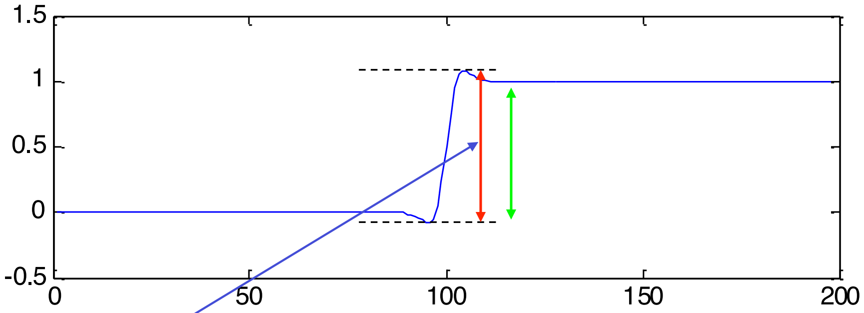


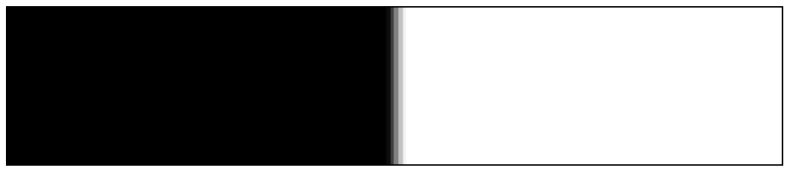


Before sharpening

$p(x)$

$p(x) - 10\dfrac{d^2 p}{dx^2}$





After sharpening

$p(x) - 10\dfrac{d^2 p}{dx^2}$

Laplacian sharpening results in larger intensity discontinuity near the edge.

3.6 Sharpening (Highpass) Spatial Filters - the Laplacian

The simplest isotropic derivative kernel is the Laplacian

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

Discrete form:

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

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

We then have:

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

3.6 Sharpening (Highpass) Spatial Filters - the Laplacian

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

The corresponding spatial filter:

0	1	0
1	−4	1
0	1	0

Other Laplacian kernels:

0	1	0
1	−4	1
0	1	0

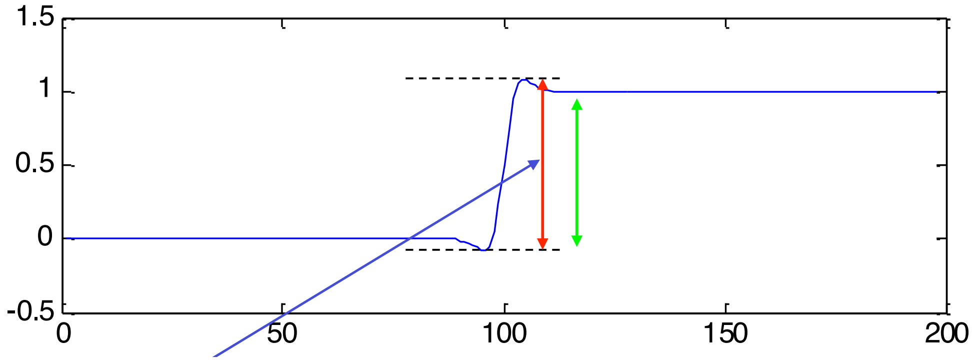
1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

3.6 Sharpening (Highpass) Spatial Filters - the Laplacian

Recall that $p(x) - 10\dfrac{d^2 p}{dx^2}$



the Laplacian for image sharpening:

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

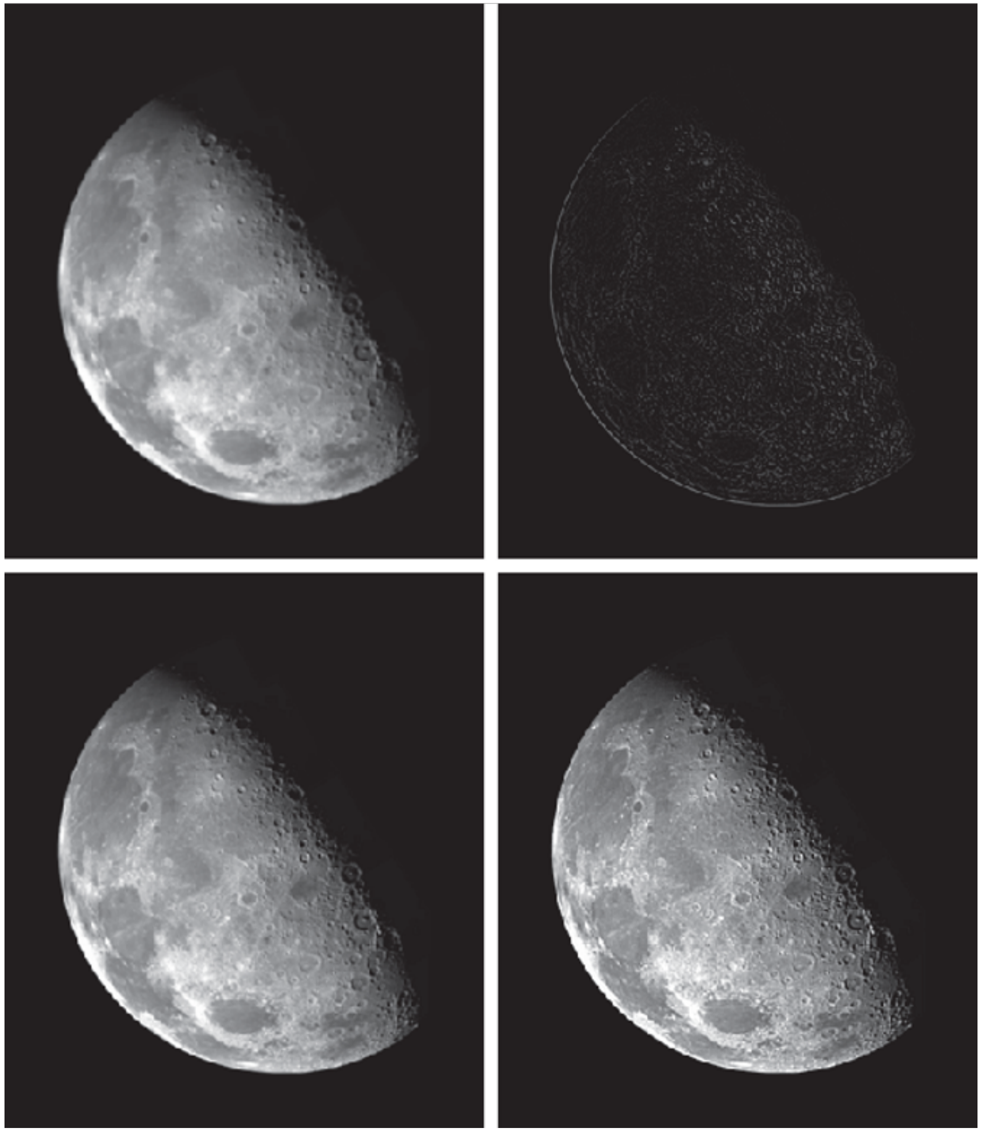
A general form in the book: $g(x,y) = f(x,y) + c\left[\nabla^2 f(x,y)\right]$

3.6 Sharpening (Highpass) Spatial Filters - the Laplacian

Example of Laplacian image sharpening

Original image

$$f(x,y) - \nabla^2 f$$



Laplacian image

0	1	0
1	−4	1
0	1	0

$$f(x,y) - \nabla^2 f$$

Laplacian kernel

1	1	1
1	−8	1
1	1	1

3.6 Sharpening (Highpass) Spatial Filters - the Laplacian

The entire enhancement can be combined into a single filtering operation

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

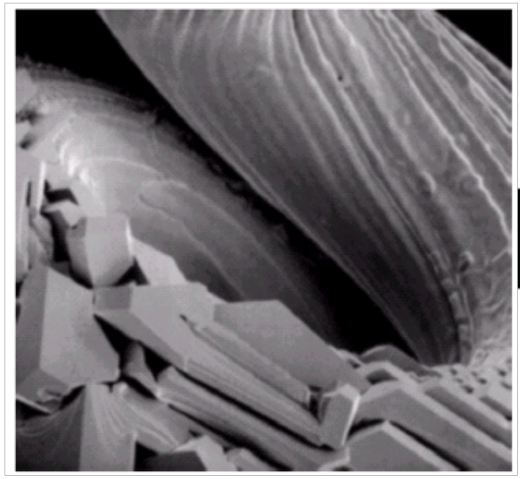
0	-1	0
-1	5	-1
0	-1	0

other variant:

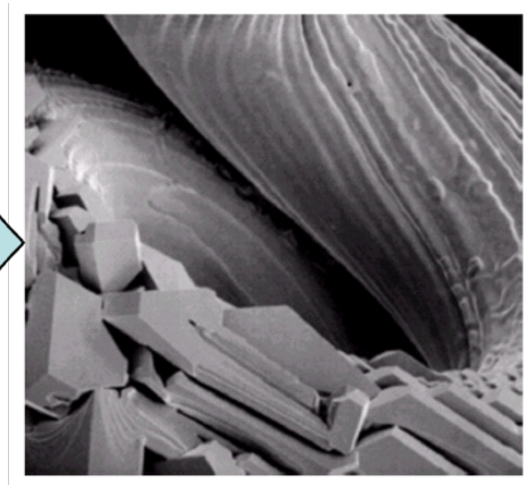
-1	-1	-1
-1	9	-1
-1	-1	-1

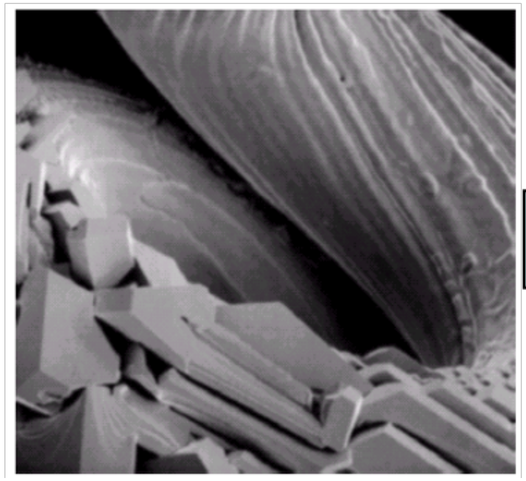
3.6 Sharpening (Highpass) Spatial Filters - the Laplacian

Example of Laplacian image sharpening

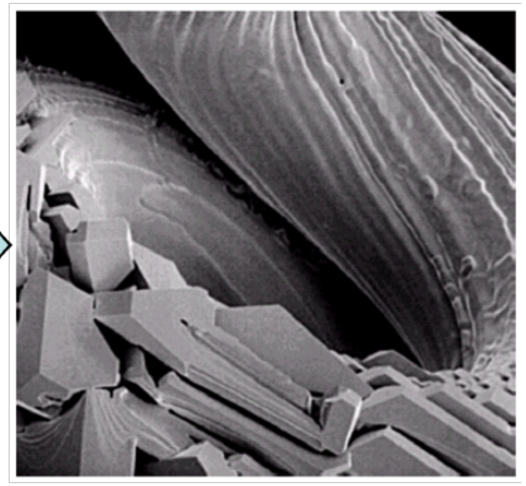


0	-1	0
-1	5	-1
0	-1	0





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



credit of this slide: C. Nikou

3.6 Sharpening (Highpass) Spatial Filters
- Unsharp Masking and Highboost Filtering

Unsharp Masking

Steps:

(1) Blur the image

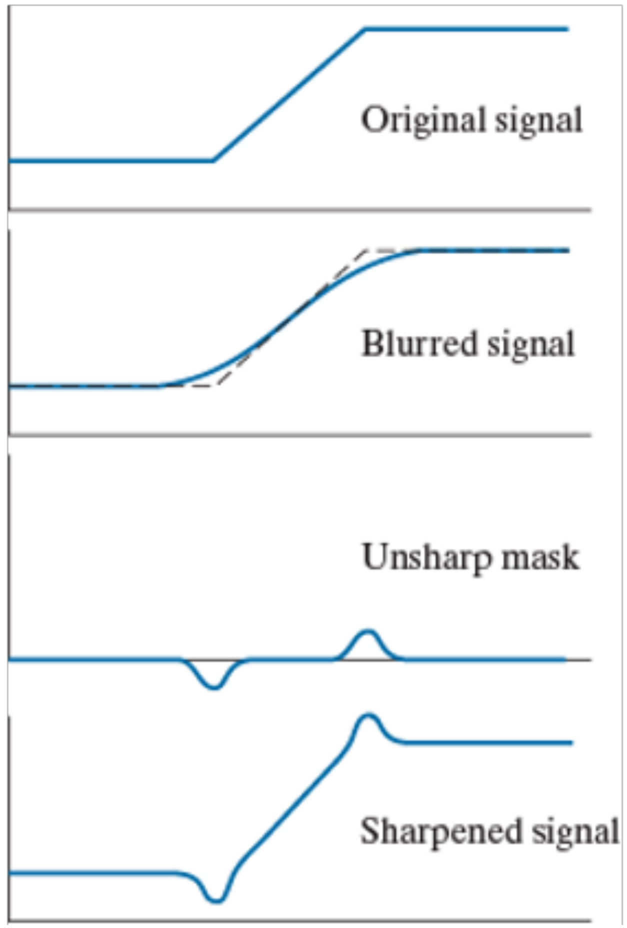
$$b(x,y)=Blur\{f(x,y)\}$$

(2) Subtract the blurred image from the original (the result is called the *mask*)

$$g_{mask}(x,y)=f(x,y)-b(x,y)$$

(3) Add the mask to the original

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



(1)
(2)
(3)

k = 1: **Unsharp Masking**

k > 1: **Highboost Filtering**

3.6 Sharpening (Highpass) Spatial Filters - Unsharp Masking and Highboost Filtering

Original Image

Blurred Image

Mask

Unsharp Masking Result

Highboost Filtering Result
(k = 4.5)

3.6 Sharpening (Highpass) Spatial Filters - the Gradient

First derivatives in image processing are implemented using the magnitude of the gradient.

$$\nabla f \equiv \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

The magnitude (length) of this vector:

$$M(x, y) = \|\nabla f\| = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

3.6 Sharpening (Highpass) Spatial Filters - the Gradient

The magnitude (length) of this vector:

$$M(x,y) = \|\nabla f\| = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

For practical reasons this can be simplified as:

$$M(x,y) \approx |g_x| + |g_y|$$

The simplest approximation:

$$g_x = (z_8 - z_5) \text{ and } g_y = (z_6 - z_5)$$

$$\boxed{\begin{matrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \end{matrix}} \qquad \boxed{\begin{matrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{matrix}}$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

3.6 Sharpening (Highpass) Spatial Filters - the Gradient

Two other definition:

Roberts Cross-Gradient Operators

−1	0
0	1

0	−1
1	0

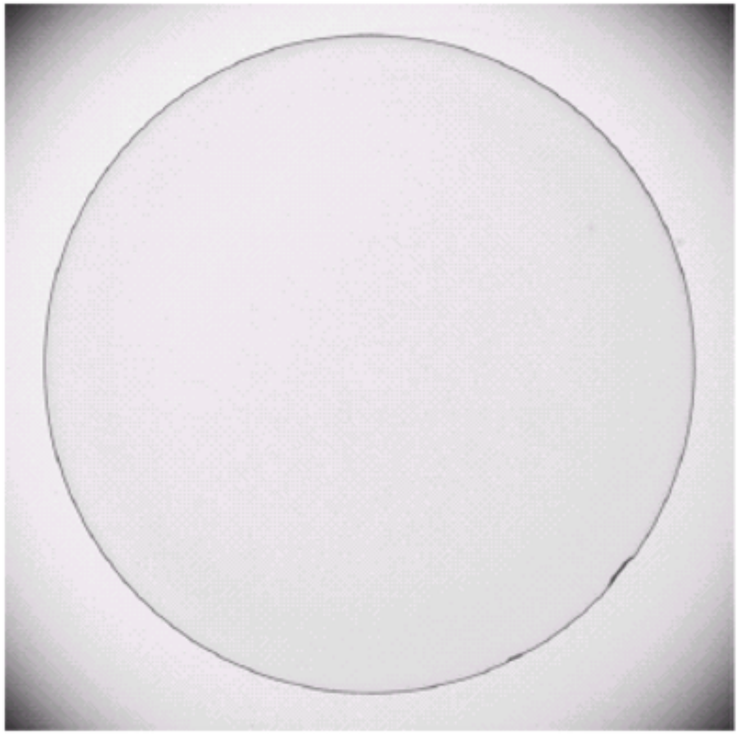
Sobel Operators

−1	−2	−1
0	0	0
1	2	1

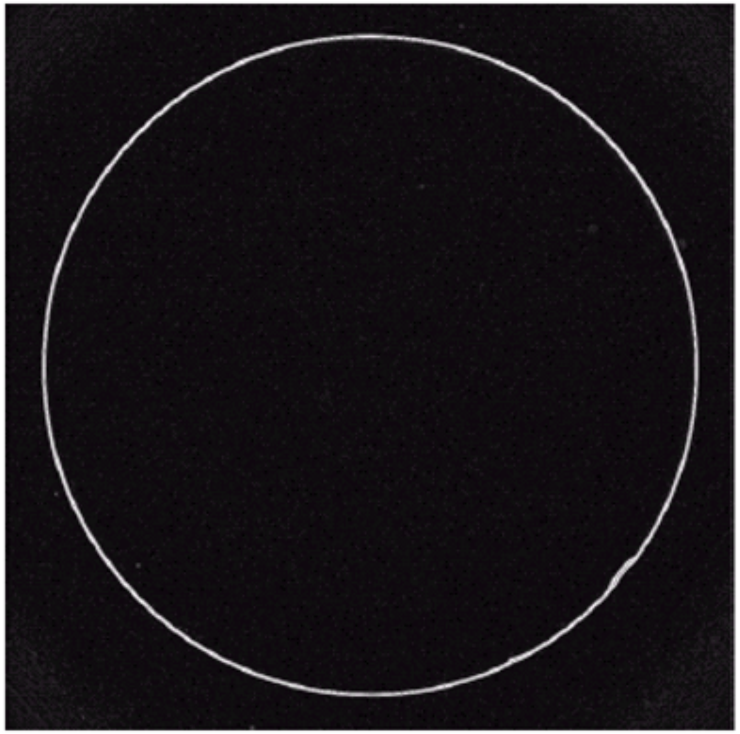
−1	0	1
−2	0	2
−1	0	1

3.6 Sharpening (Highpass) Spatial Filters - the Gradient

Example of using the gradient for edge enhancement:



Original Image

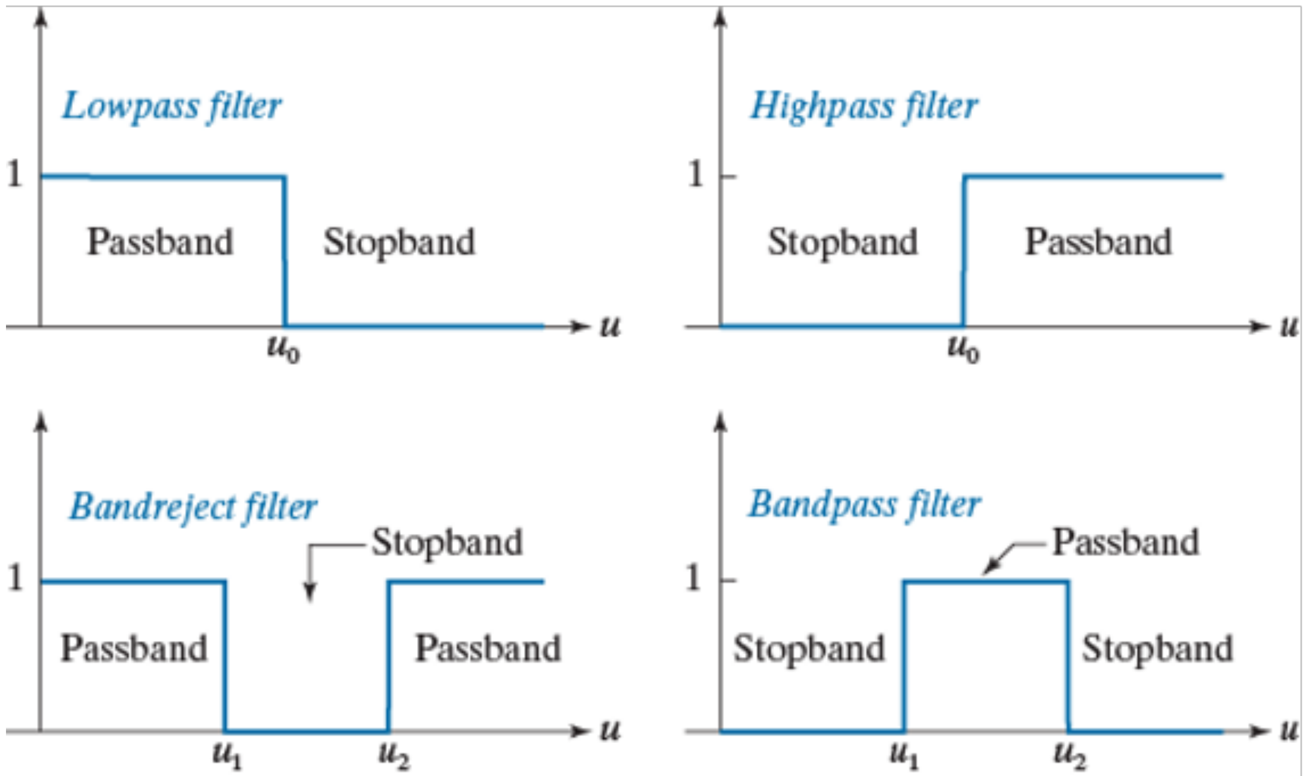


Sobel Operators

Sobel operators are typically used for edge detection.

3.7 Highpass, Bandreject, and Bandpass Filters from Low Pass Filters

Skip this subchapter, because most of the contents belongs to Chapter 4.



3.8 Combing Spatial Enhancement Methods

- Successful image enhancement is typically not achieved using a single operation. Rather we combine a range of techniques in order to achieve a final result.

credit of this slide: C. Nikou

3.8 Combing Spatial Enhancement Methods

(a)

Laplacian filter of bone scan (a)

(b)

Sharpened version of bone scan achieved by adding (a) and (b)

(c)

credit of this slide: C. Nikou

3.8 Combing Spatial Enhancement Methods

(a)

Laplacian filter of
bone scan (a)

Sh
bor
by

(c)

3.8 Combing Spatial Enhancement Methods

(a)

Laplacian filter of bone scan (a)

(b)

Sharpened version of bone scan achieved by adding (a) and (b)

(c)

Sobel operator of bone scan (a)

(d)

credit of this slide: C. Nikou

3.8 Combing Spatial Enhancement Methods

The product of (b) and (e) which will be used as a mask

Sharpened image which is sum of (a) and (f)

Result of applying a power-law trans. to (g)

(e)

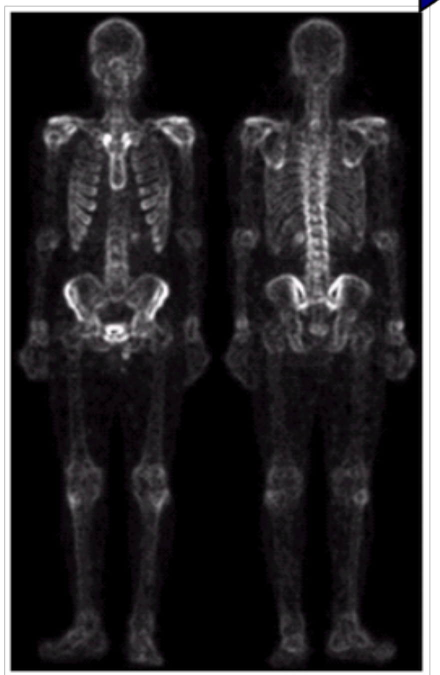
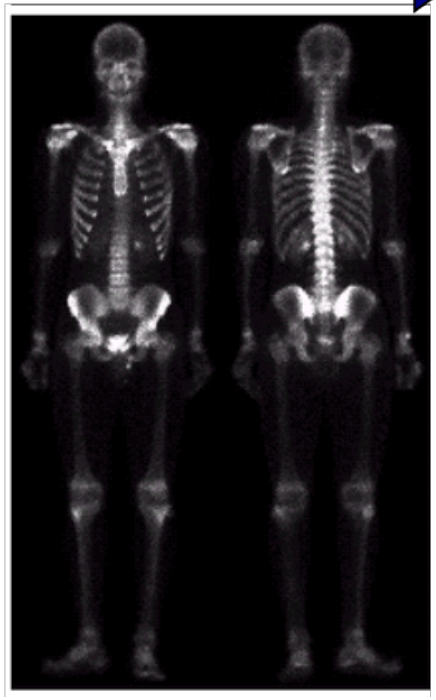


Image (d) smoothed with a 5*5 box filter

(f)

(g)



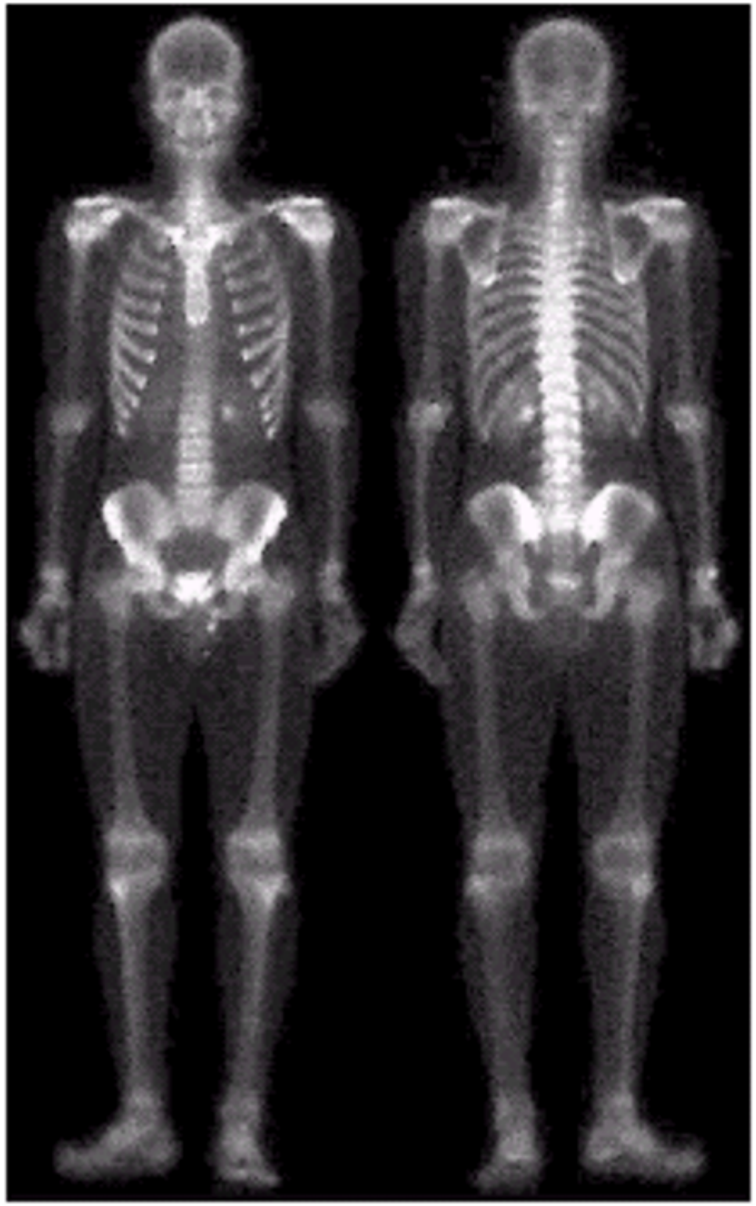
(h)

credit of this slide: C. Nikou

3.8 Combing Spatial Enhancement Methods

Original Image

Final Image



Introduction to Image Processing

Ch 3. Intensity Transformations and Spatial Filtering

Kuan-Wen Chen