DATA ANALYSIS AND VISUALIZATION

INSTRUCTOR: UMME AMMARAH

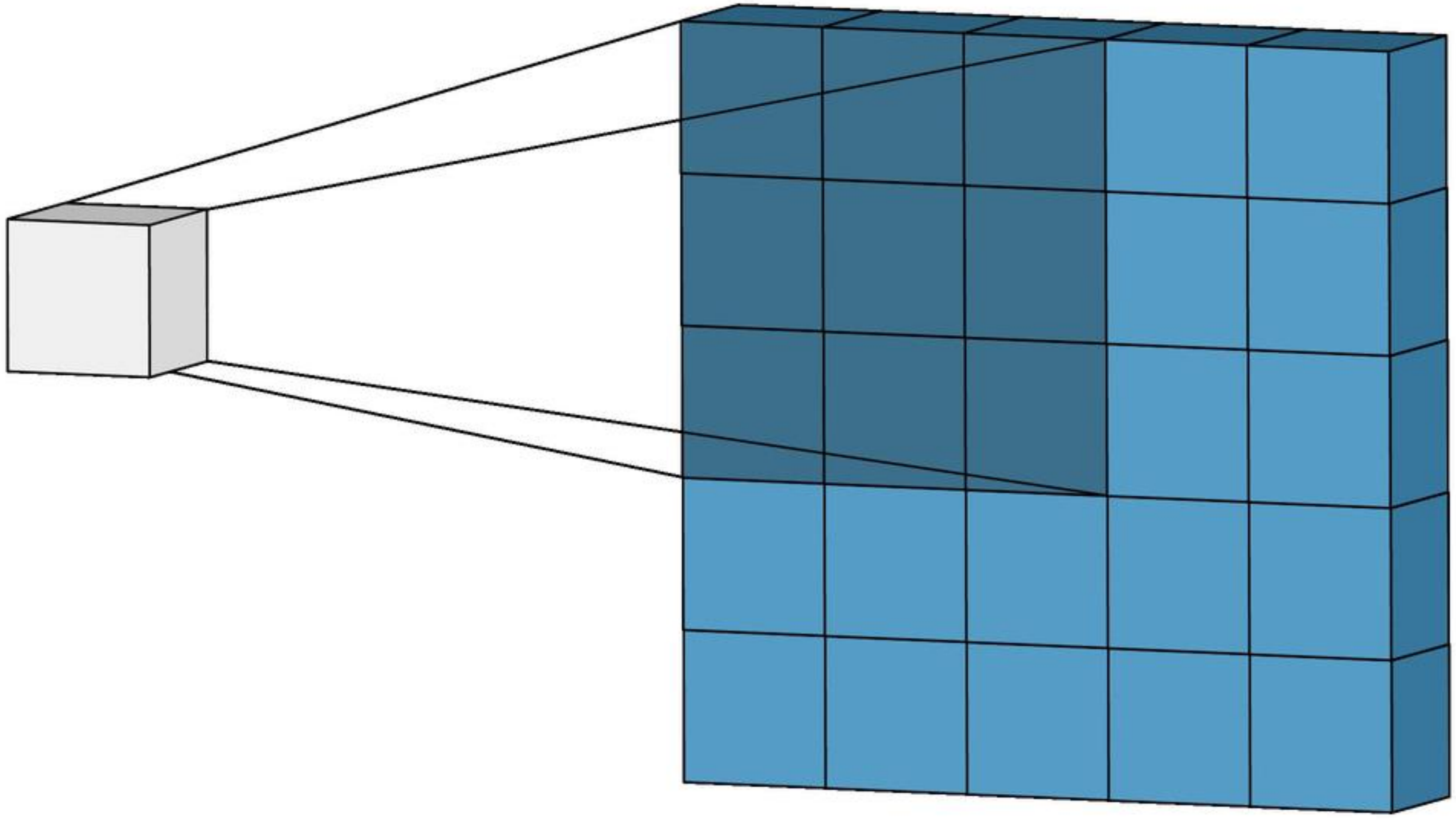
SPATIAL FILTERING

FILTER KERNEL

- The kernel is an array whose size defines the neighborhood of operation
- Whose coefficients determine the nature of the filter.
- Other terms used to refer to a spatial filter kernel are mask, template, and window.

LINEAR SPATIAL FILTERING

- The process consists simply of moving the filter mask from point to point in an image.
- At each point (x,y) the response of the filter at that point is calculated using a predefined relationship.
- A linear spatial filter performs a sum-of-products operation between an image f and a filter kernel, w.

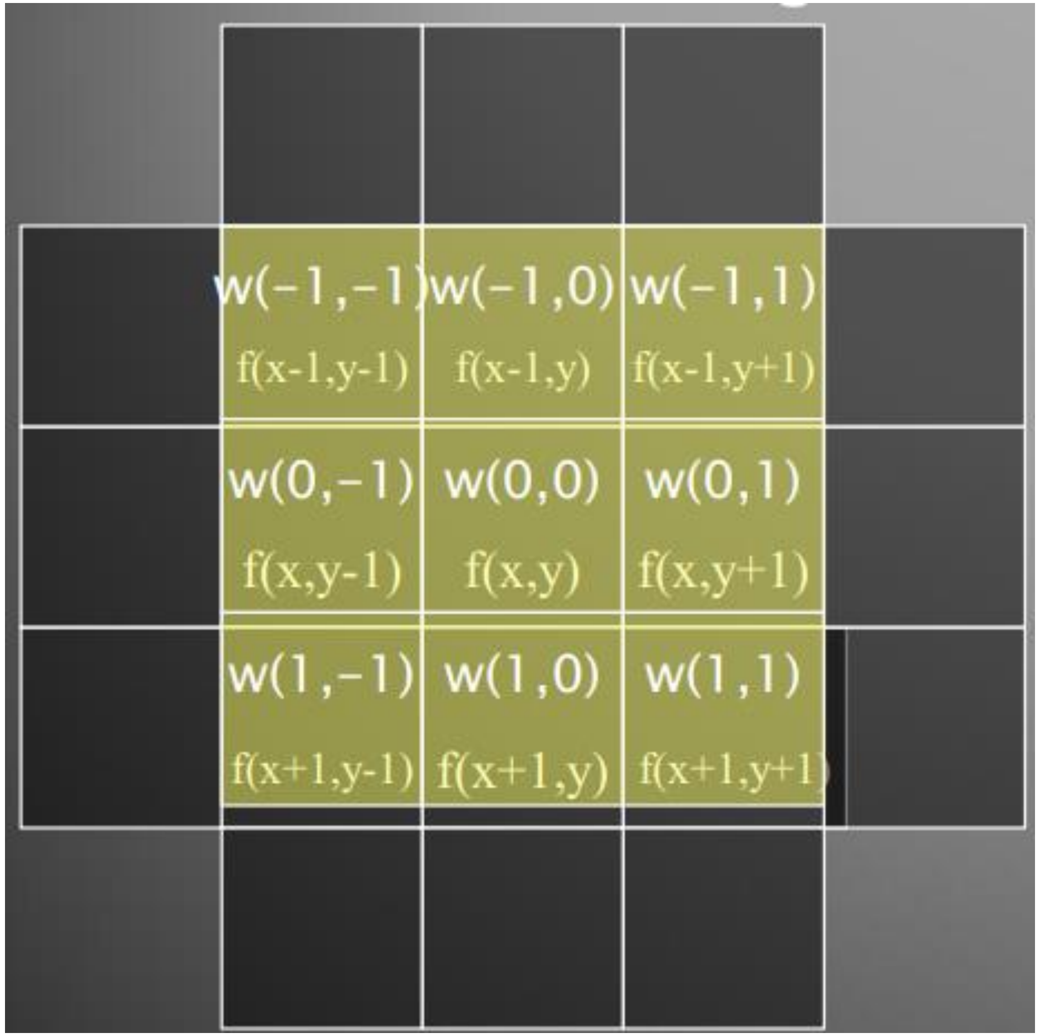


TRANSFORMED VALUE OF A PIXEL

- The result is the sum of products of the mask coefficients with the corresponding pixels directly under the mask.

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

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



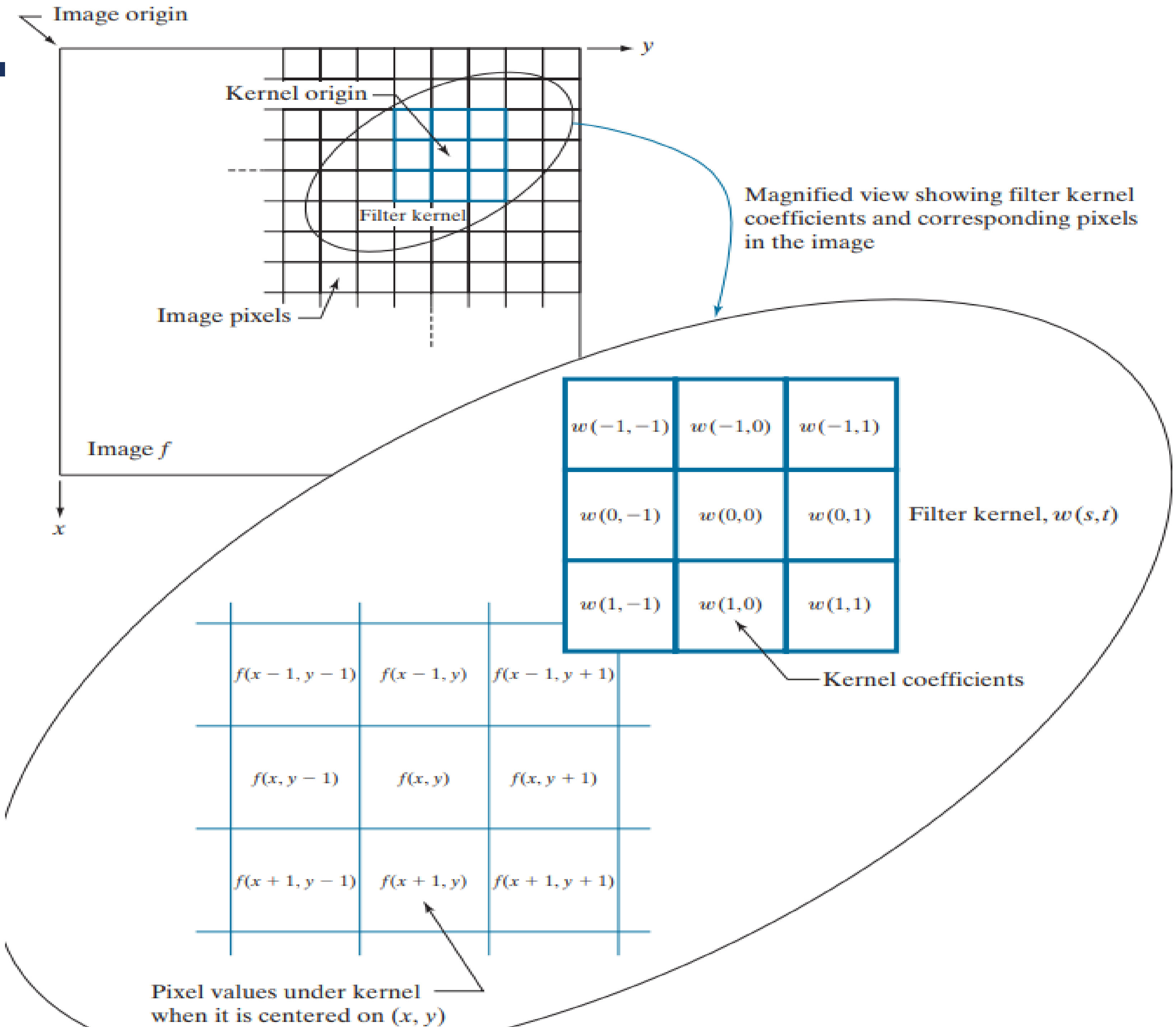


Image origin
y
Kernel origin
Filter kernel
Magnified view showing filter kernel coefficients and corresponding pixels in the image
Image pixels
Image f
x
w(−1,−1)
w(−1,0)
w(−1,1)
w(0,−1)
w(0,0)
w(0,1)
w(1,−1)
w(1,0)
w(1,1)
Filter kernel, w(s,t)
Kernel coefficients
f(x − 1, y − 1)
f(x − 1, y)
f(x − 1, y + 1)
f(x, y − 1)
f(x, y)
f(x, y + 1)
f(x + 1, y − 1)
f(x + 1, y)
f(x + 1, y + 1)
Pixel values under kernel when it is centered on (x, y)

Spatial Filtering (Neighborhood Processing)

Linear Spatial Filtering

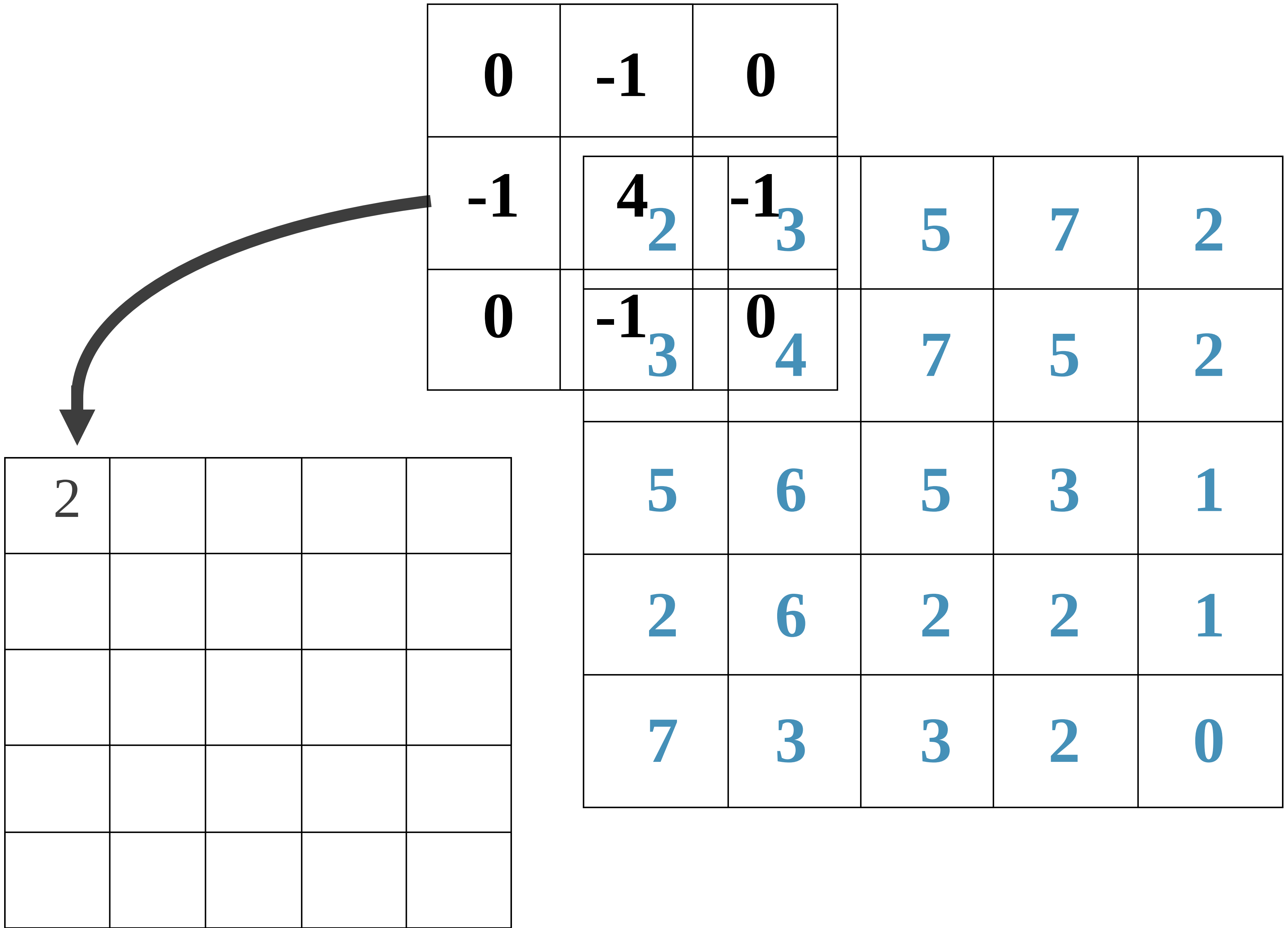
0	-1	0
-1	4	-1
0	-1	0

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

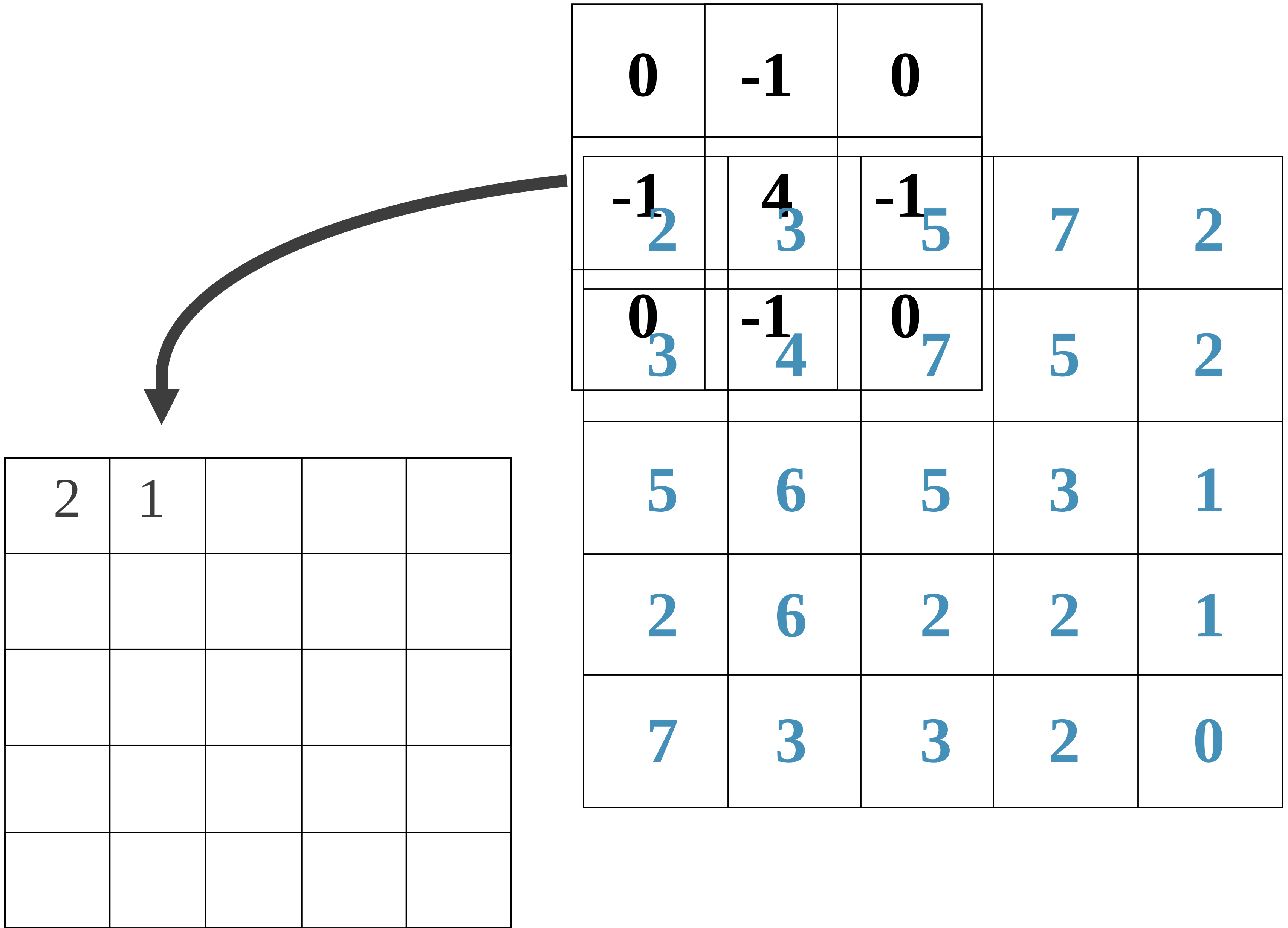
w: positioned so that its center coefficient is coincident with the origin of f

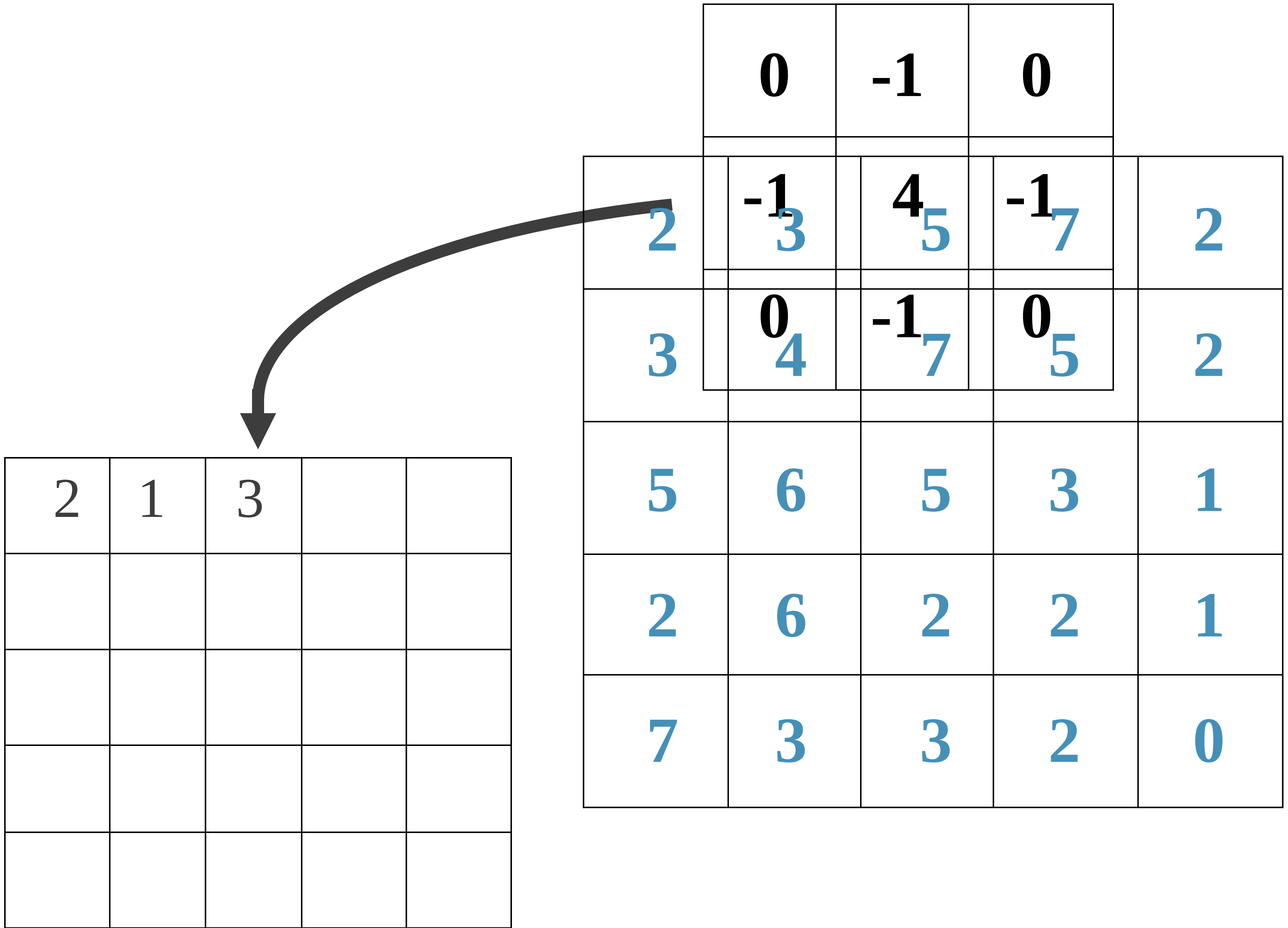
0	-1	0
-1	4	-1
0	-1	0

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

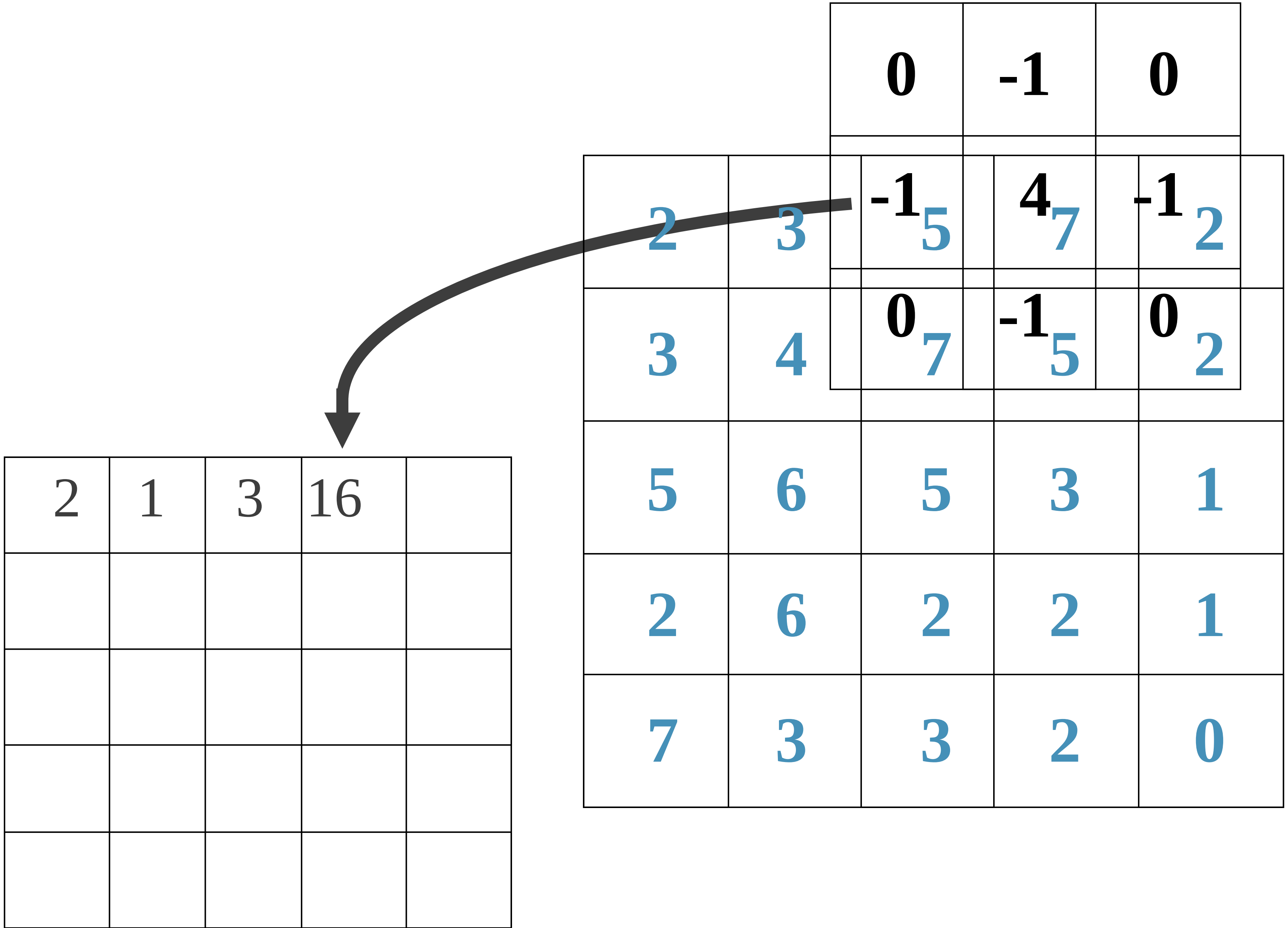


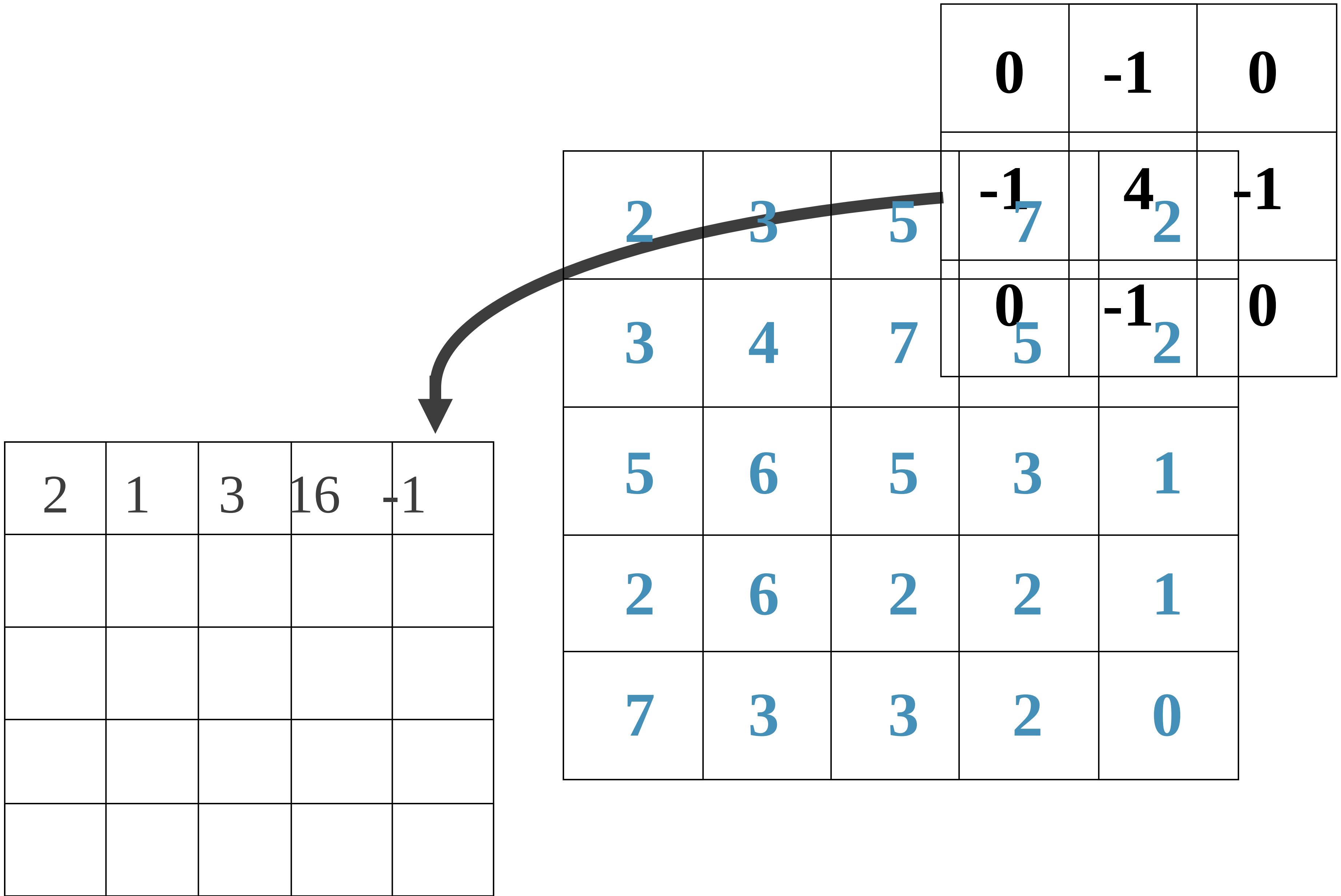
0 -1 0
-1 4 -1
0 -1 0
2 3 5 7 2
3 4 7 5 2
5 6 5 3 1
2 6 2 2 1
7 3 3 2 0
2

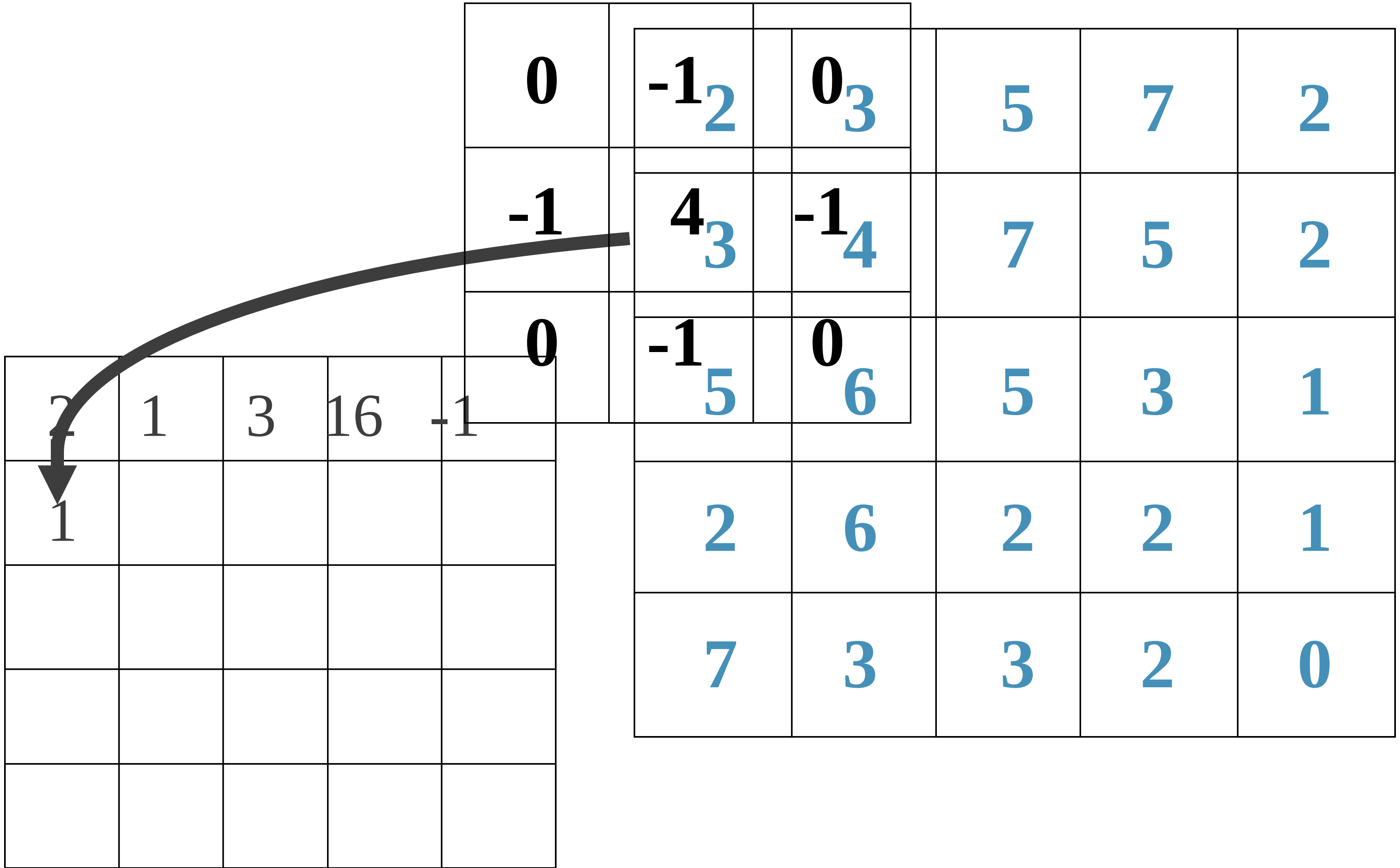

0 -1 0
-1 4 -1
0 -1 0
2 3 5 7 2
3 4 7 5 2
5 6 5 3 1
2 6 2 2 1
7 3 3 2 0
2 1

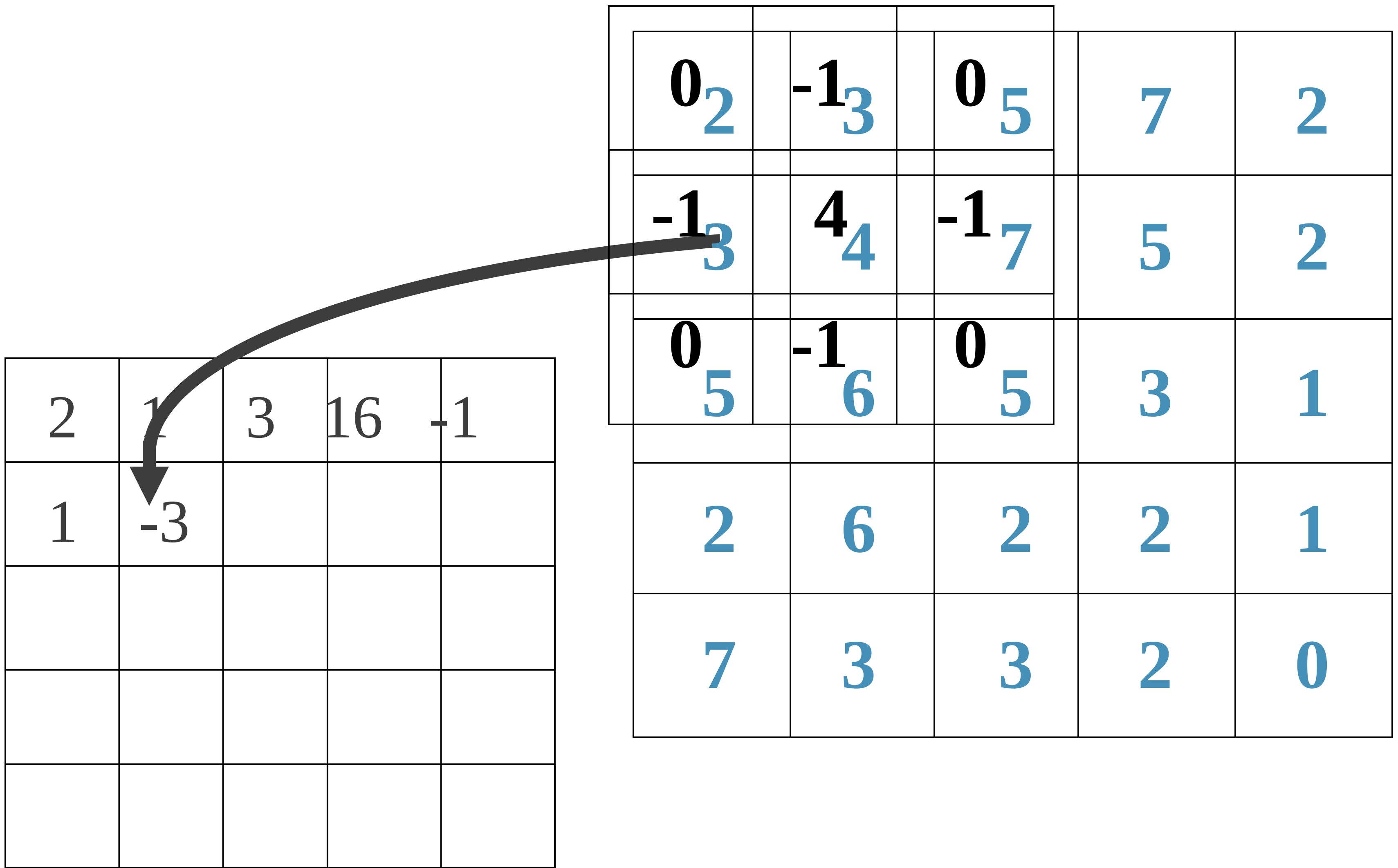


2 1 3
0 -1 0
-1 4 -1
0 -1 0
2 3 5 7 2
3 4 7 5 2
5 6 5 3 1
2 6 2 2 1
7 3 3 2 0


0 -1 0
-1 4 -1
0 -1 0
2 3 5 7 2
3 4 7 5 2
5 6 5 3 1
2 6 2 2 1
7 3 3 2 0
2 1 3 16


0 -1 0
-1 4 -1
0 -1 0
2 3 5 7 2
3 4 7 5 2
5 6 5 3 1
2 6 2 2 1
7 3 3 2 0
2 1 3 16 -1


0 -1 0
-1 4 -1
0 -1 0
2 3 5 7 2
3 4 7 5 2
5 6 5 3 1
2 6 2 2 1
7 3 3 2 0
2 1 3 16 -1
1



2 1 3 16 -1
1 -3
0 -1 0
-1 4 -1
0 -1 0
2 3 5 7 2
3 4 7 5 2
5 6 5 3 1
2 6 2 2 1
7 3 3 2 0

SPATIAL CORRELATION AND CONVOLUTION

- **Correlation** consists of moving the center of a kernel over an image, and computing the sum of products at each location.

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

- If the values of a kernel are symmetric about its center, correlation and convolution yield the same result

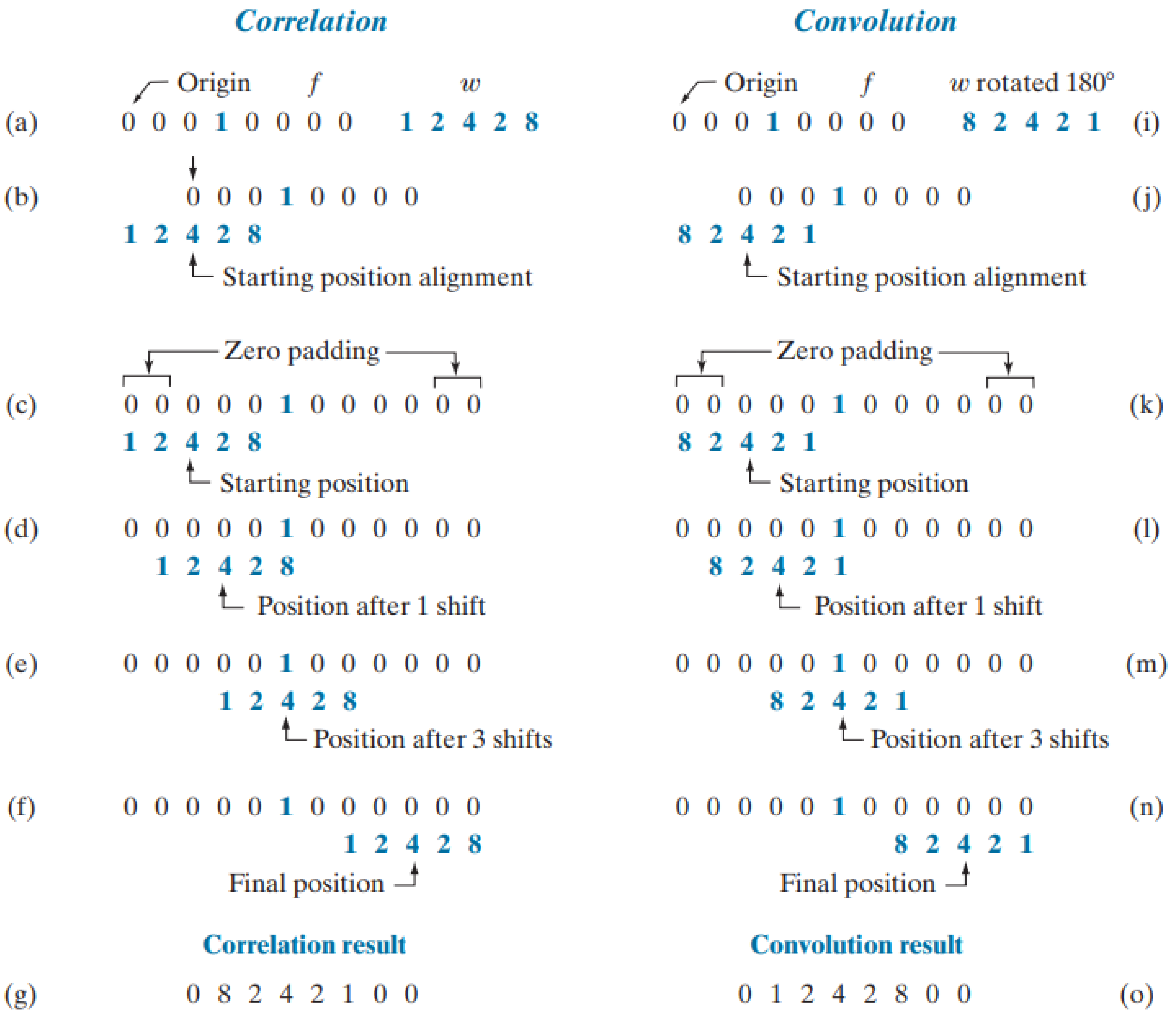
PADDING

- For a kernel of size m x n , we pad the image with a minimum of $(m-1)/2 = a$ rows of 0's at the top and bottom and $(n-1)/2 = b$ columns of 0's on the left and right. Then the size of the resulting correlation or convolution array will be MxN.

- For getting full convolution or correlation padding should be of size $(m-1)$ =2a and $(n-1)$=2b. Then the size of the resulting full correlation or convolution array will be of $S_v = m + (M-1)$ and $S_h = n + (N-1)$

- Where M x N is size of original image.

Correlation and Convolution in 1D



Correlation and Convolution in 2D

Origin f

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \qquad w \quad \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}$$

(a)

Padded f

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

(b)

Initial position for w

$$\begin{matrix} 1 & 2 & 3 & 0 & 0 & 0 & 0 \\ 4 & 5 & 6 & 0 & 0 & 0 & 0 \\ 7 & 8 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

(c)

Correlation result

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 9 & 8 & 7 & 0 \\ 0 & 6 & 5 & 4 & 0 \\ 0 & 3 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

(d)

Full correlation result

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 9 & 8 & 7 & 0 & 0 \\ 0 & 0 & 6 & 5 & 4 & 0 & 0 \\ 0 & 0 & 3 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

(e)

Rotated w

$$\begin{matrix} 9 & 8 & 7 & 0 & 0 & 0 & 0 \\ 6 & 5 & 4 & 0 & 0 & 0 & 0 \\ 3 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

(f)

Convolution result

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 \\ 0 & 4 & 5 & 6 & 0 \\ 0 & 7 & 8 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

(g)

Full convolution result

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 3 & 0 & 0 \\ 0 & 0 & 4 & 5 & 6 & 0 & 0 \\ 0 & 0 & 7 & 8 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

(h)

EQUATIONS

- Correlation:

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

- Convolution:

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

PROPERTIES

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

SMOOTHING (LOWPASS) FILTERS

SMOOTHING (LOWPASS) SPATIAL FILTERS

- Smoothing (also called averaging) spatial filters are used to reduce sharp transitions in intensity.
  - Blurring
  - Noise reduction
  - Removal of small details from an image
  - Bridging of small gaps in lines or curves

- There are 2 types of smoothing spatial filters
  - Smoothing Linear Filters
  - Order-Statistics Filters

SMOOTHING LINEAR FILTERS

- Convolving a smoothing kernel with an image blurs the image.
- The degree of blurriness is determined by the size of the kernel and the values of its coefficients.
- Sometimes called "averaging filters".

- Types:
  - Box filter kernel
  - Gaussian kernel

BOX FILTER KERNEL

- The simplest lowpass filter kernel, whose coefficients have the same value (typically 1).

TWO 3X3 SMOOTHING LINEAR FILTERS

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

Standard average

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

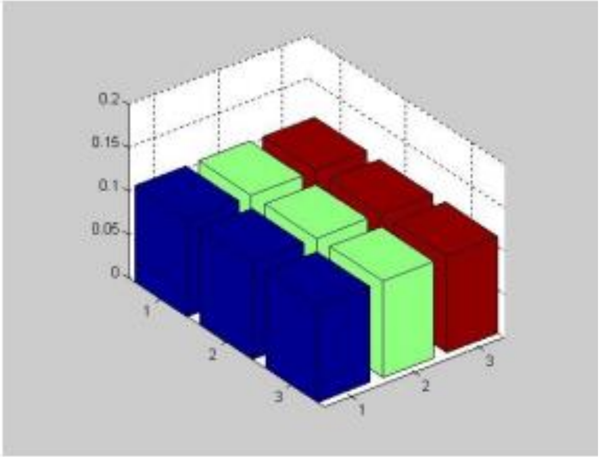
Weighted average

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

Normalization factor

Image averaging $R = \frac{1}{9}\sum_{i=1}^{9} z_i$

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9



$* \frac{1}{9}$

1	1	1
1	1	1
1	1	1

$=$

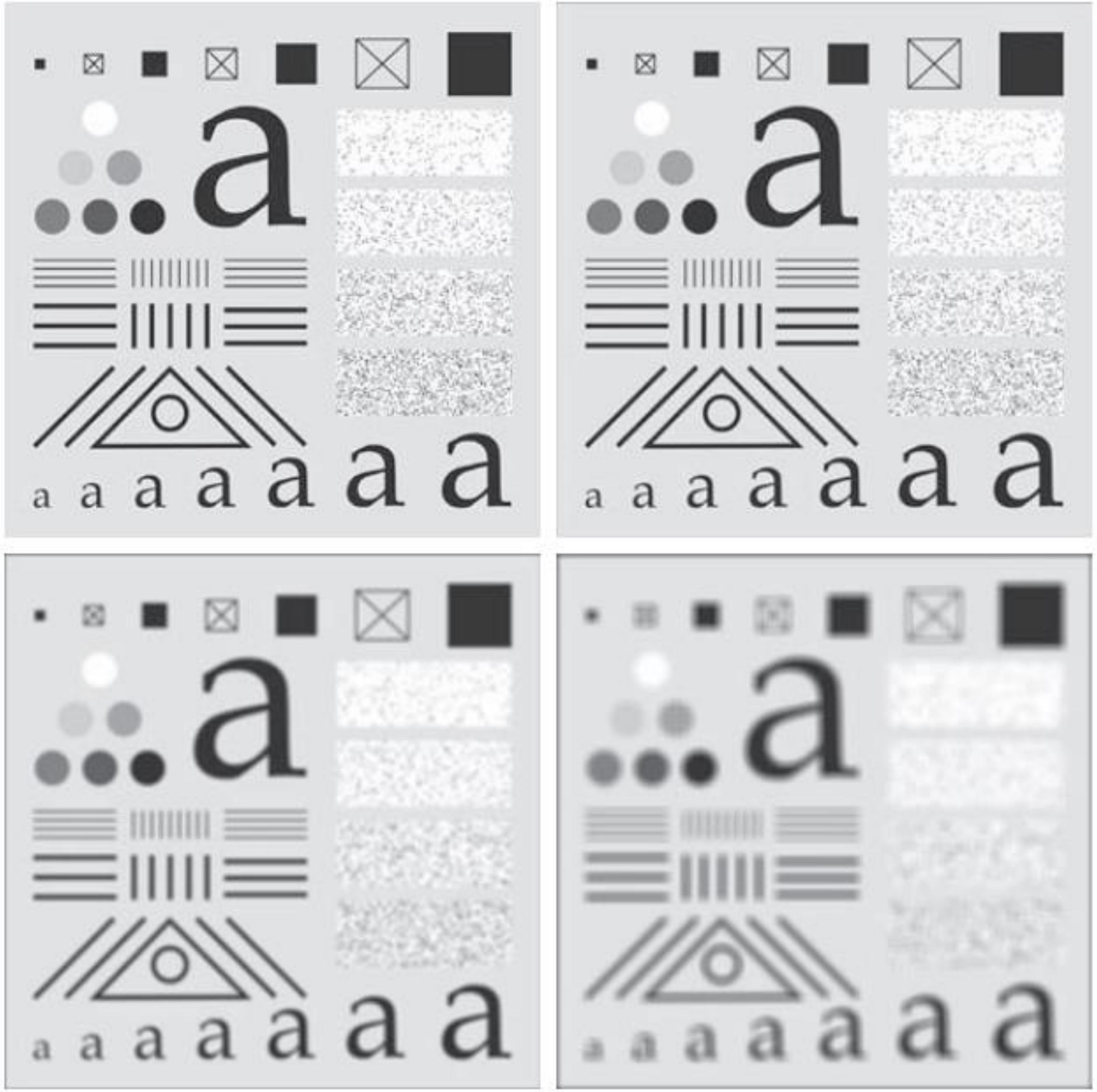
Result of smoothing filter of different sizes

Original Image

[3x3]

[5x5]

[7 x 7]



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.

LOWPASS GAUSSIAN FILTER KERNELS

- To reduce the loss of visual information, it is possible to assign different weights to mask coefficients so that more importance is attached to some pixels. This is to ensure that filter has one peak.

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

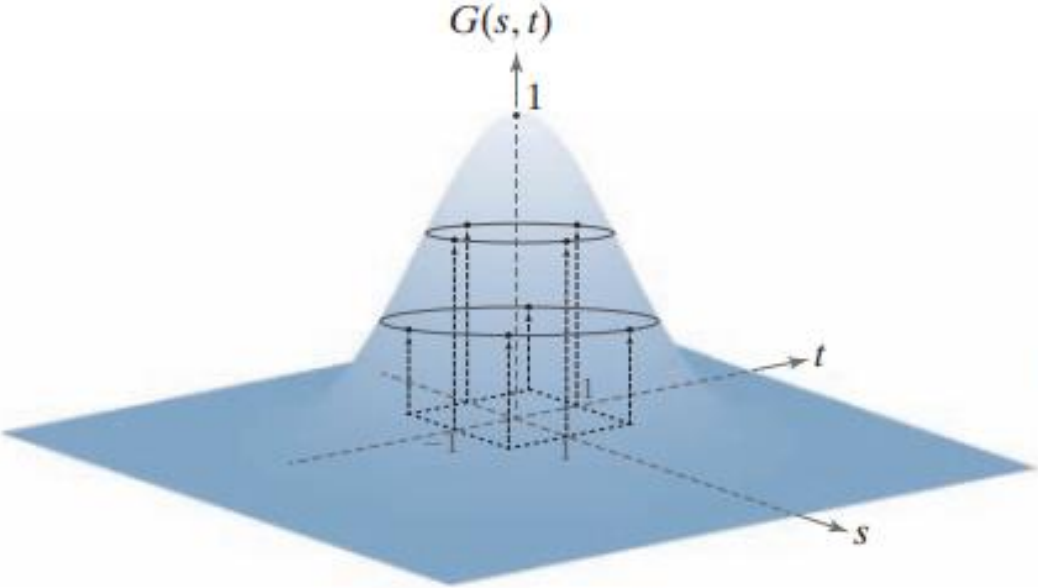
By letting $r = [s^2 + t^2]^{1/2}$

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

GAUSSIAN KERNEL

a b

FIGURE 3.35
(a) Sampling a Gaussian function to obtain a discrete Gaussian kernel. The values shown are for $K = 1$ and $\sigma = 1$. (b) Resulting 3×3 kernel [this is the same as Fig. 3.31(b)].



$$\frac{1}{4.8976} \times \begin{array}{|c|c|c|} \hline 0.3679 & 0.6065 & 0.3679 \\ \hline 0.6065 & 1.0000 & 0.6065 \\ \hline 0.3679 & 0.6065 & 0.3679 \\ \hline \end{array}$$

EXAMPLES

- Gaussian kernels have to be larger than box filters to achieve the same degree of blurring.

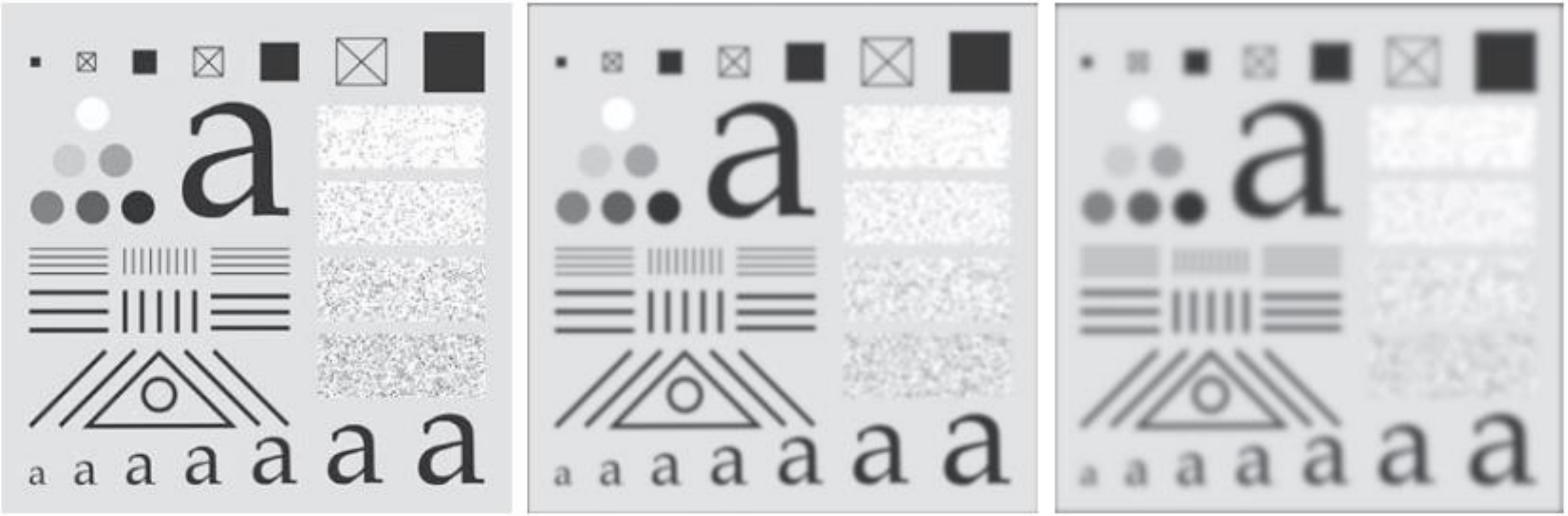
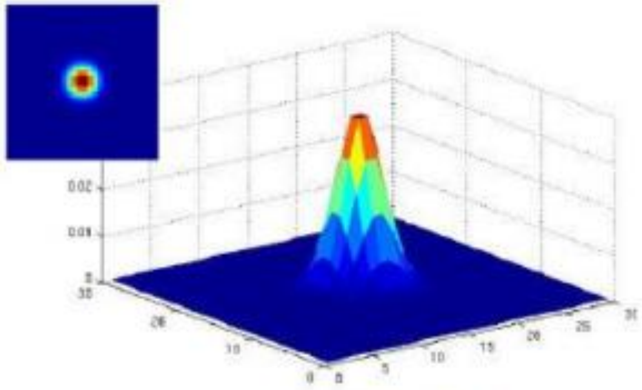


FIGURE 3.36 (a)A test pattern of size 1024×1024. (b) Result of lowpass filtering the pattern with a Gaussian kernel of size 21×21, with standard deviations $\sigma = 3.5$. (c) Result of using a kernel of size 43×43, with $\sigma = 7$. This result is comparable to Fig. 3.33(d). We used $K = 1$ in all cases.

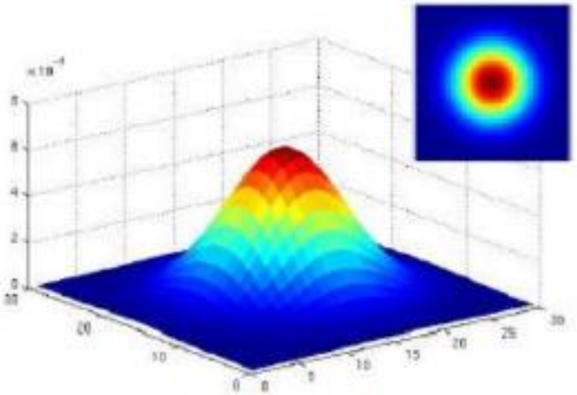
a b c

FIGURE 3.37 (a) Result of filtering Fig. 3.36(a) using a Gaussian kernels of size 43×43, with $\sigma = 7$. (b) Result of using a kernel of 85×85, with the same value of σ. (c) Difference image.

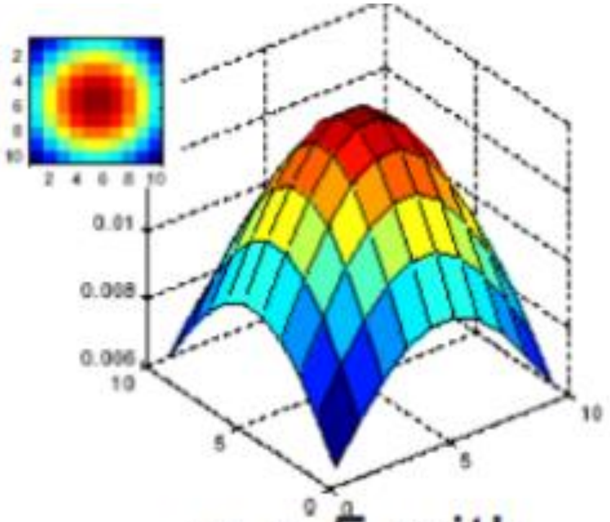
There is nothing to be gained by using a Gaussian kernel larger than $\lceil 6\sigma \rceil \times \lceil 6\sigma \rceil$. for image processing.



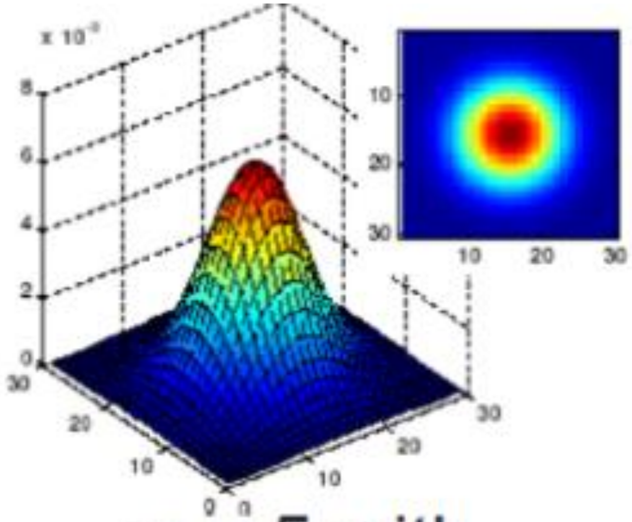
σ = 2 with
30 x 30
kernel



σ = 5 with
30 x 30
kernel

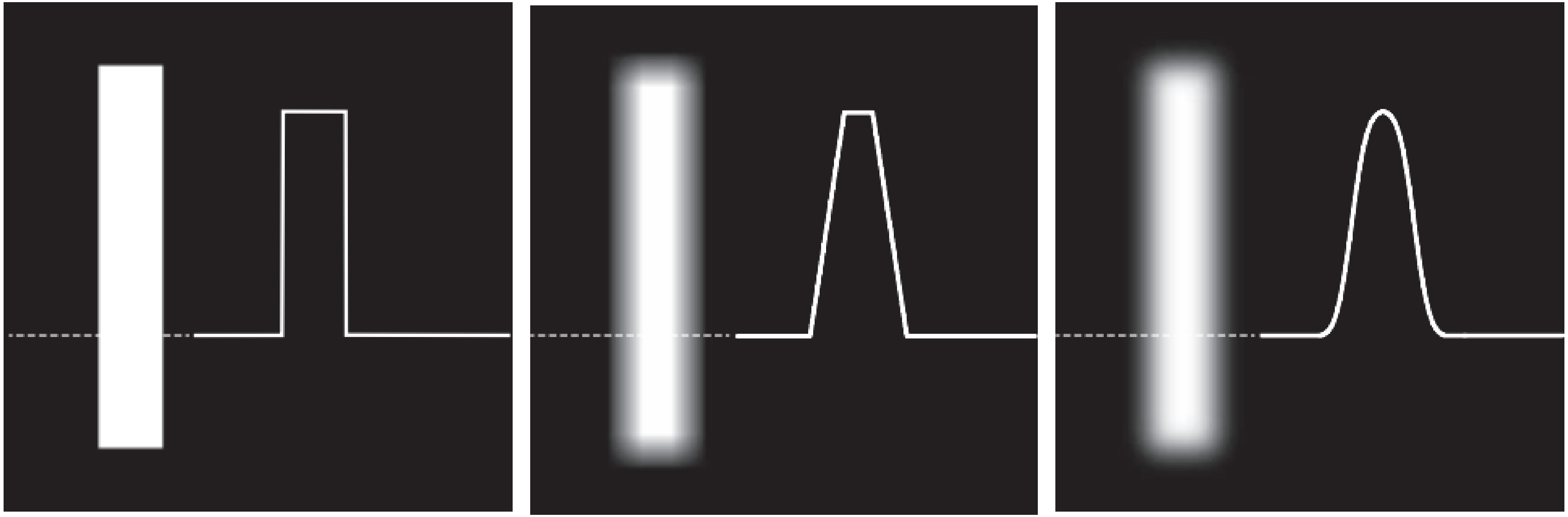


σ = 5 with
10 x 10
kernel



σ = 5 with
30 x 30
kernel

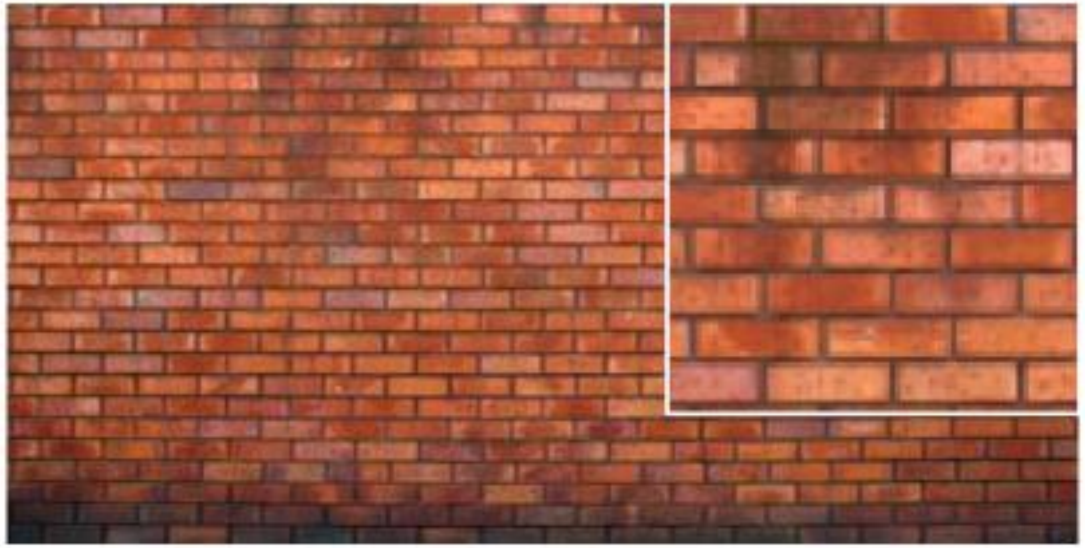
Box Kernel VS Gaussian kernel



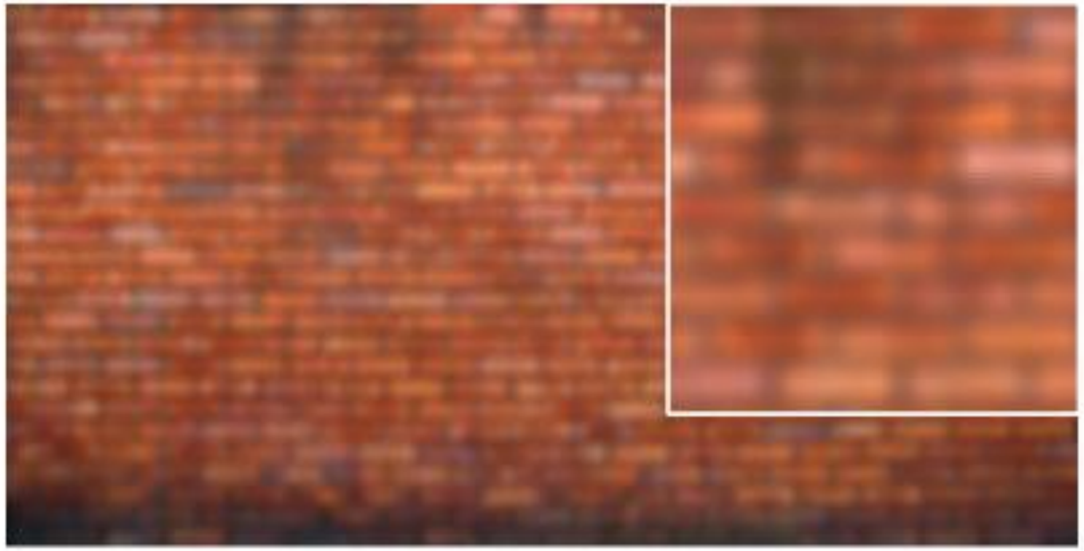
a b c

FIGURE 3.38 (a) Image of a white rectangle on a black background, and a horizontal intensity profile along the scan line shown dotted. (b) Result of smoothing this image with a box kernel of size 71×71, and corresponding intensity profile. (c) Result of smoothing the image using a Gaussian kernel of size 151×151, with $K = 1$ and $\sigma = 25$. Note the smoothness of the profile in (c) compared to (b). The image and rectangle are of sizes 1024×1024 and 768×128 pixels, respectively.

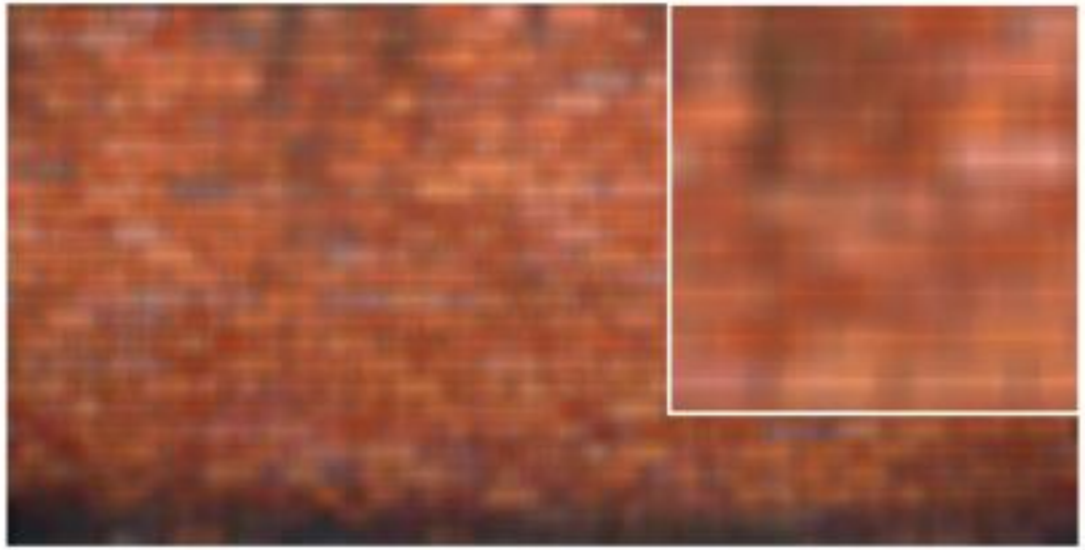
Gaussian vs box filtering



original



7x7 Gaussian



7x7 box

TYPES OF PADDING

- Types of padding
  - Zero padding
  - Mirror/reflect/symmtery padding
  - Replicate padding
- Advantages of Padding
  - Maintain image size
  - Reduce Border effects
  - Increase CNN Accuracy

TYPES OF PADDING

1 Zero Padding

0	0	0	0	0
0	1	2	3	0
0	4	5	6	0
0	7	8	9	0
0	0	0	0	0

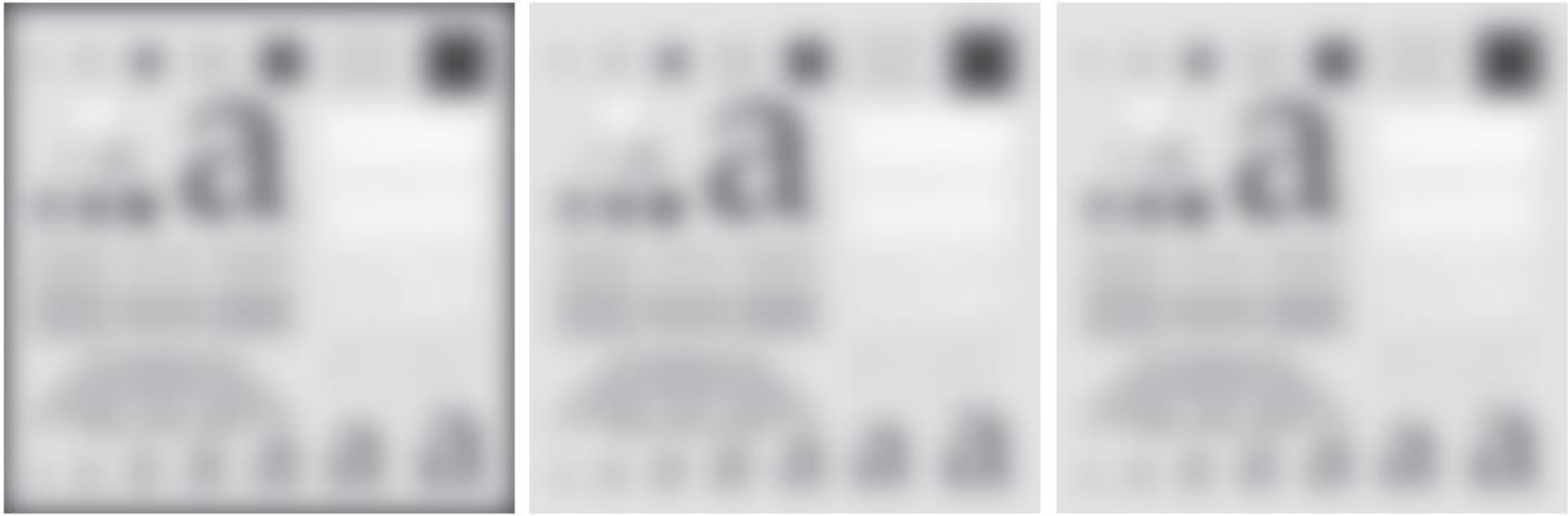
2 Replication Padding

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

3 Reflection Padding

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

TYPES OF PADDING



a b c

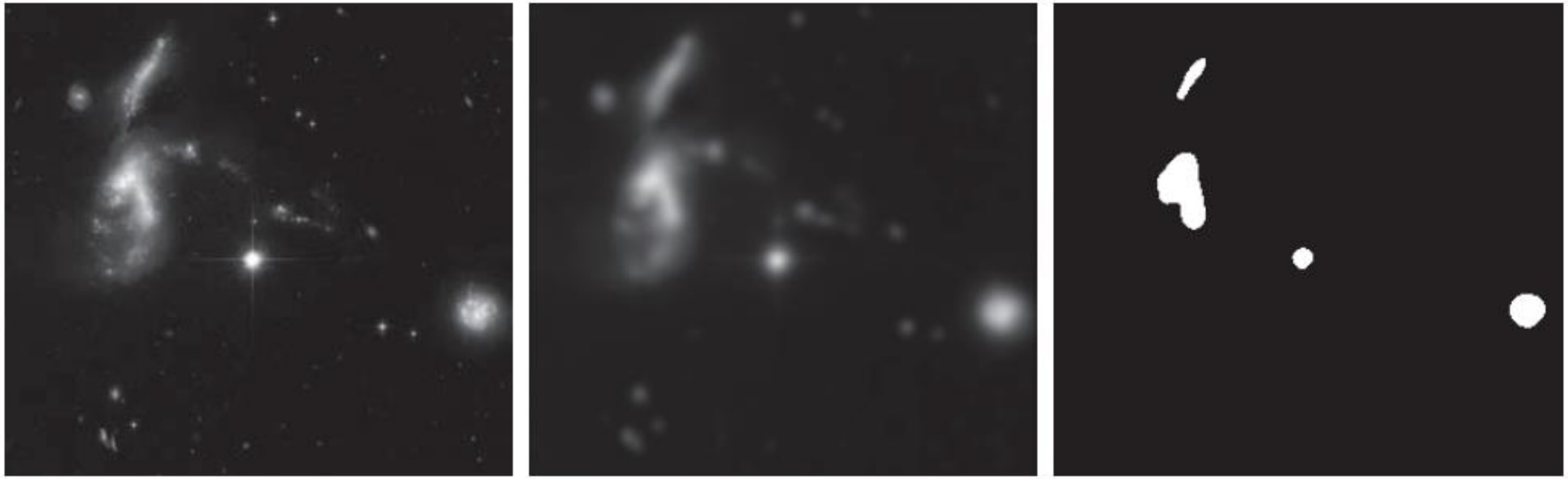
FIGURE 3.39 Result of filtering the test pattern in Fig. 3.36(a) using (a) zero padding, (b) mirror padding, and (c) replicate padding. A Gaussian kernel of size 187×187, with $K = 1$ and $\sigma = 31$ was used in all three cases.

SMOOTHING PERFORMANCE AS A FUNCTION OF KERNEL AND IMAGE SIZE

a b c

FIGURE 3.40 (a) Test pattern of size 4096 × 4096 pixels. (b) Result of filtering the test pattern with the same Gaussian kernel used in Fig. 3.39. (c) Result of filtering the pattern using a Gaussian kernel of size 745 × 745 elements, with $K = 1$ and $\sigma = 124$. Mirror padding was used throughout.

USING LOW PASS FILTERING AND THRESHOLDING FOR REGION EXTRACTION

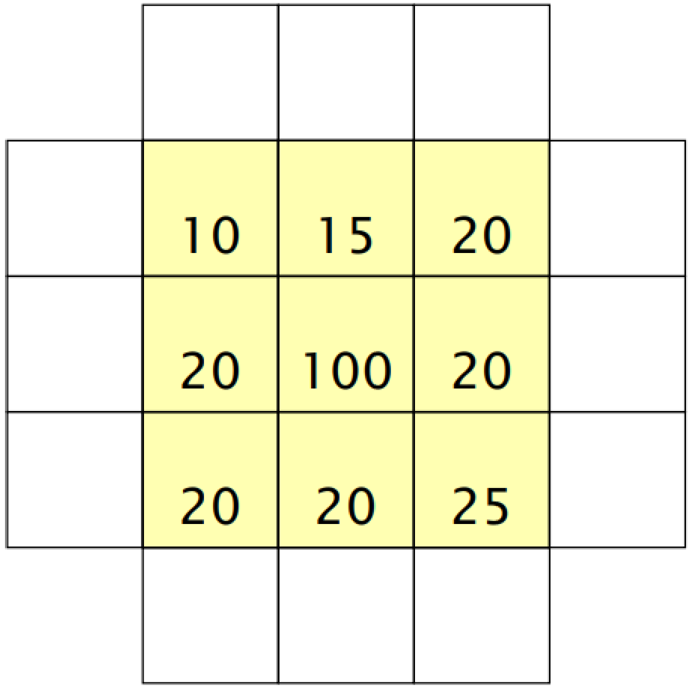


- Image size: 2566 x 2758, intensities scaled in range [0,1]
- The result of filtering the original image with a Gaussian kernel of size 151x151(approximately 6% of the image width) and standard deviation s = 25

ORDER-STATISTIC (NONLINEAR) FILTERS

- Spatial filters whose response is based on ordering (ranking) the pixels contained in the region encompassed by the filter.
- Smoothing is achieved by replacing the value of the center pixel with the value determined by the ranking result.
- The best-known filter in this category is the median filter.
- Median filters provide excellent noise reduction capabilities for certain types of random noise, with considerably less blurring than linear smoothing filters of similar size.

PROCESS OF MEDIAN FILTER



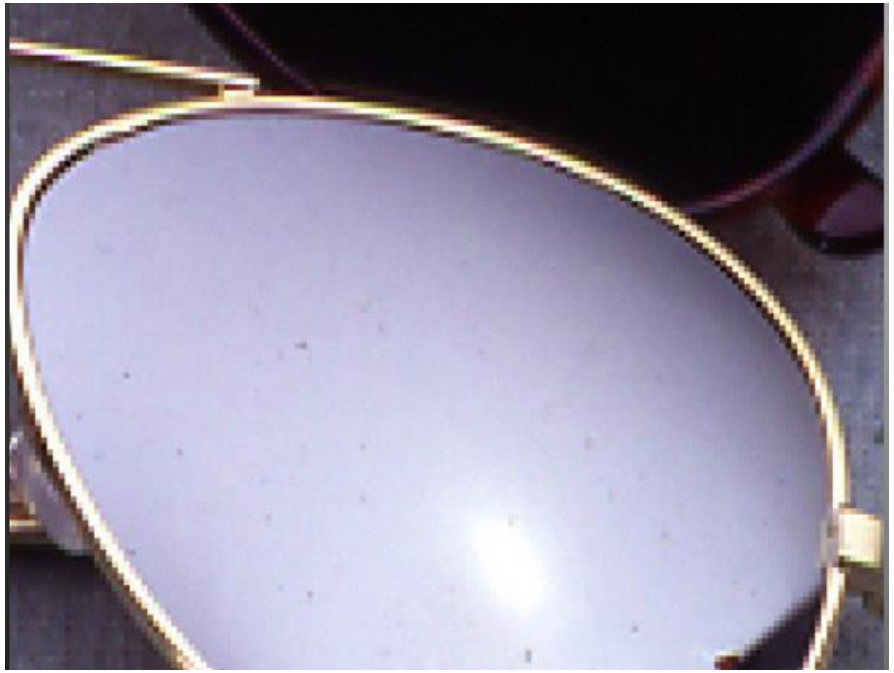
10, 15, 20, 20, 20, 20, 20, 25, 100

5th

- Sort the values of the pixel in our region
- In the MxN mask the median is

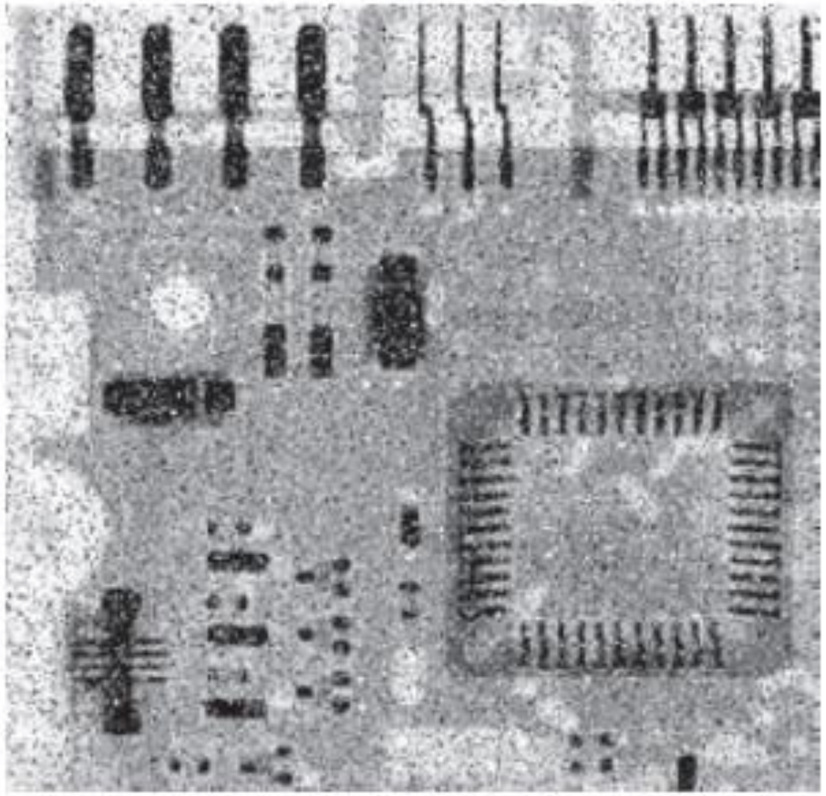
$$\frac{(MxN)+1}{2}$$

EXAMPLES

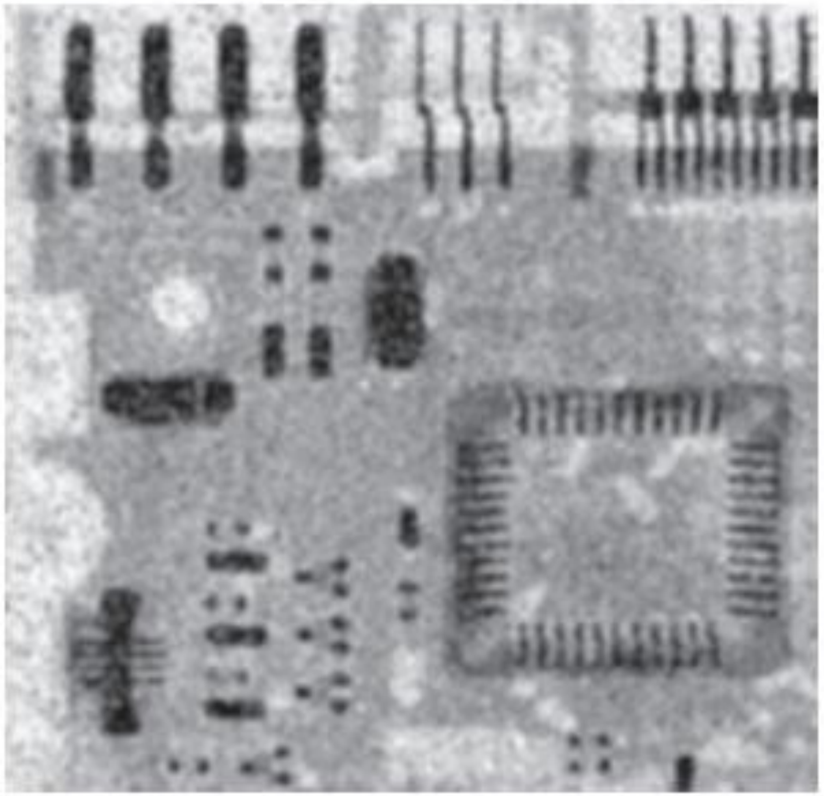


Noisy Image

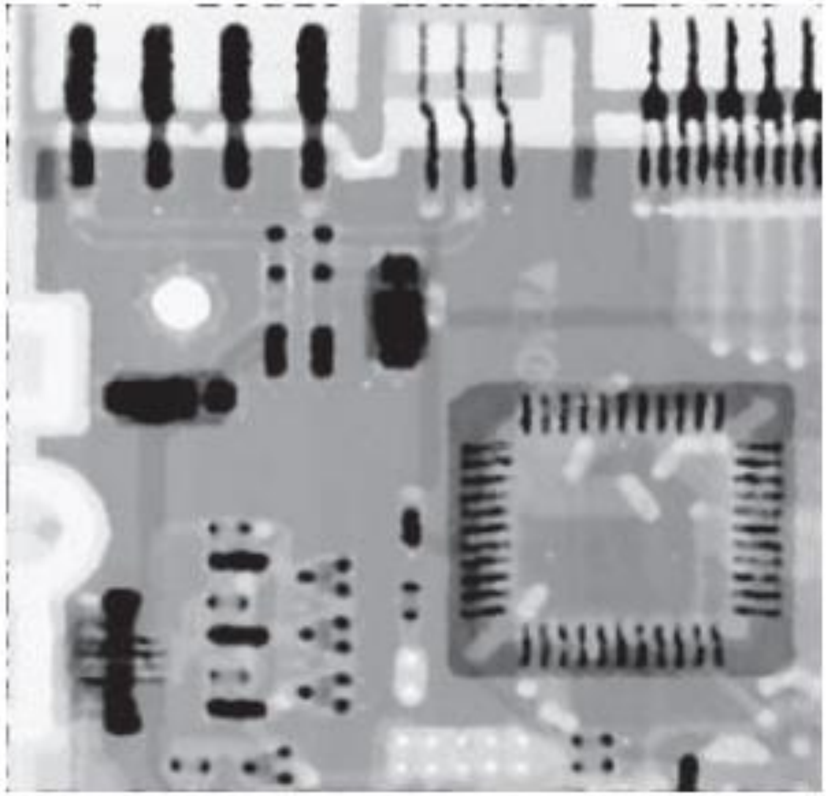
Noise removed by using median filter



(a) X-ray image of a circuit board, corrupted by salt-and-pepper noise.

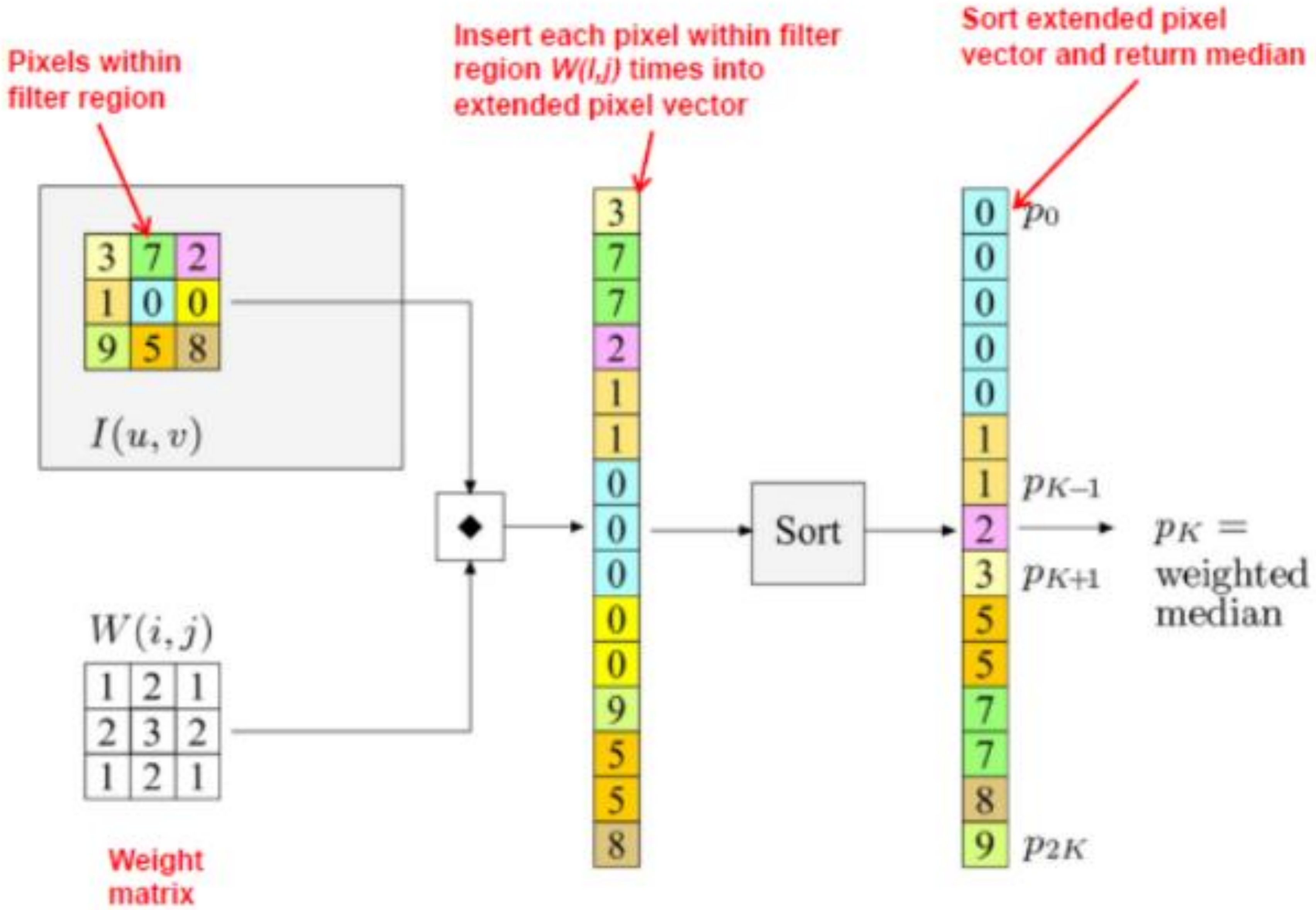


(b) Noise reduction using a 19 x19 Gaussian lowpass filter kernel with s = 3.



(c) Noise reduction using a 7 7 × median filter.

Weighted Median Filter



Comparison – Average and Median Filter

Averaging Filter
(11 x 11)

Median Filter
(11 x 11)

MIN AND MAX FILTER

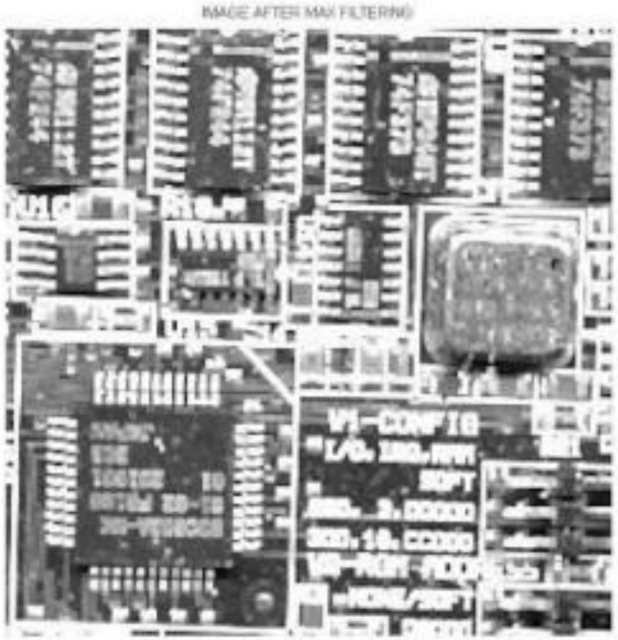
- MIN FILTER:
  - The transformation replaces the central pixel with the darkest one in the running window.
  - For example, if you have text that is lightly printed, the minimum filter makes letters thicker.

- MAX FILTER
  - The maximum filter replaces it with the lightest one.
  - For example, if you have a text string drawn with a thick pen, you can make the sign skinnier.

EXAMPLES

Image after MIN FILTERING

Image after MAX FILTERING



Median

Averging

MIN

MAX

SHARPENING (HIGHPASS) FILTERS

SHARPENING (HIGHPASS) FILTERS

- Sharpening highlights transitions in intensity.
- Sharpening is often referred to as highpass filtering. In this case, high frequencies (which are responsible for fine details) are passed, while low frequencies are attenuated or rejected.
- Averaging is analogous to integration, Sharpening could be accomplished by spatial differentiation.

smooth sharpen

Sharpening – highlight the transitions in intensity by differentiation

Smoothing – blur the transitions by summation

FOUNDATION

- We are interested in the behavior of these derivatives in areas of constant gray level(flat segments), at the onset and end of discontinuities(step and ramp discontinuities), and along gray-level ramps.

- These types of discontinuities can be noise points, lines, and edges.

PROPERTIES OF FIRST DERIVATIVE

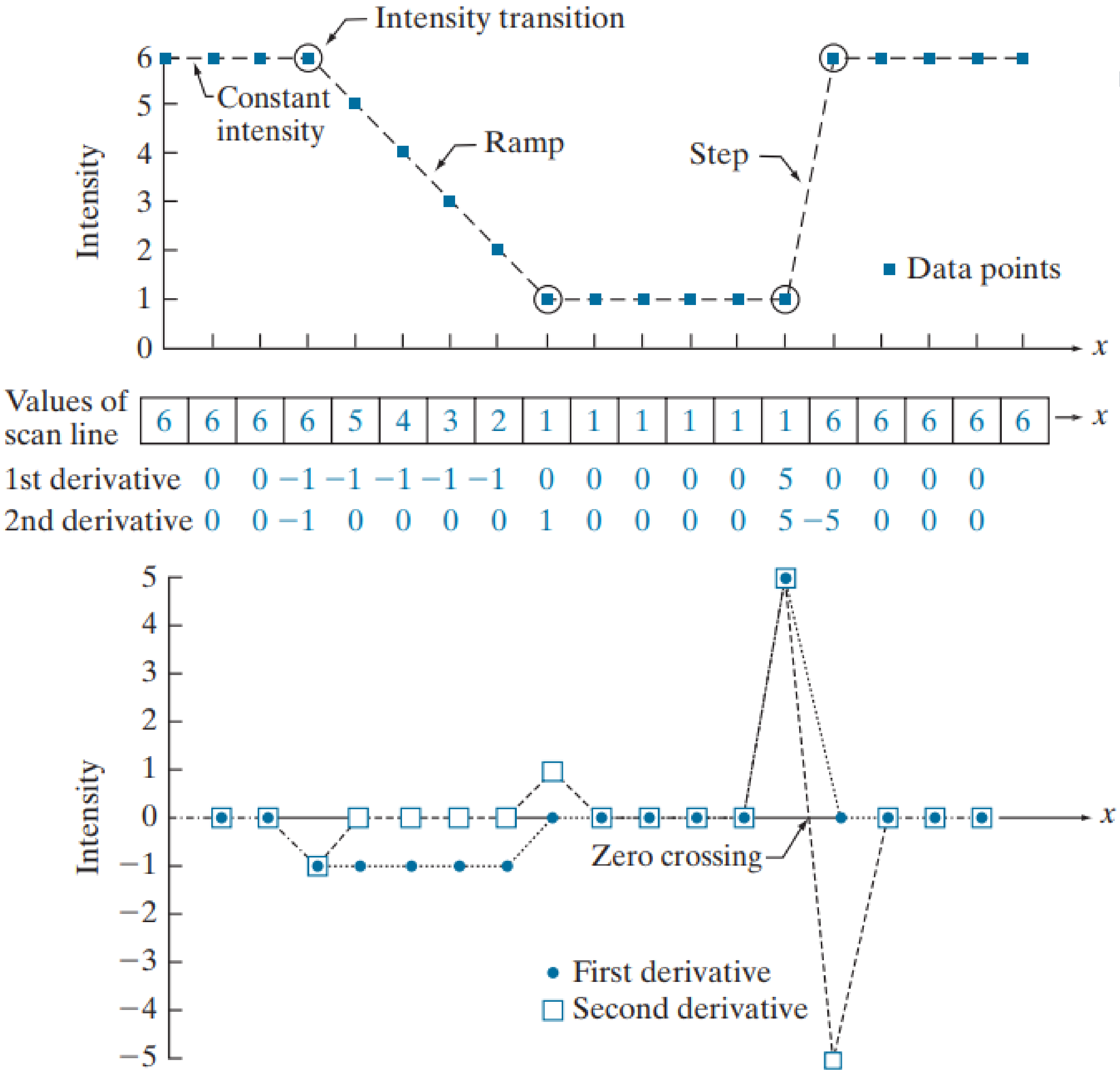
$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

- Must be zero in flat segments
- Must be nonzero at the onset of a gray-level step or ramp
- Must be nonzero along ramps.

PROPERTIES OF SECOND DERIVATIVE

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

- Must be zero in flat areas
- Must be nonzero at the onset and end of a gray-level step or ramp
- Must be zero along ramps of constant slope


Intensity transition
Constant intensity
Ramp
Step
Data points
Intensity
x
Values of scan line
6 6 6 6 5 4 3 2 1 1 1 1 1 1 6 6 6 6 6
1st derivative 0 0 −1 −1 −1 −1 −1 0 0 0 0 0 5 0 0 0 0
2nd derivative 0 0 −1 0 0 0 0 1 0 0 0 0 5 −5 0 0 0
Zero crossing
First derivative
Second derivative

THE LAPLACIAN

- USING THE SECOND DERIVATIVE FOR IMAGE SHARPENING

- The simplest isotropic derivative operator is the Laplacian, is defined as

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

DISCRETE FORM OF DERIVATIVES

$f(x-1,y)$	$f(x,y)$	$f(x+1,y)$

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

$f(x,y-1)$
$f(x,y)$
$f(x,y+1)$

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

2 DIMENSIONAL LAPLACIAN

- The digital implementation of the 2- Dimensional Laplacian is obtained by summing 2 components

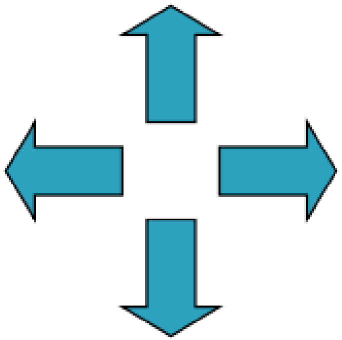
$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial x^2}$$

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

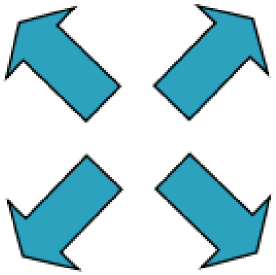
	1	
1	-4	1
	1	

LAPLACIAN

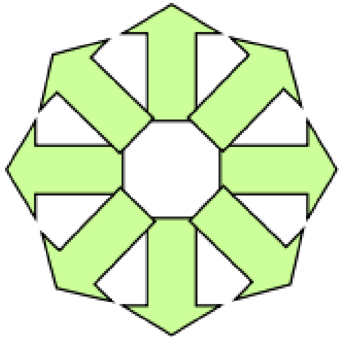
0	1	0
1	−4	1
0	1	0



1	0	1
0	−4	0
1	0	1

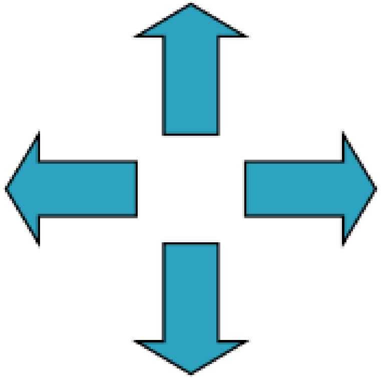


1	1	1
1	−8	1
1	1	1

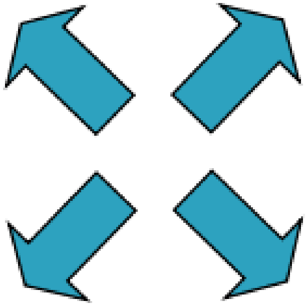


LAPLACIAN

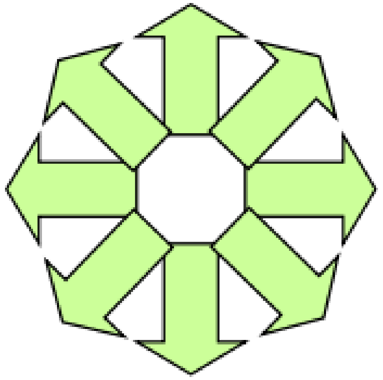
0	−1	0
−1	4	−1
0	−1	0



−1	0	−1
0	4	0
−1	0	−1



−1	−1	−1
−1	8	−1
−1	−1	−1



IMPLEMENTATION

$$g(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y) \\ f(x,y)+\nabla^2 f(x,y) \end{cases}$$

If the center coefficient is negative

If the center coefficient is positive

$$g(x,y)=f(x,y)+c\left[\nabla^2 f(x,y)\right]$$

(a) Blurred image of the North Pole of the moon.

(b) Laplacian image obtained using the kernel.

(c) Image sharpened using equation on previous slide with $c = -1$.

(d) Image sharpened using the same procedure, but with the kernel that enhance diagonals also.

THE LAPLACIAN IMAGE

The Laplacian image from previous example, scaled to the full [0, 255] range of intensity values. Black pixels correspond to the most negative value in the unscaled Laplacian image, grays are intermediate values, and white pixels corresponds to the highest positive value.

UNSHARP MASKING

- Subtracting an unsharp (smoothed) version of an image from the original image to sharpen images. This process, called unsharp masking, consists of the following steps:
  - Blur the original image.
  - Subtract the blurred image from the original (the resulting difference is called the mask.)
  - Add the mask to the original.

UNSHARP MASKING AND HIGH BOOST FILTERING

$\bar{f}(x, y)$ denote the blurred image, the mask in equation form is given by:

$$g_{\text{mask}}(x, y) = f(x, y) - \bar{f}(x, y)$$

Then we add a weighted portion of the mask back to the original image:

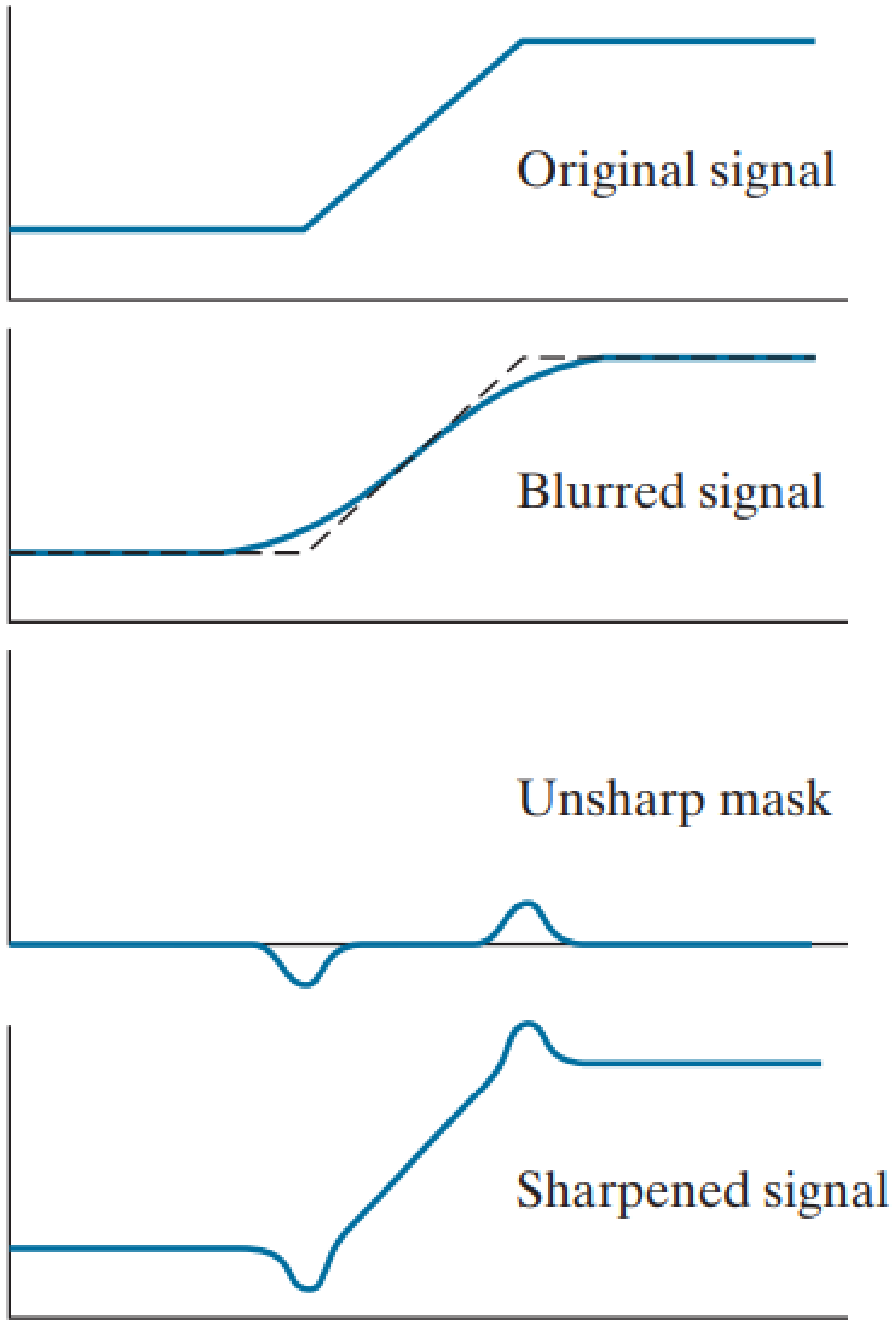
$$g(x, y) = f(x, y) + k\,g_{\text{mask}}(x, y)$$

When k = 1 we have **unsharp masking**.
When k > 1, the process is referred to as **highboost filtering**.
Choosing k < 1 reduces the contribution of the unsharp mask.

Unsharp mask is similar to what we would obtain using a second-order derivative.

Observe that negative values were added to the original. Thus, it is possible for the final result to have negative intensities if the original image has any zero values.

Negative values cause dark halos around edges that can become objectionable if k is too large.



a
b
c
d

FIGURE 3.48 1-D illustration of the mechanics of unsharp masking. (a) Original signal. (b) Blurred signal with original shown dashed for reference. (c) Unsharp mask. (d) Sharpened signal, obtained by adding (c) to (a).

a b c
d e

FIGURE 3.49 (a) Original image of size 600×259 pixels. (b) Image blurred using a 31×31 Gaussian lowpass filter with $\sigma = 5$. (c) Mask. (d) Result of unsharp masking using Eq. (3-56) with $k = 1$. (e) Result of highboost filtering with $k = 4.5$.

FIRST-ORDER DERIVATIVES FOR IMAGE SHARPENING—THE GRADIENT

- First Derivatives in image processing are implemented using the magnitude of the gradient.
- The gradient of function f(x,y) is

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

GRADIENT

- The magnitude of this vector is given by

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

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

ROBERT'S METHOD

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

denote the intensities of pixels in a 3×3 region

The simplest approximations to a first order derivative.

$$g_x = (z_9 - z_5) \quad \text{and} \quad g_y = (z_8 - z_6)$$

$$M(x, y) = \left[(z_9 - z_5)^2 + (z_8 - z_6)^2 \right]^{1/2}$$

Roberts cross-gradient operators

−1	0
0	1

0	−1
1	0

$$M(x, y) \approx |z_9 - z_5| + |z_8 - z_6|$$

SOBEL'S METHOD

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- Mask of even size are awkward to apply.
- The smallest filter mask should be 3x3.

- The difference between the third and first rows of the 3x3 mage region approximate derivative in x-direction

- The difference between the third and first column approximate derivative in y-direction.

SOBEL'S METHOD

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

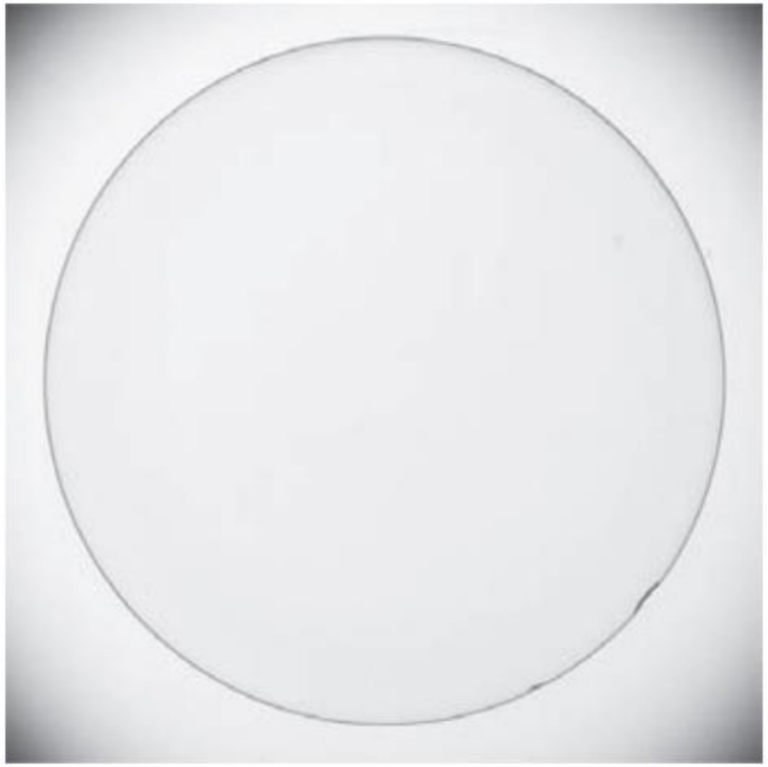
$$M(x,y) = \left[g_x^2 + g_y^2\right]^{\frac{1}{2}} = \Big[\left[(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)\right]^2 + \left[(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)\right]^2\Big]^{\frac{1}{2}}$$

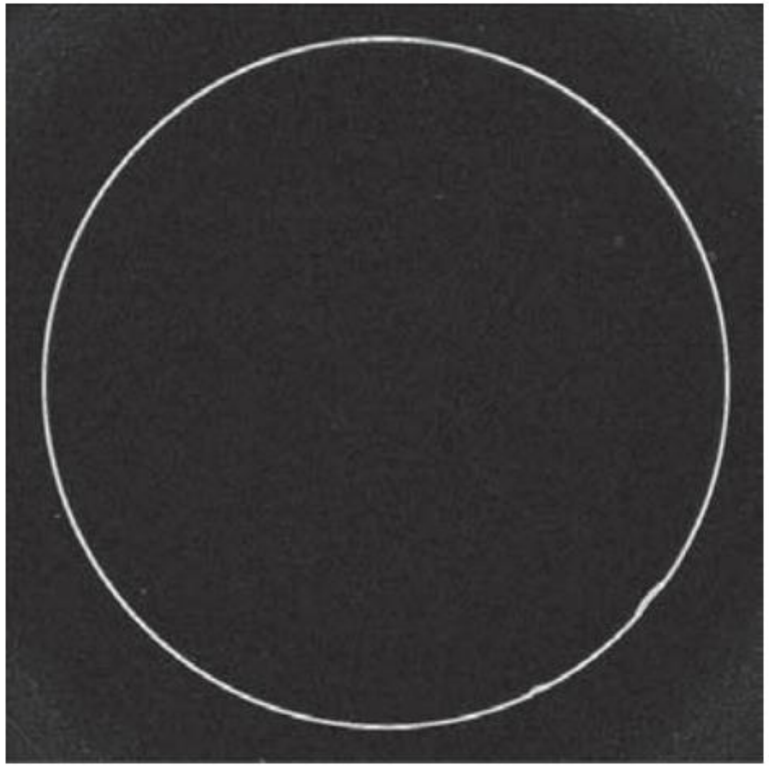
USING GRADIENT FOR EDGE ENHANCEMENT

An optical image of a contact lens. and the gradient obtained using the two Sobel kernels

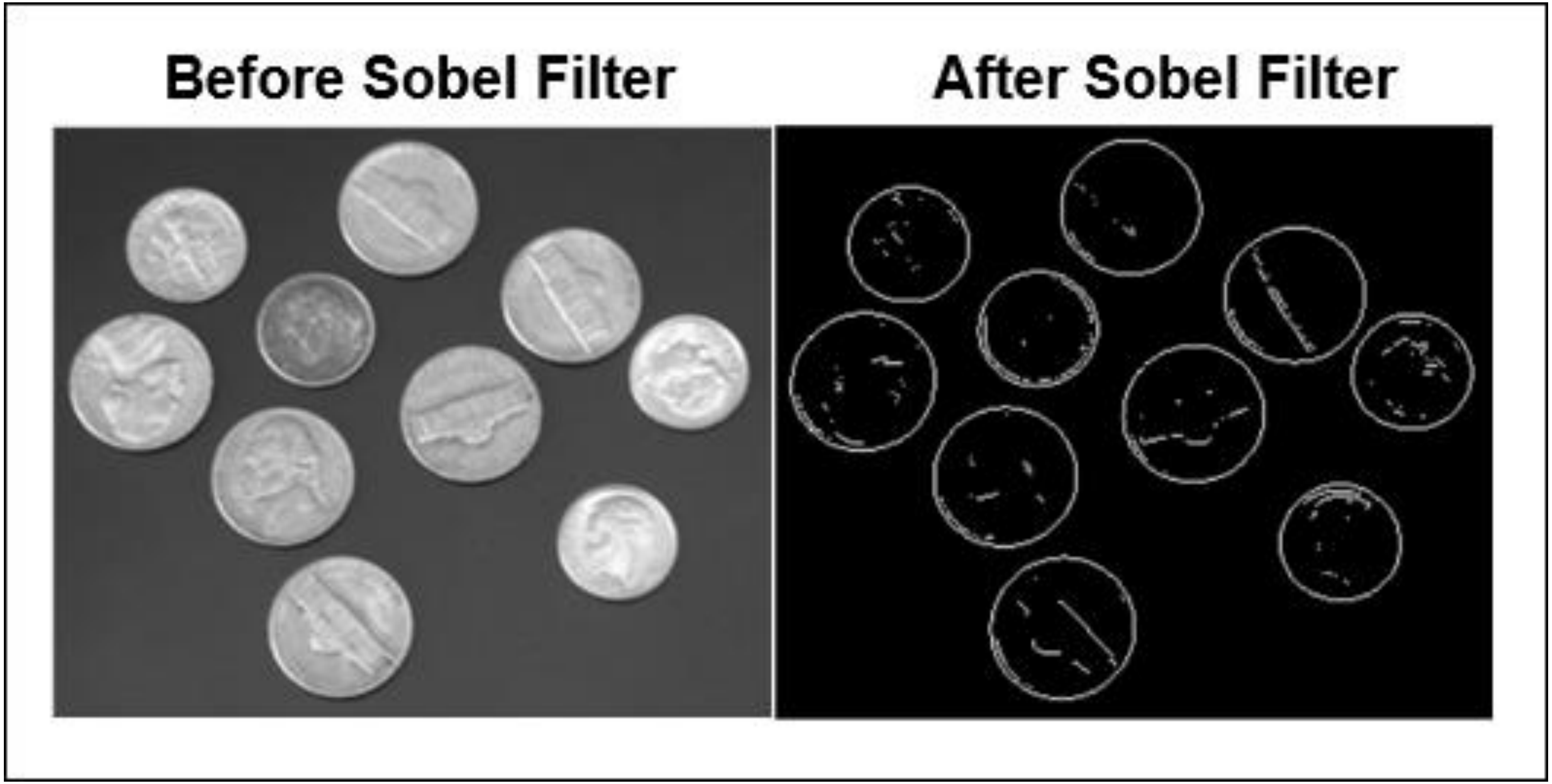
The gradient can be used also to highlight small specs that may not be readily visible in a gray-scale image.

The ability to enhance small discontinuities in an otherwise flat gray field is another important feature of the gradient.





SOBEL FILTER FOR EDGE DETECTION



EXAMPLE

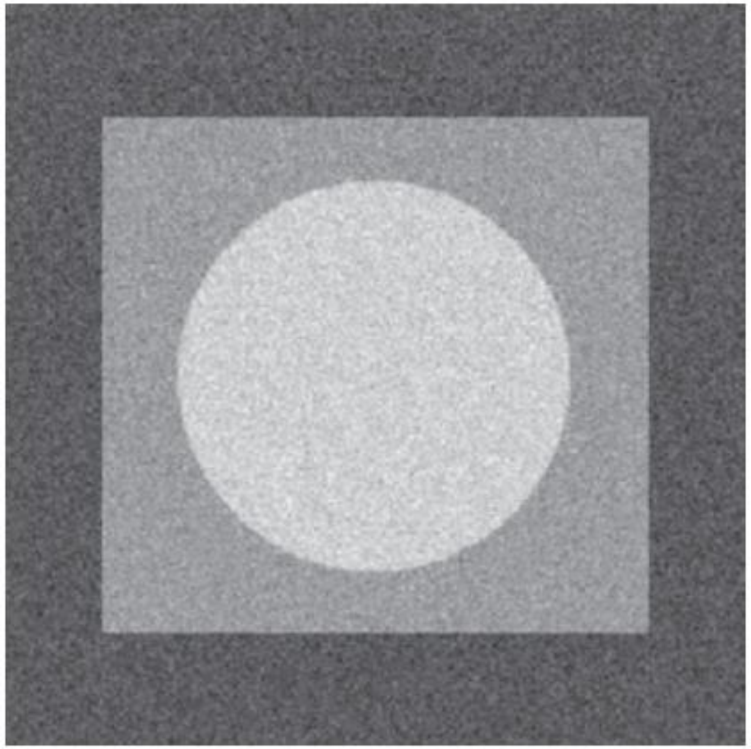
SOURCES OF NOISE

- Image acquisition (digitization)
- Image transmission
- Spatial properties of noise
  - Statistical behavior of the gray-level values of pixels
  - correlation with the image
- Probability density function defines the distribution of noise in an image

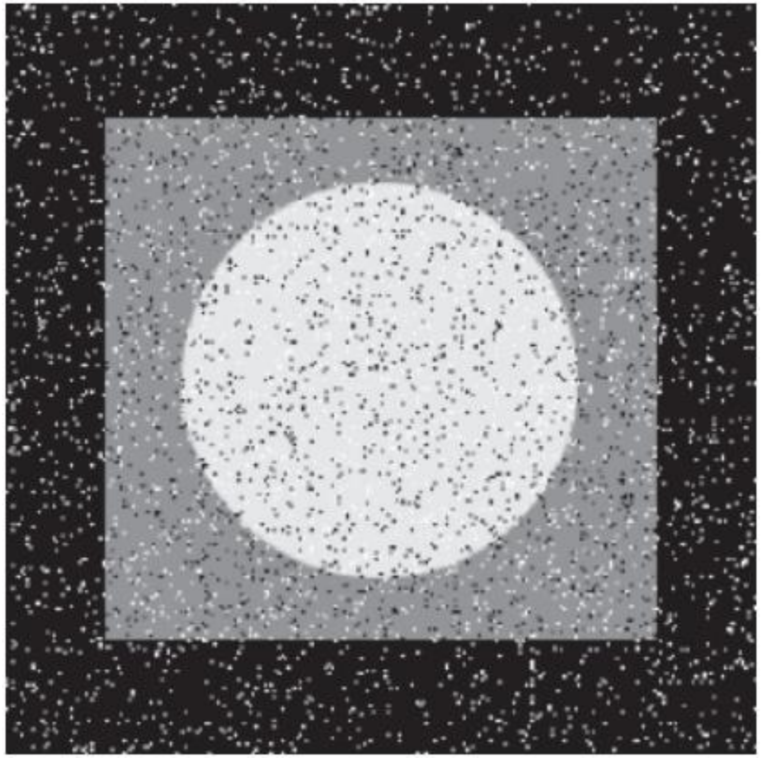
TYPES OF NOISES

- We shall be concerned with the statistical behavior of the intensity values in the noise component of the model.
- These may be considered random variables, characterized by a probability density function (PDF).
- Types:
  - Gaussian Noise
  - Salt and Pepper Noise

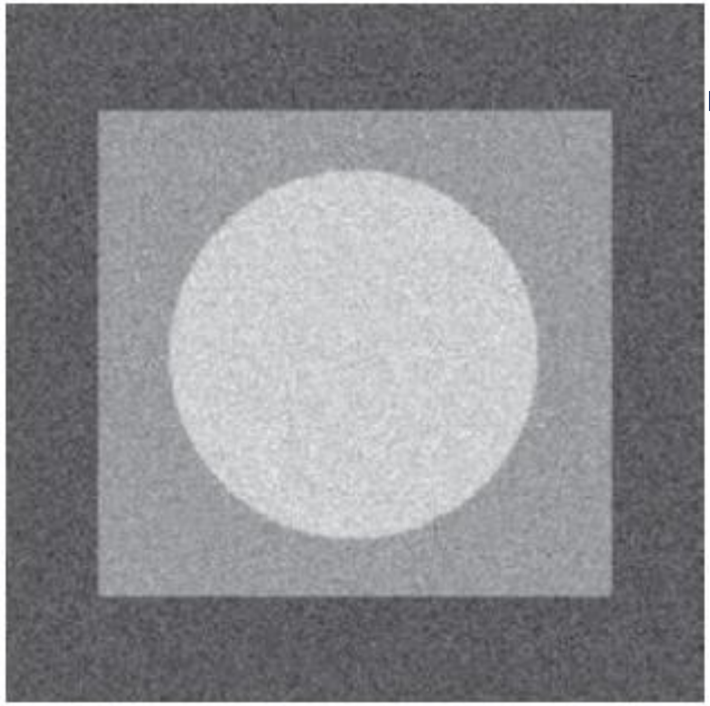
Original Image

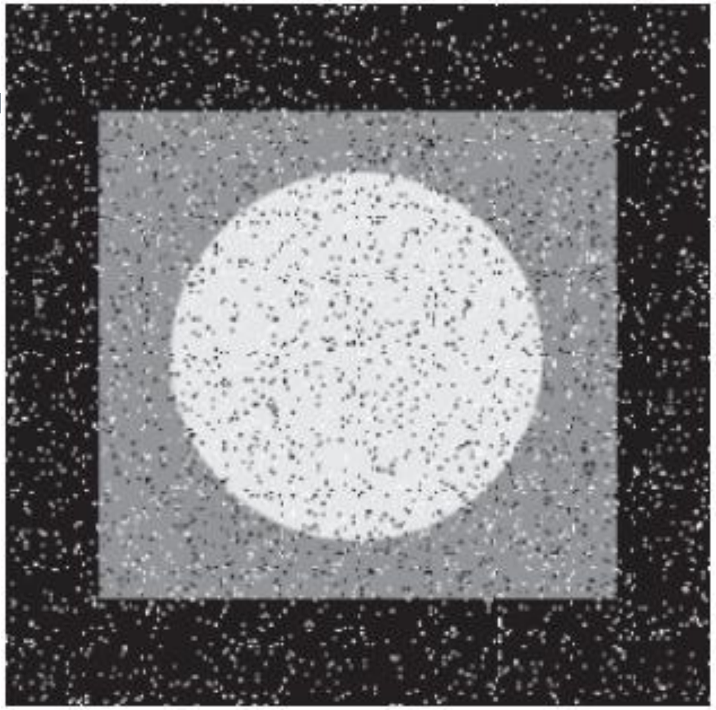


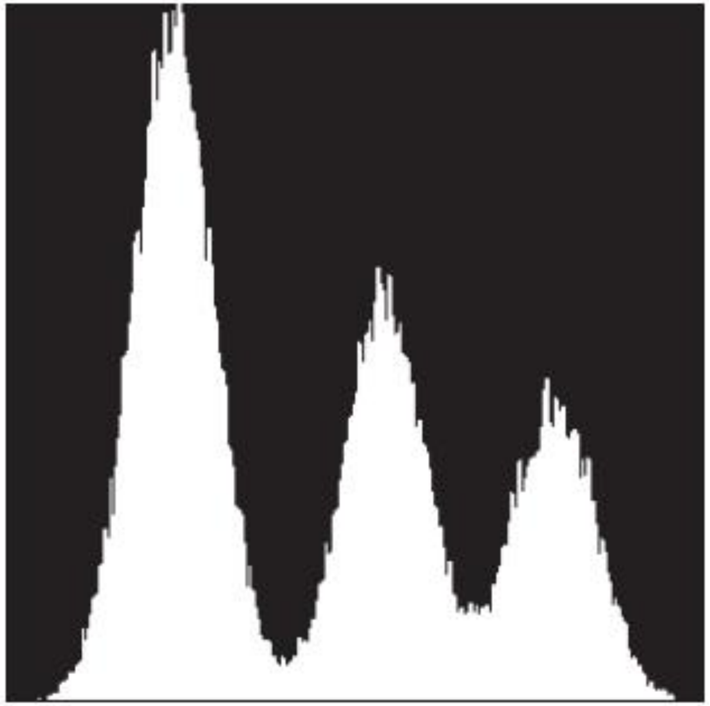
Gaussian Noise



Salt and Pepper Noise







APPROPRIATE FILTERS FOR REDUCTION OF NOISE

- Order Statistic (Non-Linear) Filters are suitable for removing salt and pepper noise
- Averaging filters (Gaussian) are suitable for removing Gaussian noise.