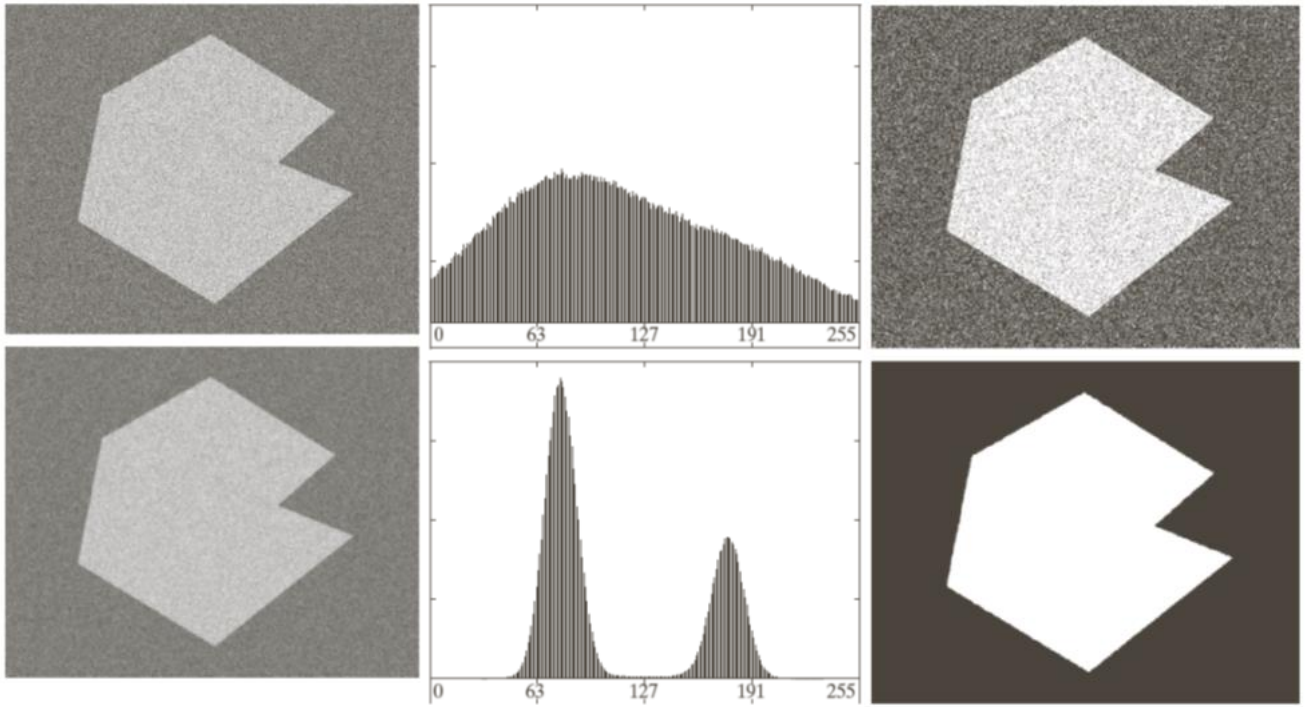
Spatial Filtering

Noise removal

Effect of Noise



Filter mask

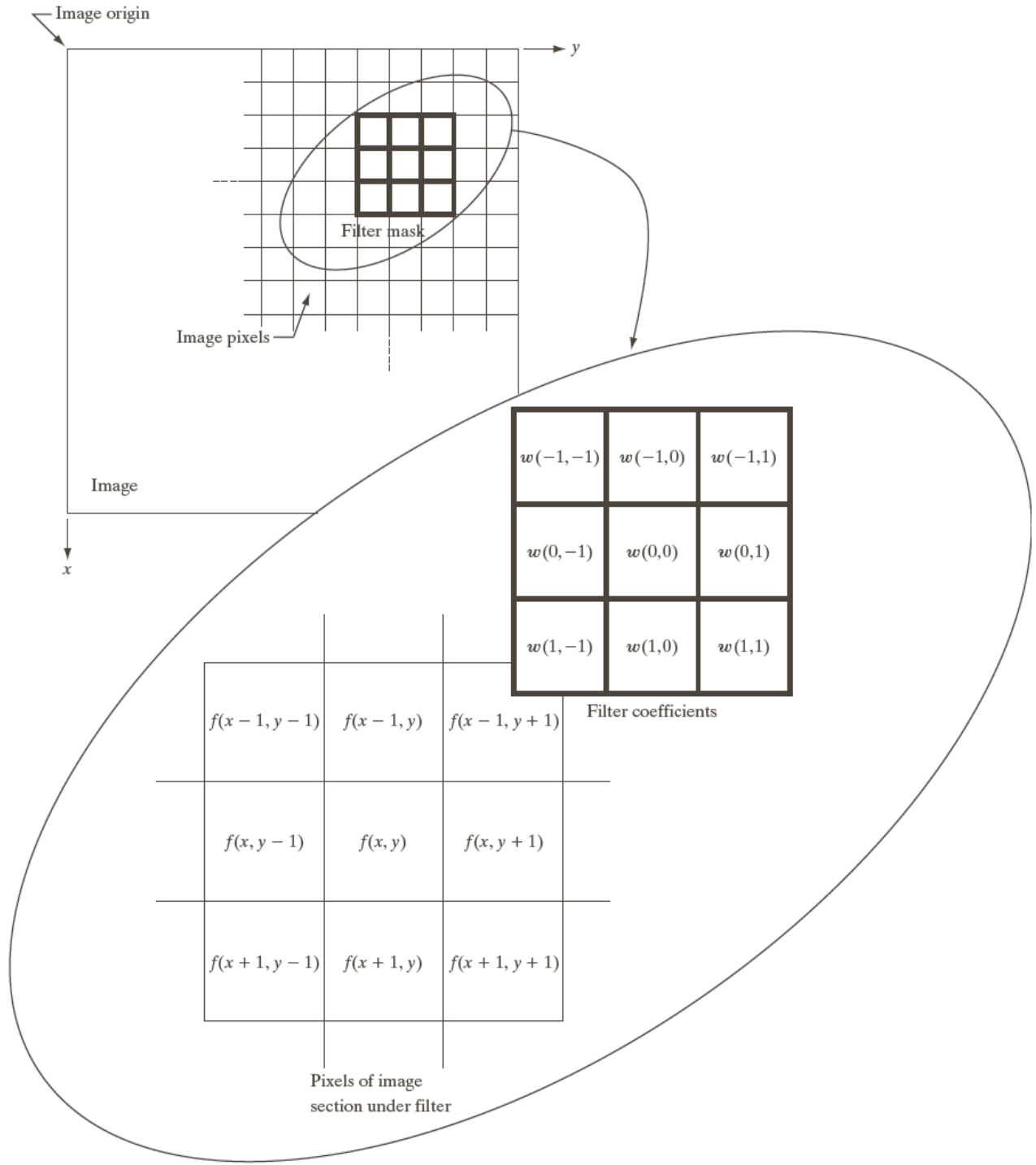


Image origin
y
Filter mask
Image pixels
Image
x
w(−1,−1) w(−1,0) w(−1,1)
w(0,−1) w(0,0) w(0,1)
w(1,−1) w(1,0) w(1,1)
Filter coefficients
f(x − 1, y − 1) f(x − 1, y) f(x − 1, y + 1)
f(x, y − 1) f(x, y) f(x, y + 1)
f(x + 1, y − 1) f(x + 1, y) f(x + 1, y + 1)
Pixels of image
section under filter

Correlation and Convolution (1D)

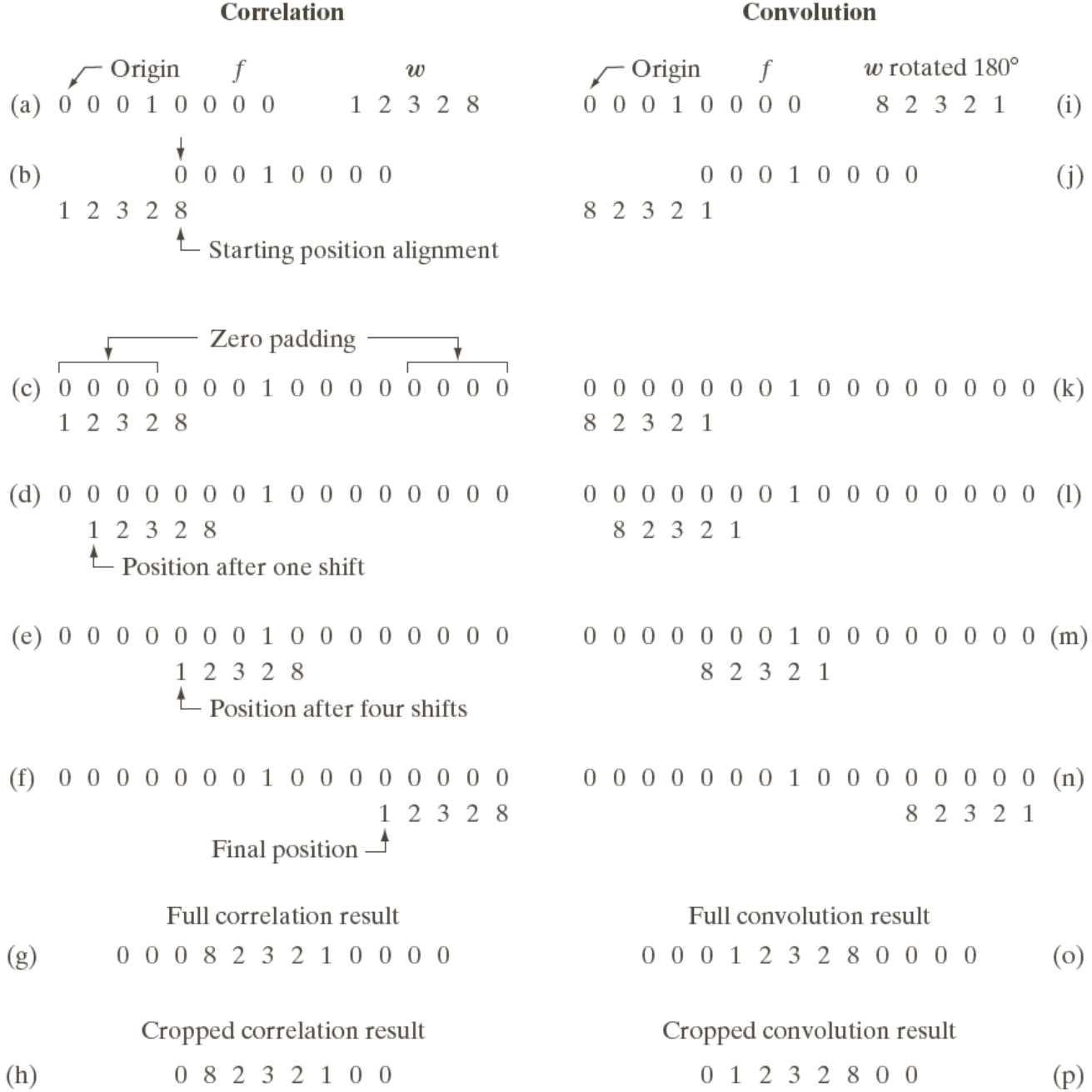


FIGURE 3.29 Illustration of 1-D correlation and convolution of a filter with a discrete unit impulse. Note that correlation and convolution are functions of *displacement*.

Correlation and Convolution

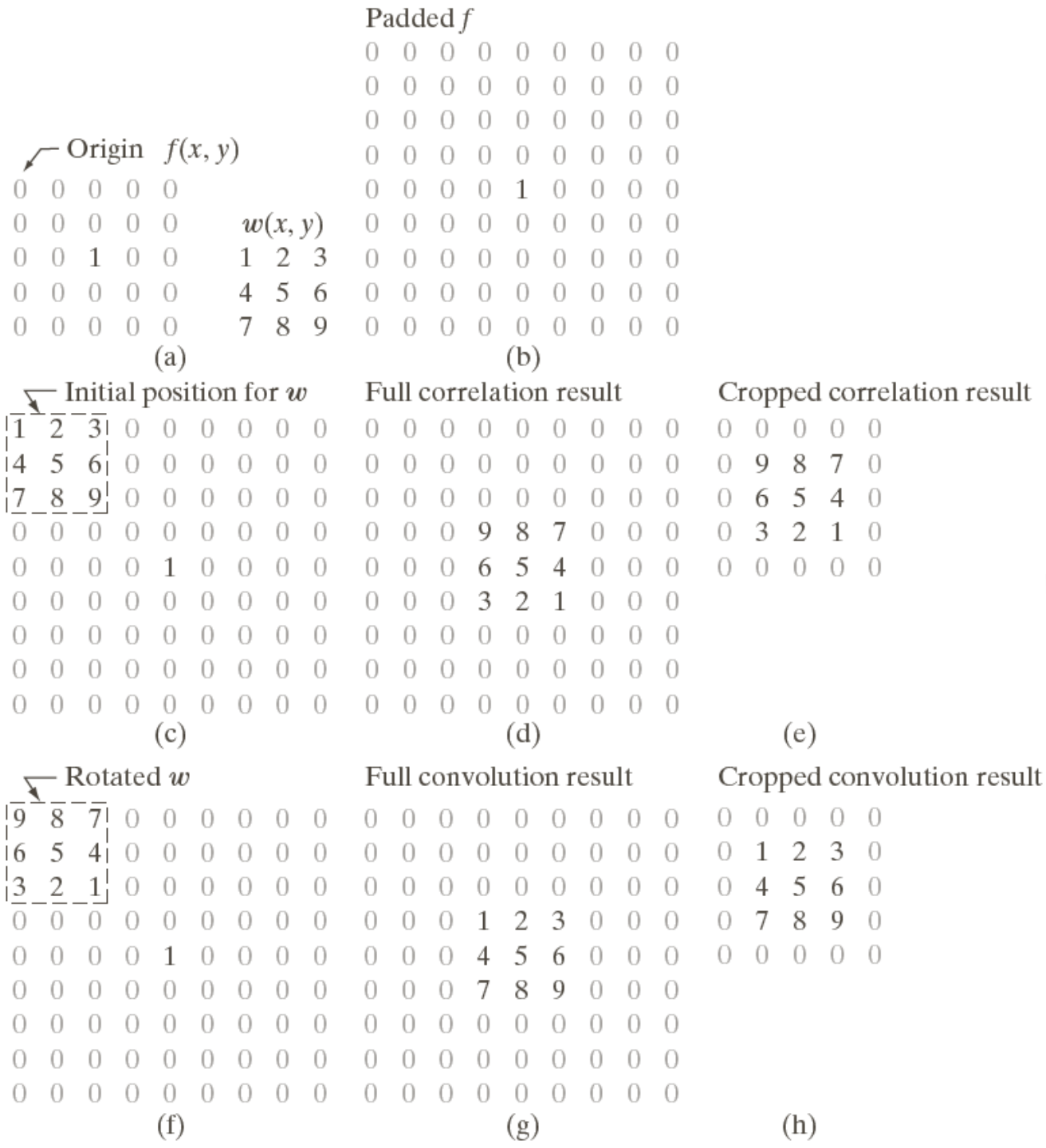
Correlation

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

Convolution

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

Correlation and Convolution (2D)



Vector representation of linear filtering

$$
\begin{aligned}
R &= w_1 z_1 + w_2 z_2 + \ldots + w_{mn} z_{mn} \\
&= \sum_{k=1}^{mn} w_k z_k \\
&= \mathbf{w}^T \mathbf{z}
\end{aligned}
$$

Vector representation of linear filtering

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

$$
\begin{aligned}
R &= w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9 \\
&= \sum_{k=1}^{9} w_k z_k \\
&= \mathbf{w}^T \mathbf{z}
\end{aligned}
$$

Smoothing Linear Filter (Average filter)

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

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

Average filter

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

Gaussian filter

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

Average filter

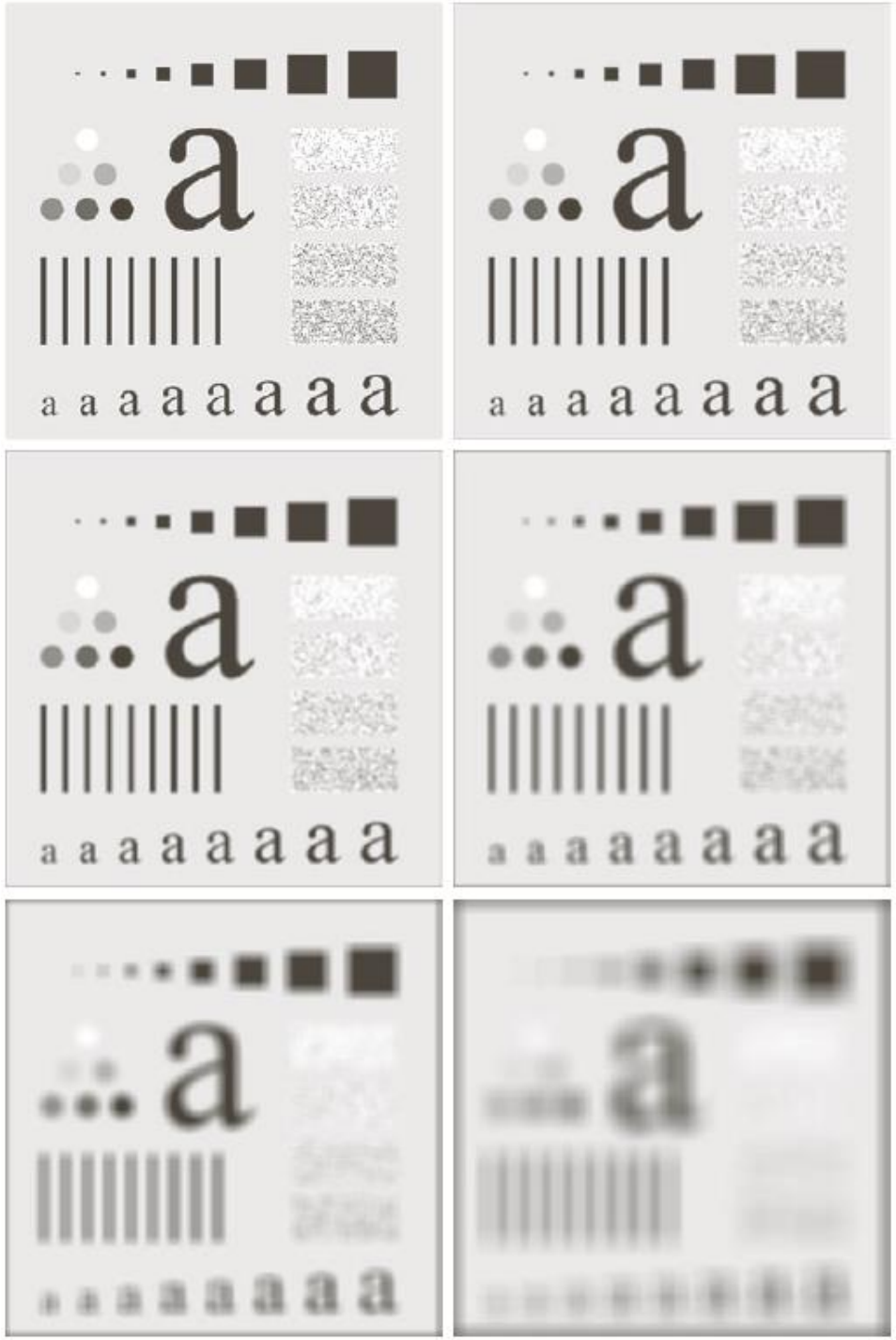
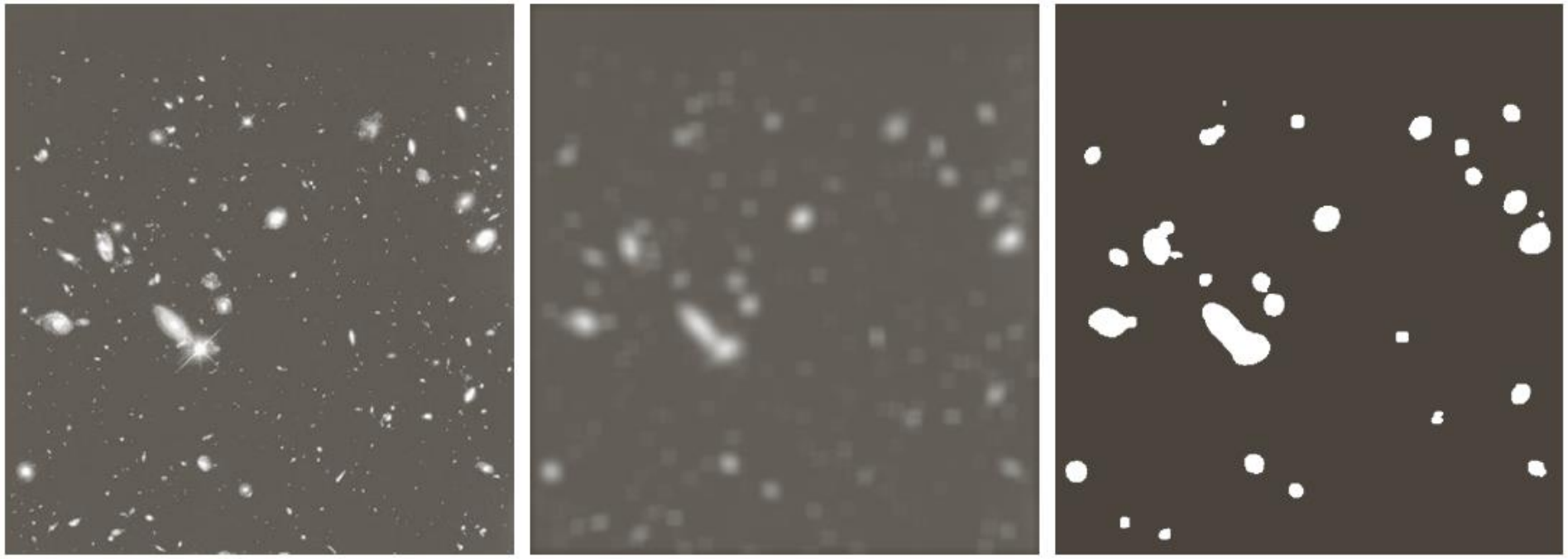


FIGURE 3.33 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $m = 3, 5, 9, 15$, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their intensity levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50 × 120 pixels.

a b
c d
e f

Average filter



a b c

FIGURE 3.34 (a) Image of size 528 × 485 pixels from the Hubble Space Telescope. (b) Image filtered with a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

Image blurring Algo

1. Define the type and size of filter mask
2. Image Padding depending on the size of filter mask
3. Apply the filter mask on input image using correlation or convolution technique
4. Crop the output image to make it of same size as input image

Image blurring in OpenCV

```
Blur(image, smoothed_image, Size(3, 3));
```

```
GaussianBlur(image, smoothed_image, Size(5, 5), 1.5);
```

Inserting Gaussian Noise

```
import cv2
import numpy as np
# Load the image
img = cv2.imread('test_image.jpg')
# Generate random Gaussian noise
mean = 0
stddev = 180
noise = np.zeros(img.shape, np.uint8)
cv2.randn(noise, mean, stddev)
# Add noise to image
noisy_img = cv2.add(img, noise)
# Save noisy image
cv2.imwrite('noisy_img.jpg', noisy_img)
```