# Spatial Filtering

Image Processing

CSE 166

Lecture 4

# Announcements

- Assignment 1 is due today, 11:59 PM
- Assignment 2 will be released today
  - Due Apr 15, 11:59 PM
- Reading
  - Chapter 3: Intensity Transformations and Spatial Filtering
    - Sections 3.4, 3.5, 3.6, and 3.8

# Image filtering and enhancement

- Spatial filtering



Example: low-pass filter

# 1D correlation and convolution

1D correlation (also called cross-correlation)

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

$$g(x) = w(x) \star f(x), \text{ where } w(x) \star f(x) = \sum_{s=-a}^{a} w(s) f(x+s)$$

1D convolution

$$g(x) = \sum_{s=-a}^{a} w(s) f(x-s)$$

$$g(x) = w(x) \bigstar f(x), \text{ where } w(x) \bigstar f(x) = \sum_{s=-a}^{a} w(s) f(x-s)$$

Equivalent to correlation with $w$ rotated 180 degrees where

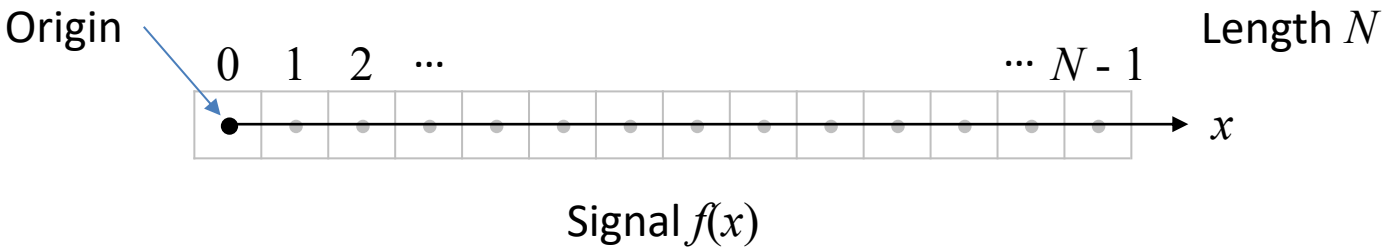
When kernel is symmetric, convolution and correlation give the same result

$w(s)$ is spatial filtering kernel (also called mask, template, and window)

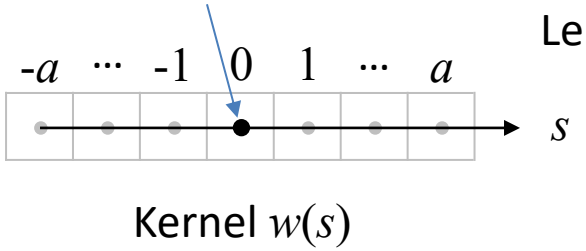
$f(x)$ is input function (signal)

$g(x)$ is output function (signal)

# 1D spatial filtering




Origin (also called anchor) at center




$a$ is a positive integer, so length is an odd positive integer

1D correlation $$w(x) \star f(x) = \sum_{s=-a}^{a} w(s) f(x+s)$$

1D convolution $$w(x) \bigstar f(x) = \sum_{s=-a}^{a} w(s) f(x-s)$$

For convolution, kernel is rotated 180 degrees

# 1D correlation and convolution

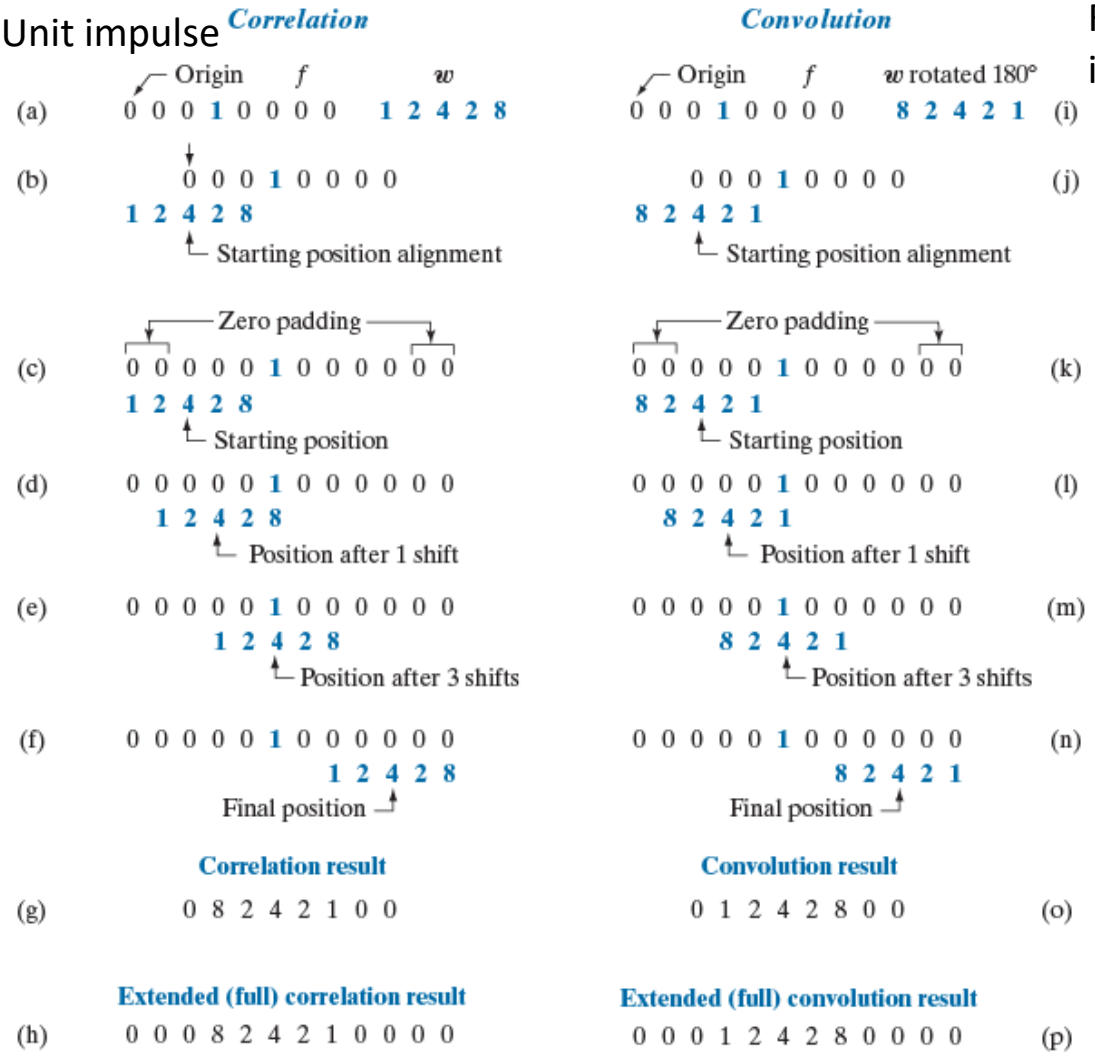



**FIGURE 3.35** Illustration of 1-D correlation and convolution of a kernel, $w$, with a function $f$ consisting of a discrete unit impulse. Note that correlation and convolution are functions of the variable $x$, which acts to *displace* one function with respect to the other. For the extended correlation and convolution results, the starting configuration places the right-most element of the kernel to be coincident with the origin of $f$. Additional padding must be used.

Unit impulse

For convolution, kernel is rotated 180 degrees

# 2D correlation and convolution

2D correlation (also called cross-correlation)

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

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

2D convolution

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

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

Equivalent to correlation with $w$ rotated 180 degrees where

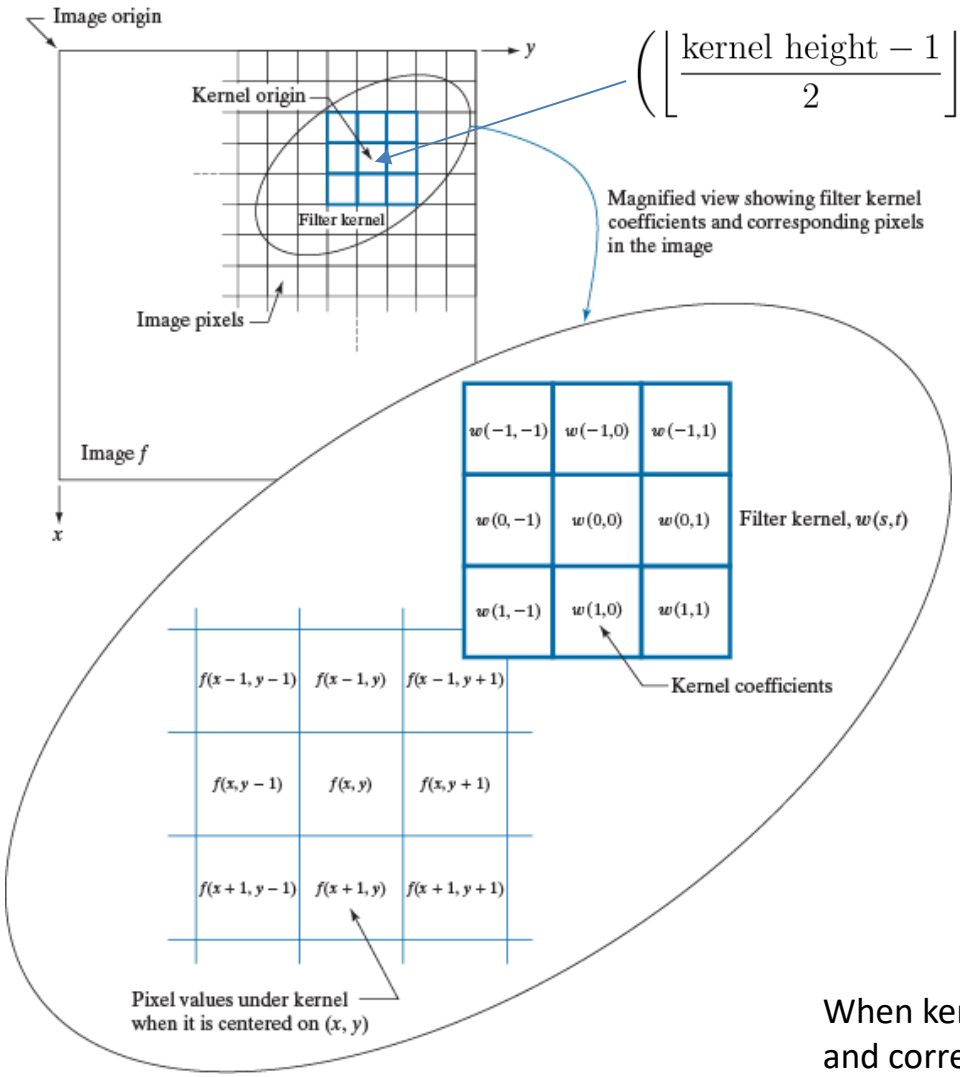
When kernel is symmetric, convolution and correlation give the same result

$w(s,t)$ is spatial filtering kernel (also called mask, template, and window)

$f(x,y)$ is input image

$g(x,y)$ is output image

# 2D spatial filtering




$$\left(\left\lfloor \frac{\text{kernel height} - 1}{2} \right\rfloor, \left\lfloor \frac{\text{kernel width} - 1}{2} \right\rfloor\right)$$

$m$ x $n$ kernel is $(2a + 1)$ x $(2b + 1)$, where $a$ and $b$ are positive integers, so $m$ and $n$ are odd positive integers

2D correlation

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

2D convolution

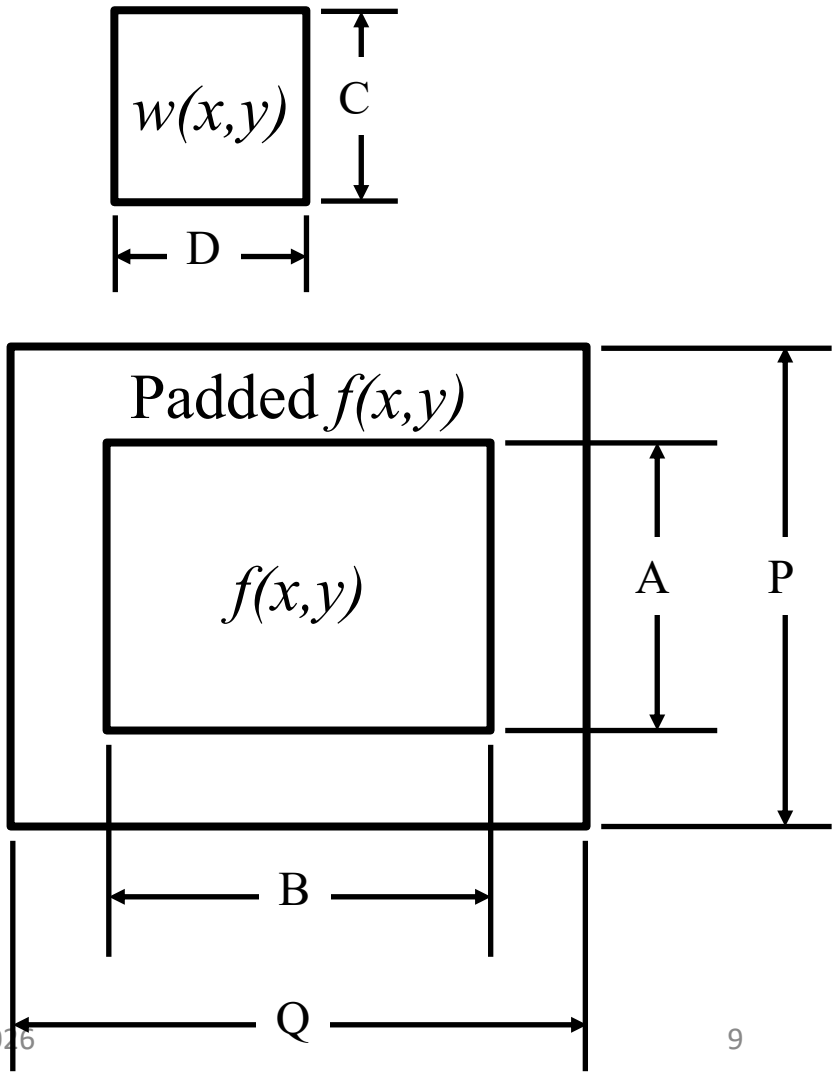
For convolution, kernel is rotated 180 degrees

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

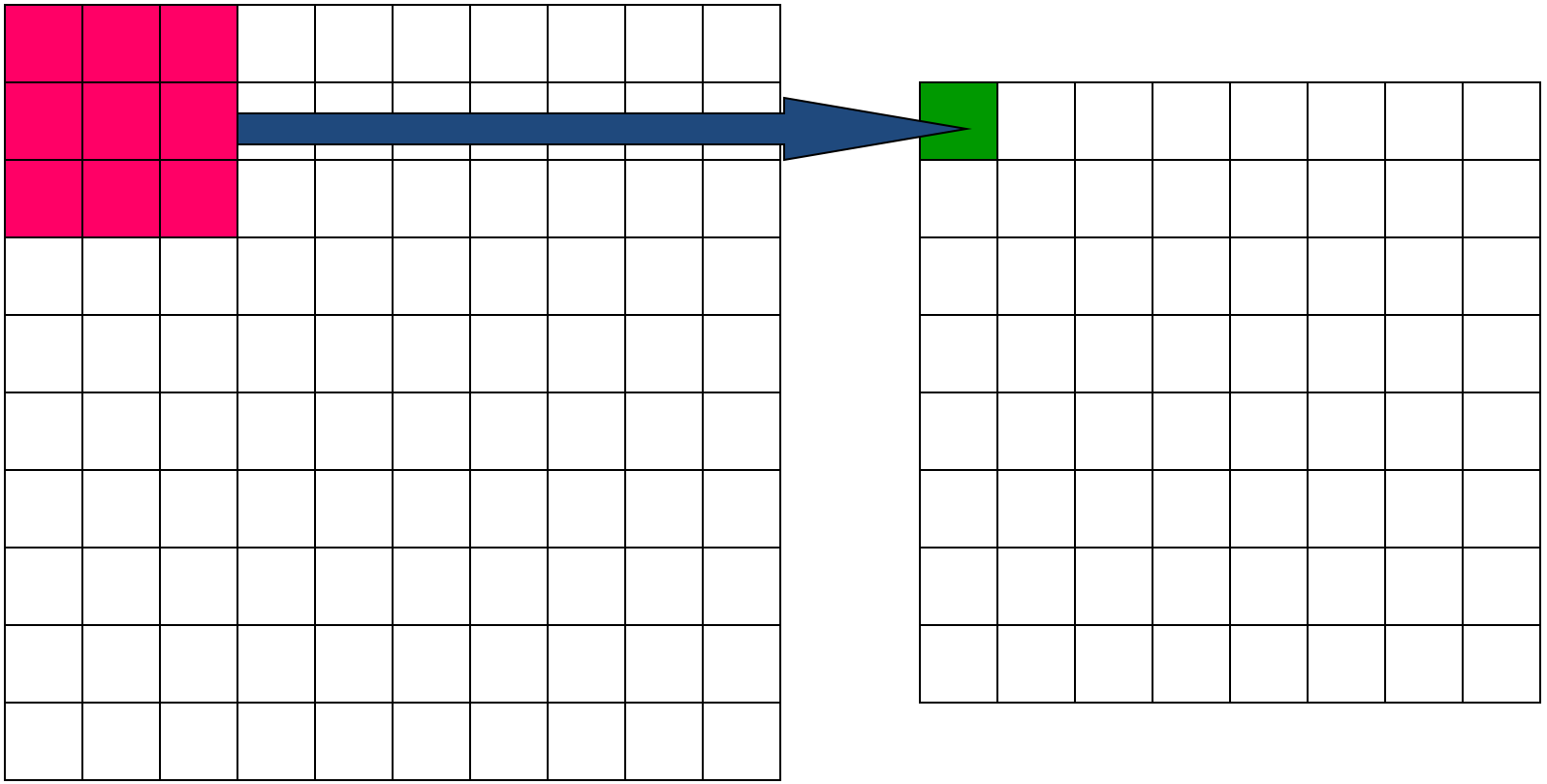
When kernel is symmetric, convolution and correlation give the same result

# 2D spatial filtering, shape

- Full
  - $w(x,y)$ and $f(x,y)$ have at least 1 pixel overlap
  - $P = A + 2(C - 1)$
  - $Q = B + 2(D - 1)$
  - Output $g(x,y)$
    - Width is $B + D - 1$, height is $A + C - 1$
- Same
  - $P = A + C - 1$
  - $Q = B + D - 1$
  - Output $g(x,y)$
    - Width is $B$, height is $A$
- Valid
  - $w(x,y)$ must be completely inside $f(x,y)$
  - $P = A$
  - $Q = B$
  - Output $g(x,y)$
    - Width is $B - D + 1$, height is $A - C + 1$



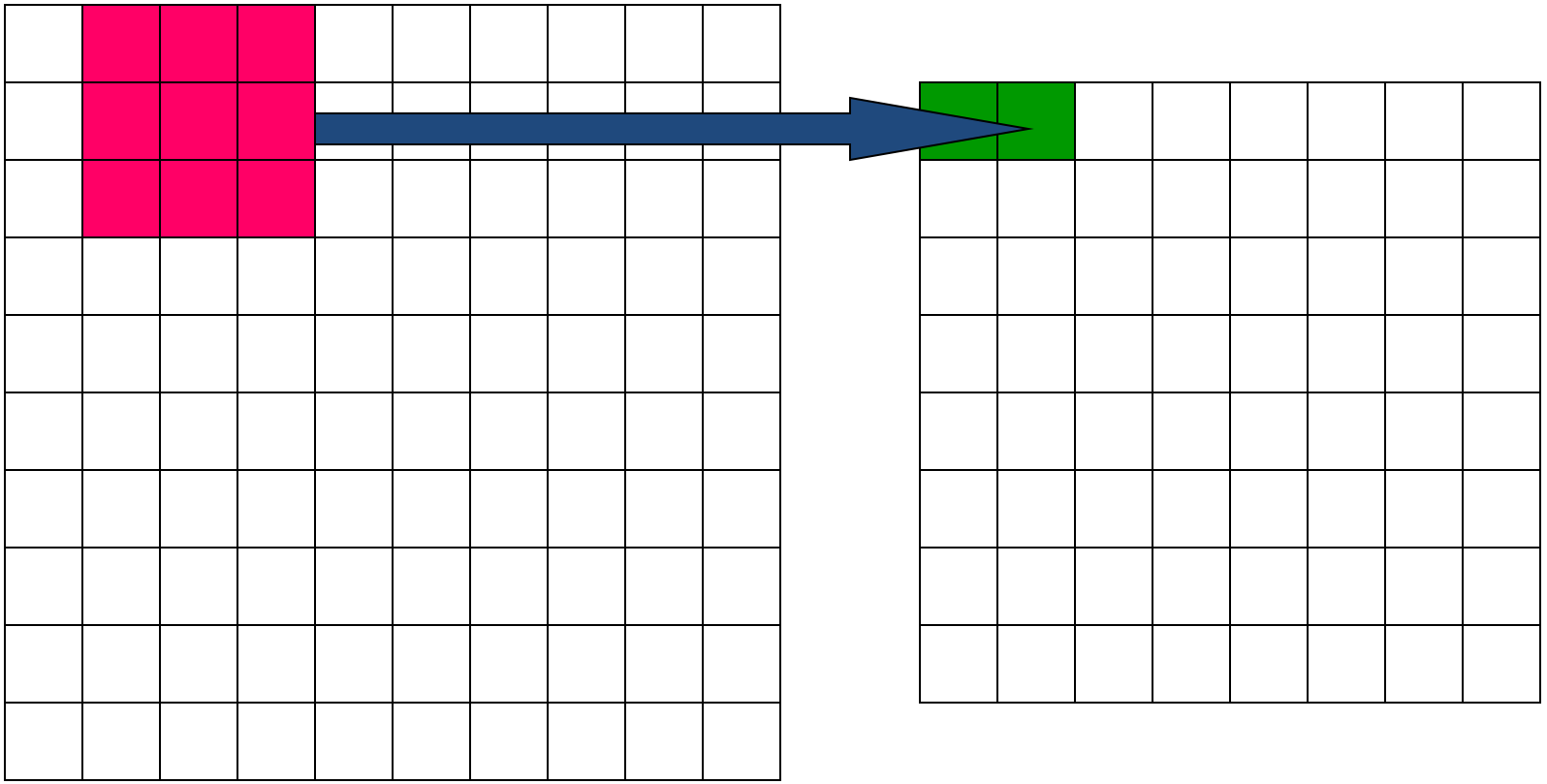
# 2D spatial filtering, valid shape



Input image

Output image

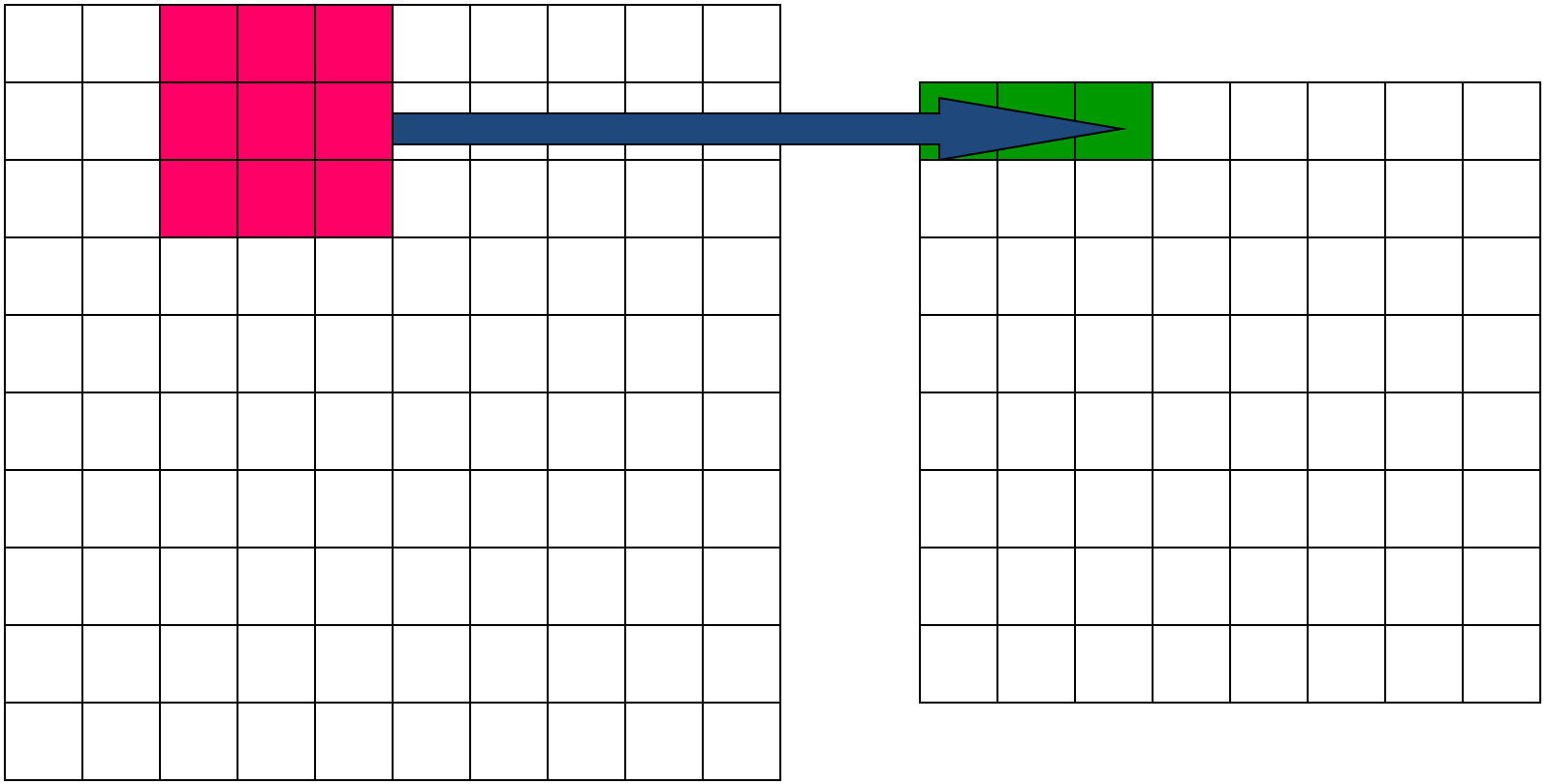
# 2D spatial filtering, valid shape



Input image

Output image

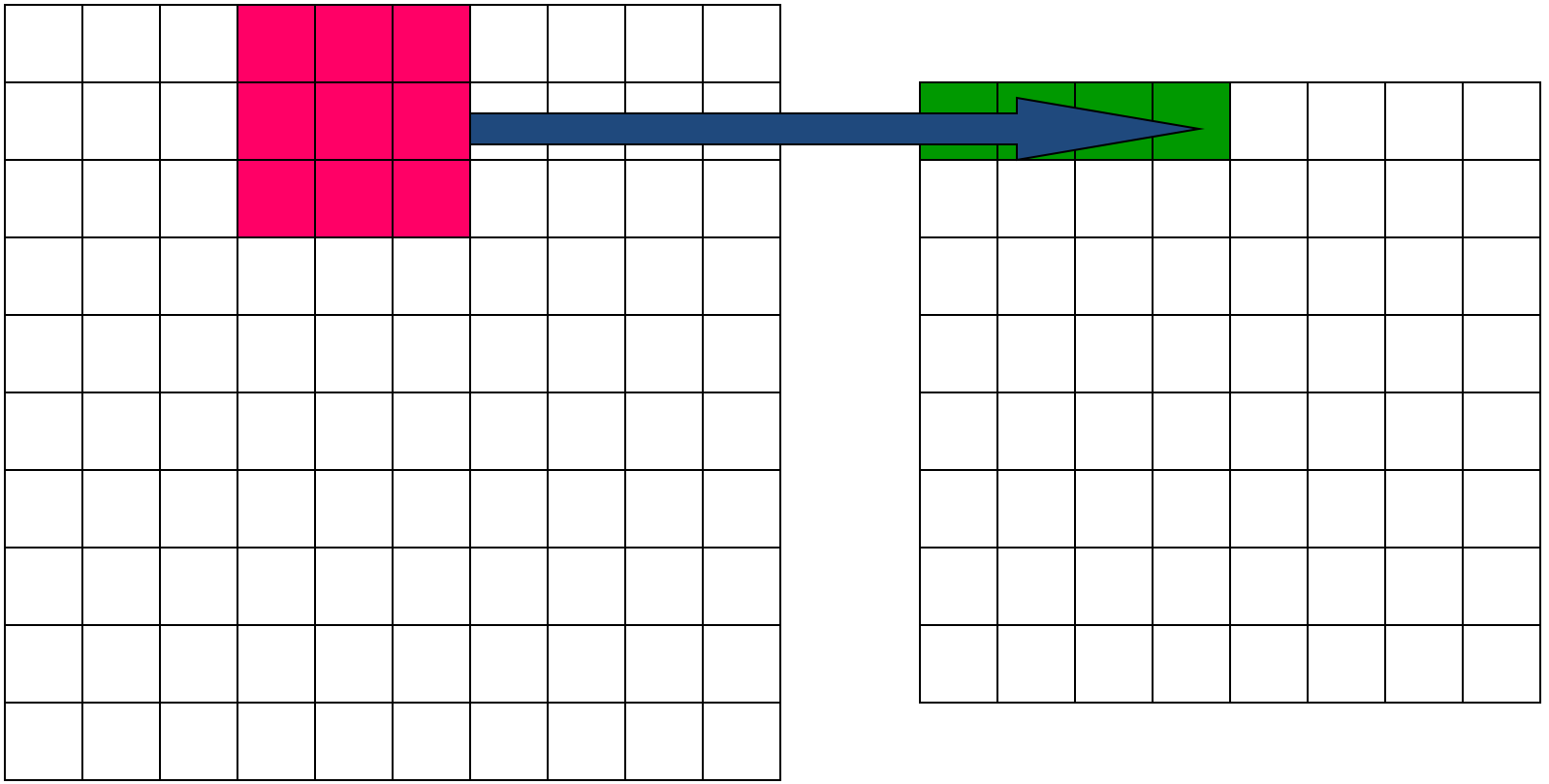
# 2D spatial filtering, valid shape



Input image

Output image

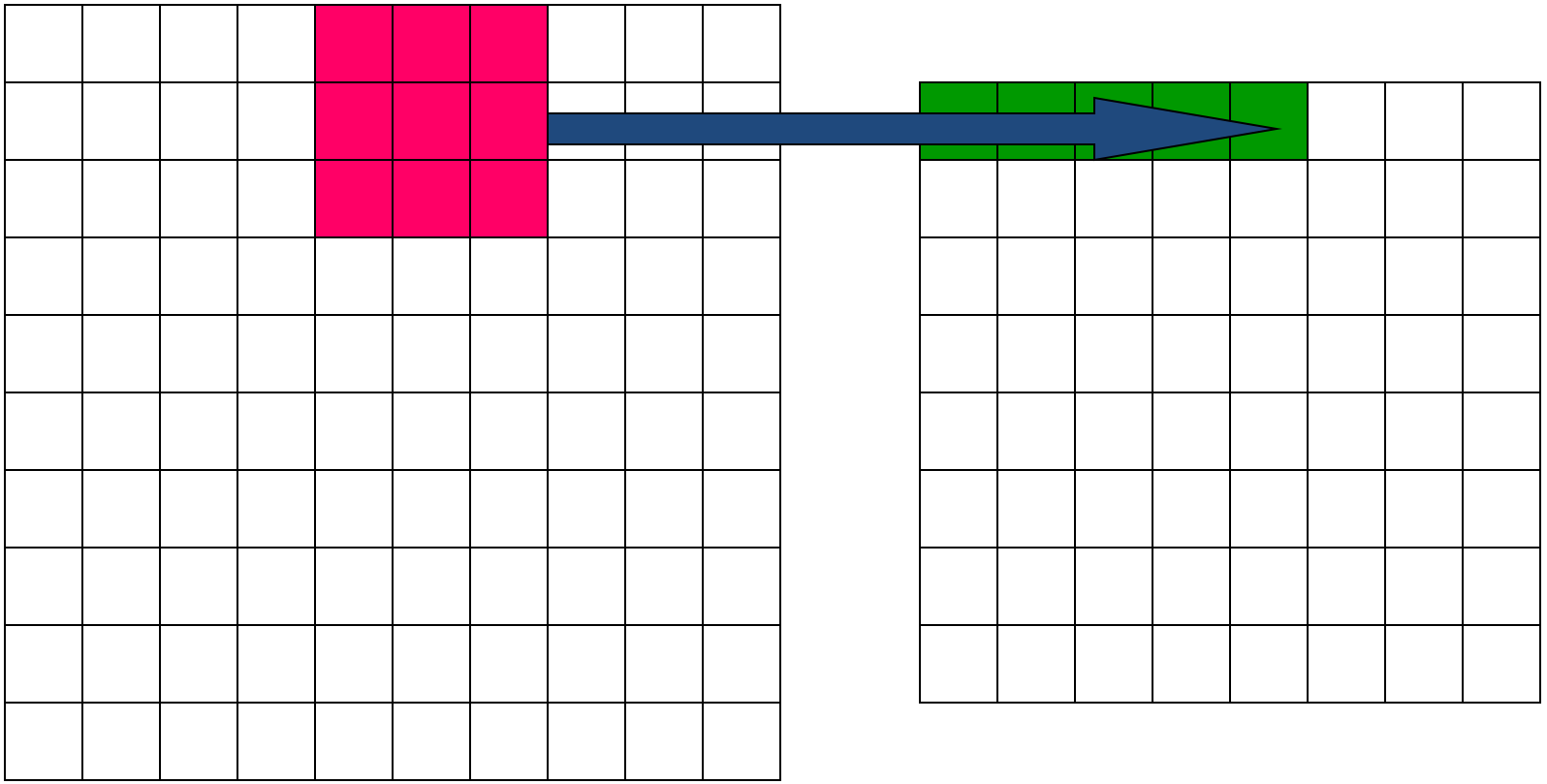
# 2D spatial filtering, valid shape



Input image

Output image

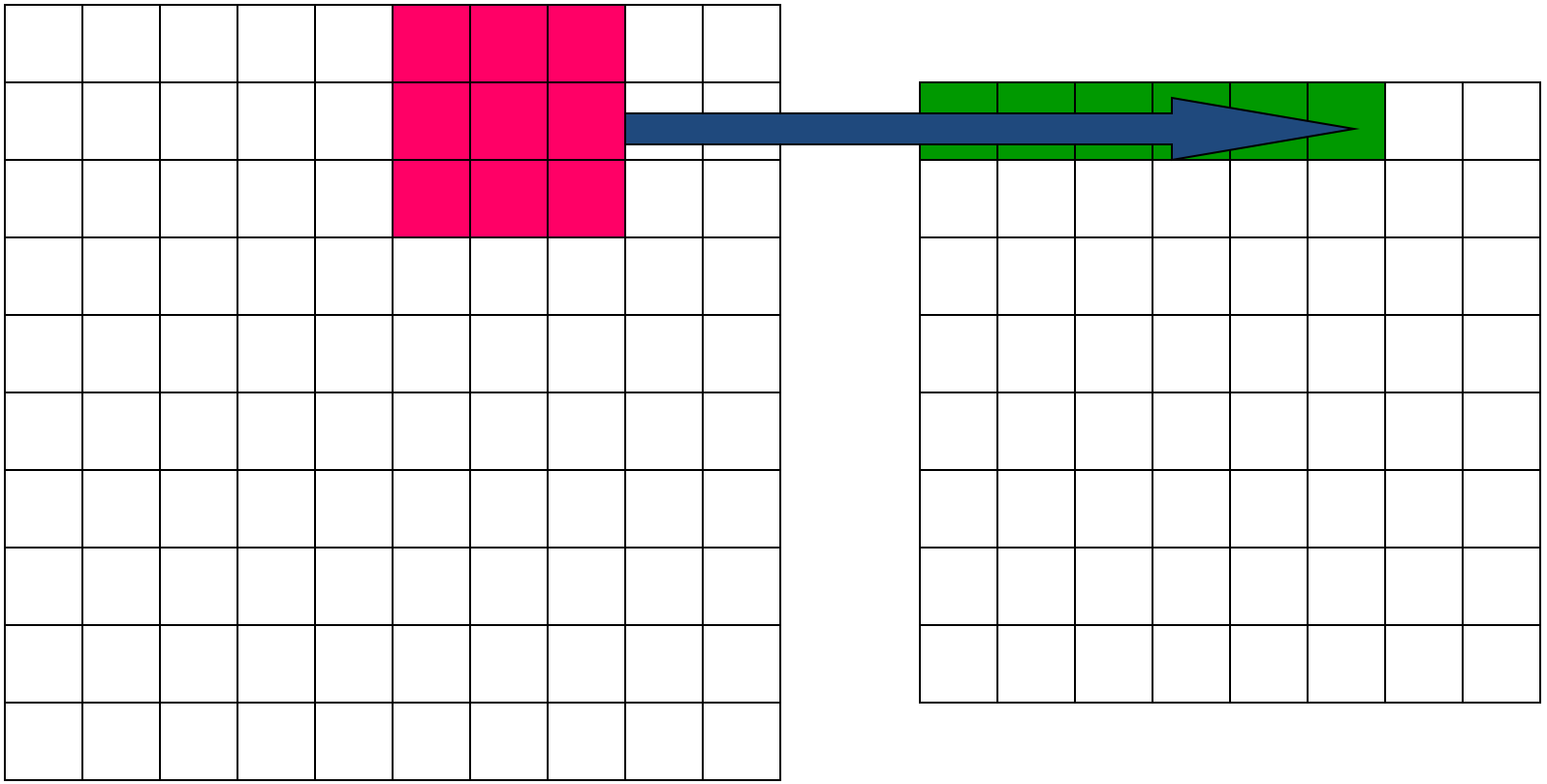
# 2D spatial filtering, valid shape



Input image

Output image

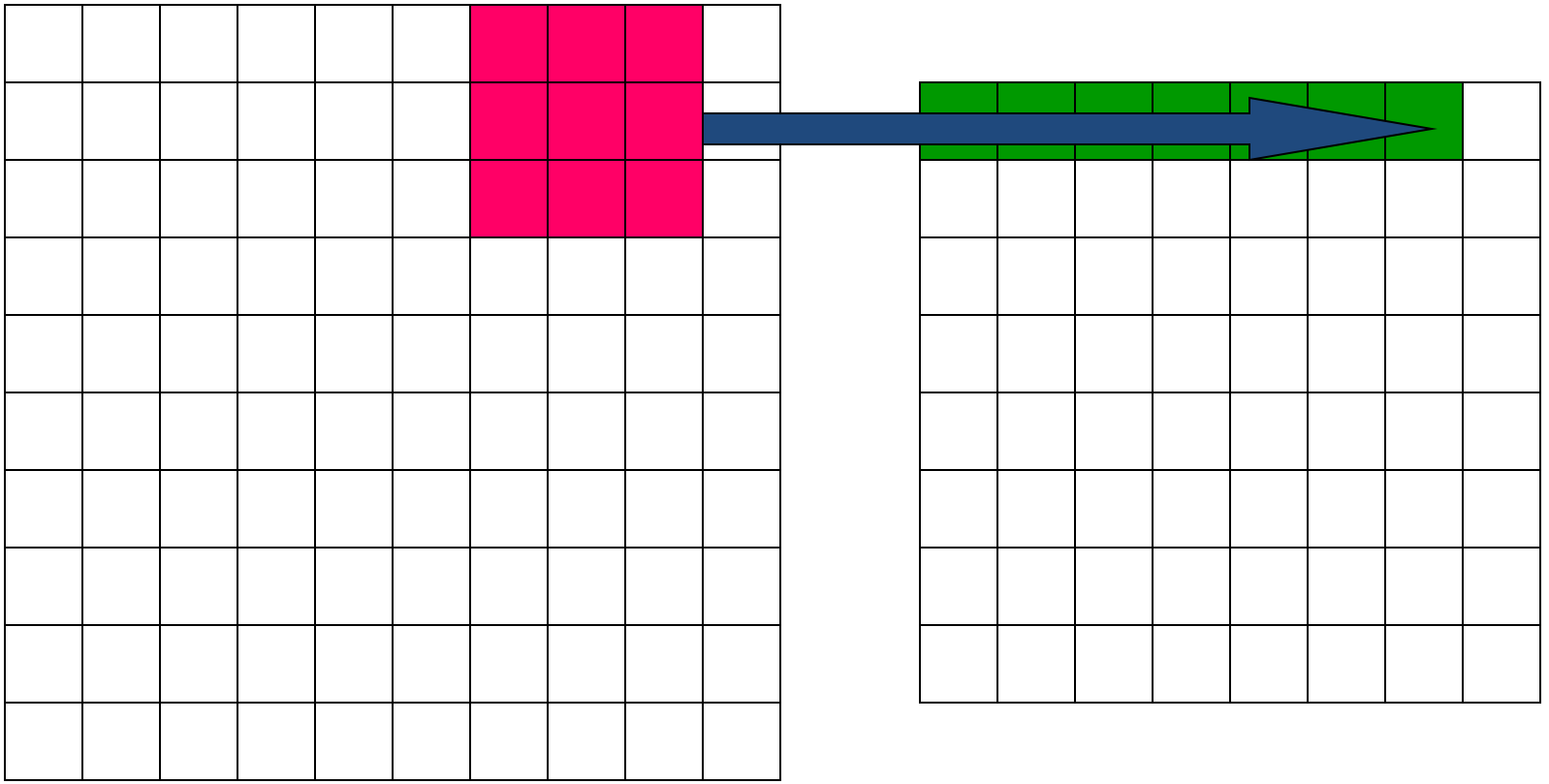
# 2D spatial filtering, valid shape



Input image

Output image

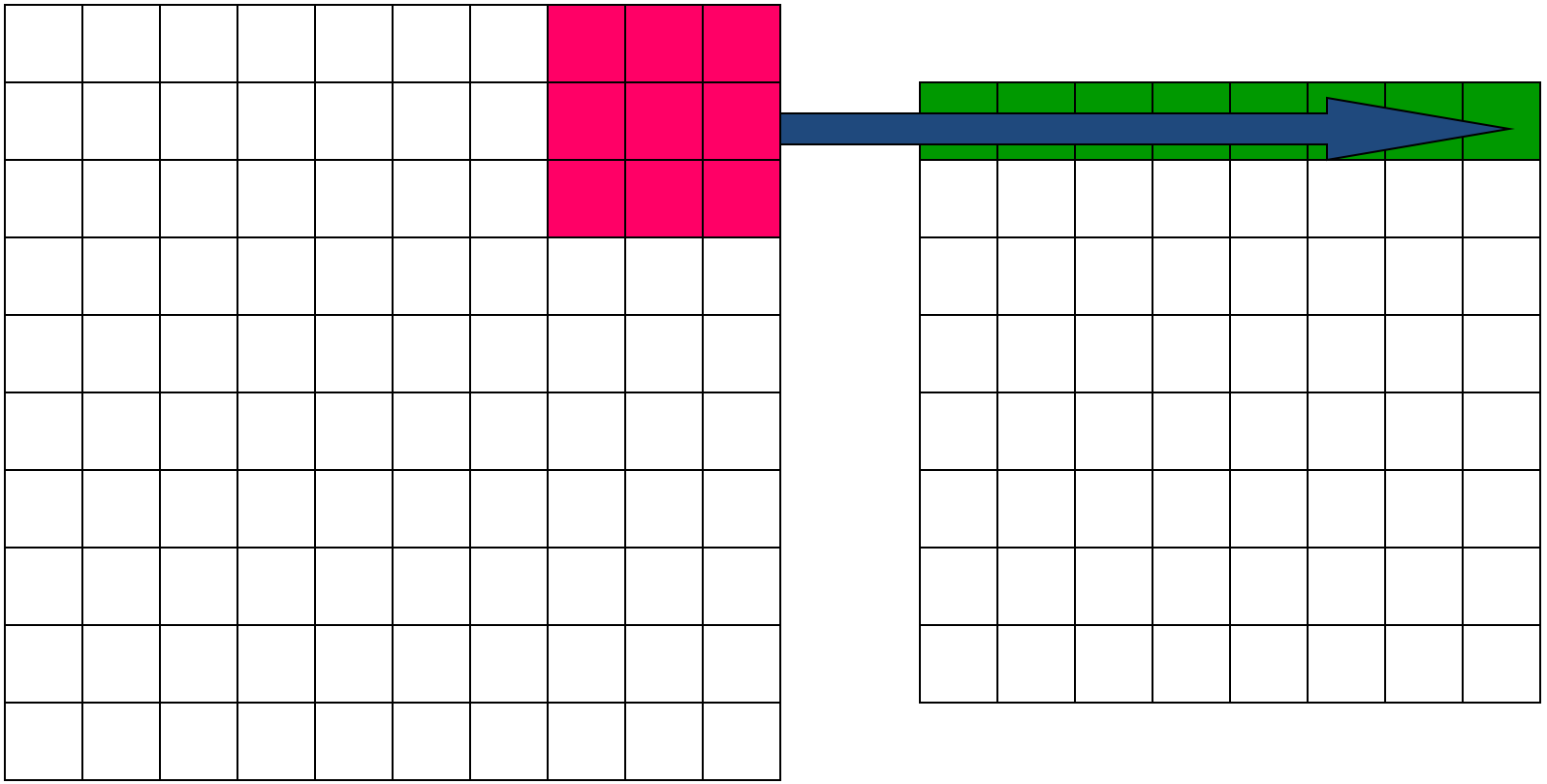
# 2D spatial filtering, valid shape



Input image

Output image

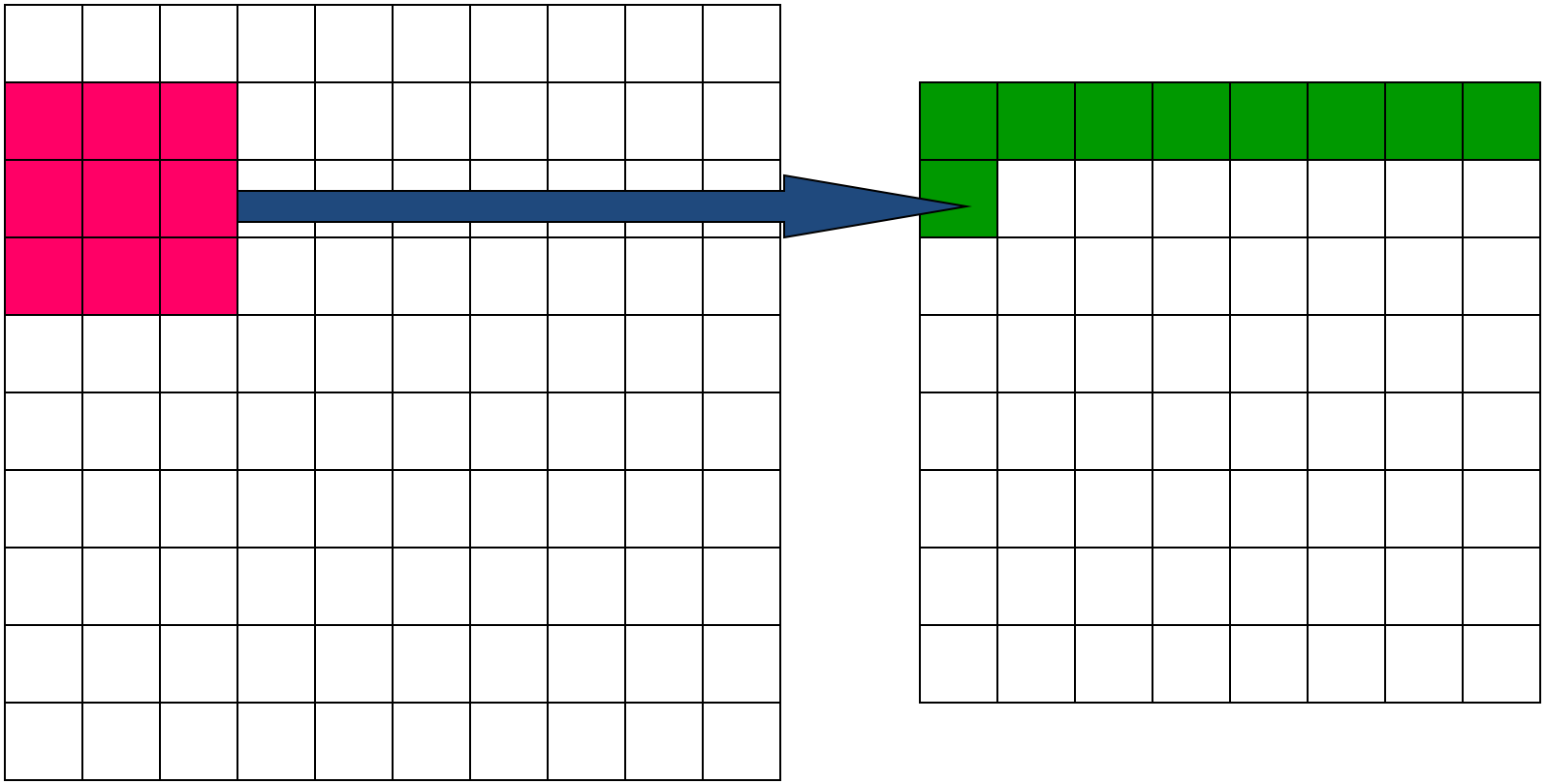
# 2D spatial filtering, valid shape



Input image

Output image

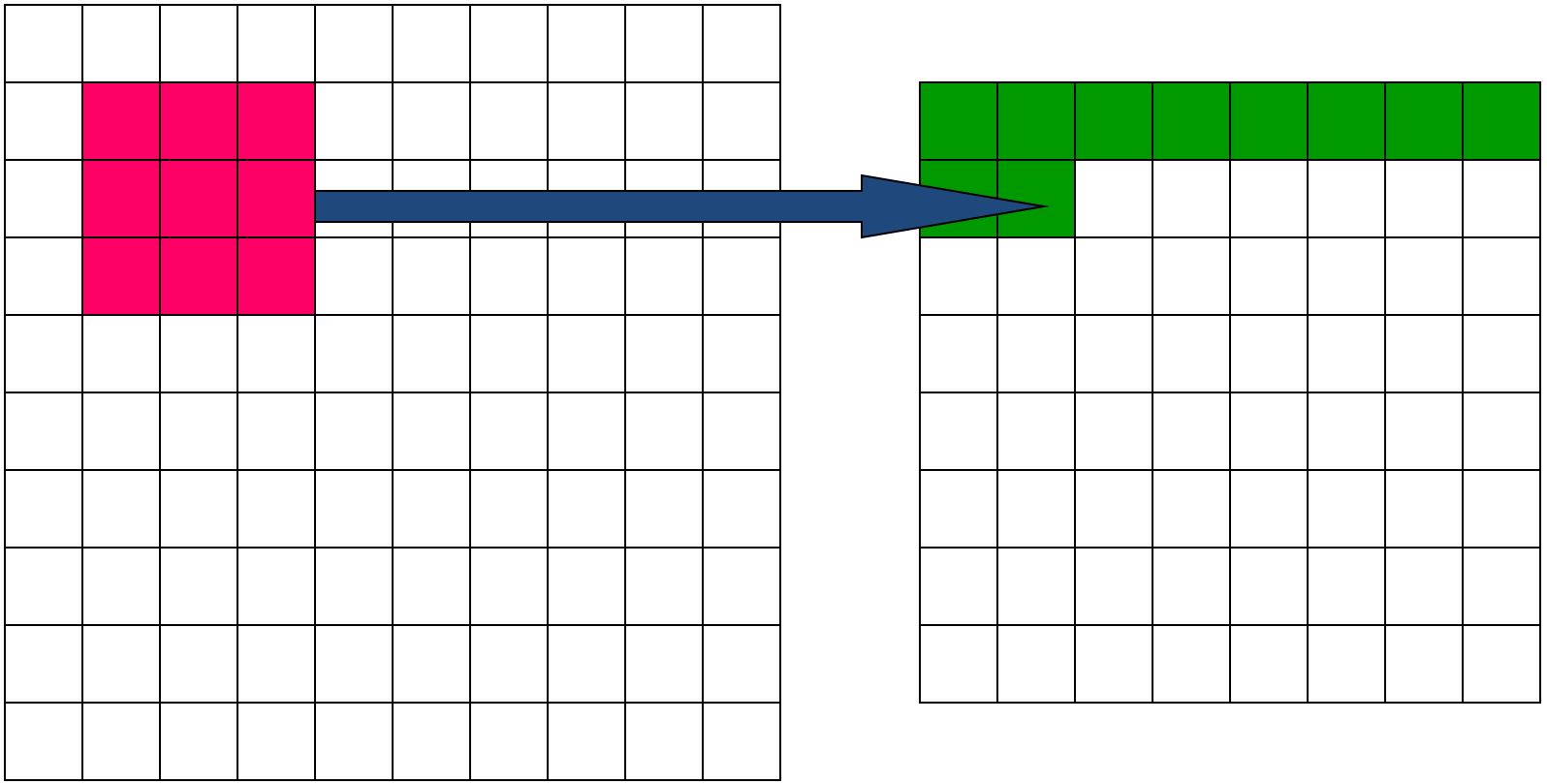
# 2D spatial filtering, valid shape



Input image

Output image

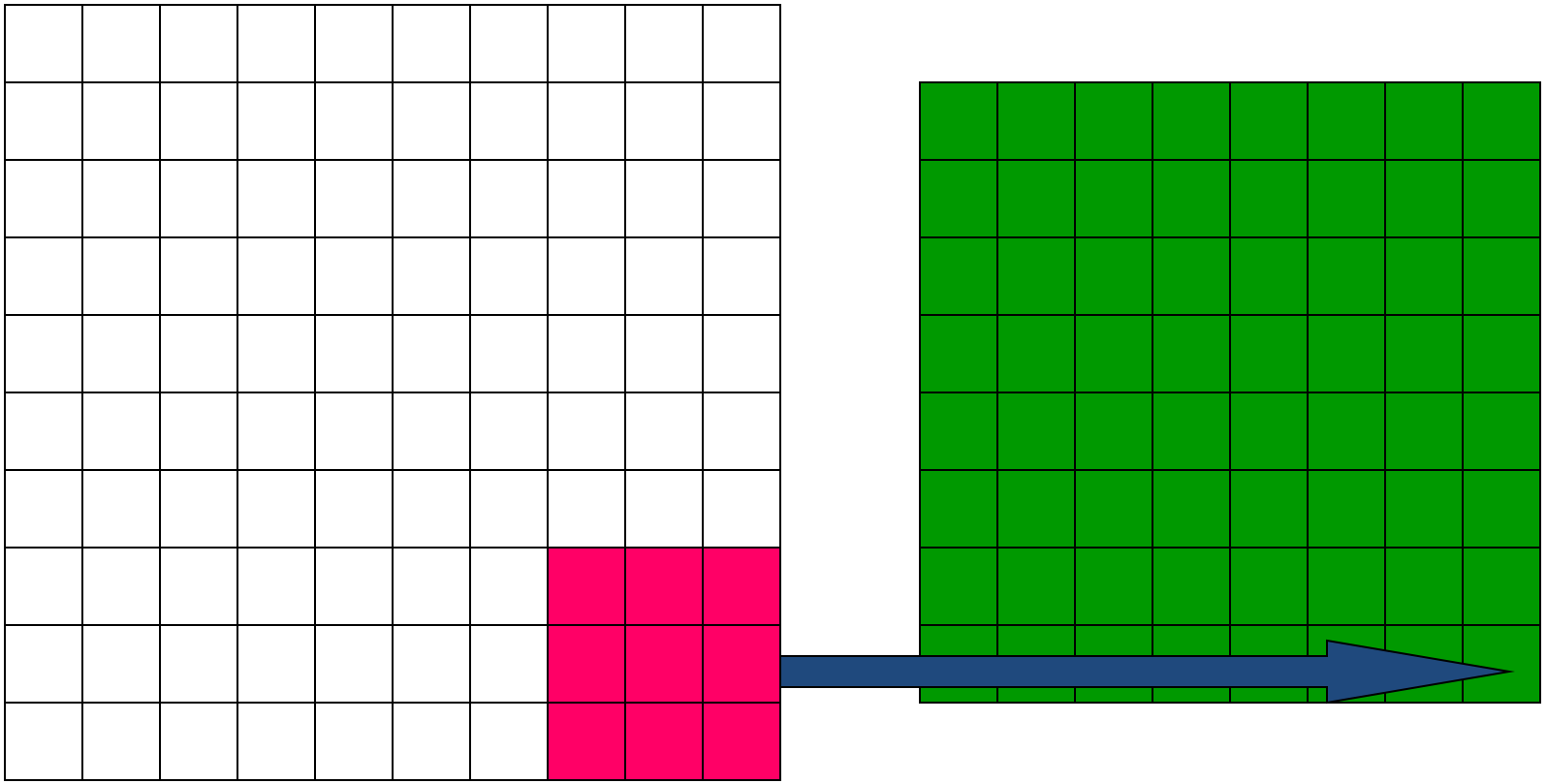
# 2D spatial filtering, valid shape



Input image

Output image

# 2D spatial filtering, valid shape



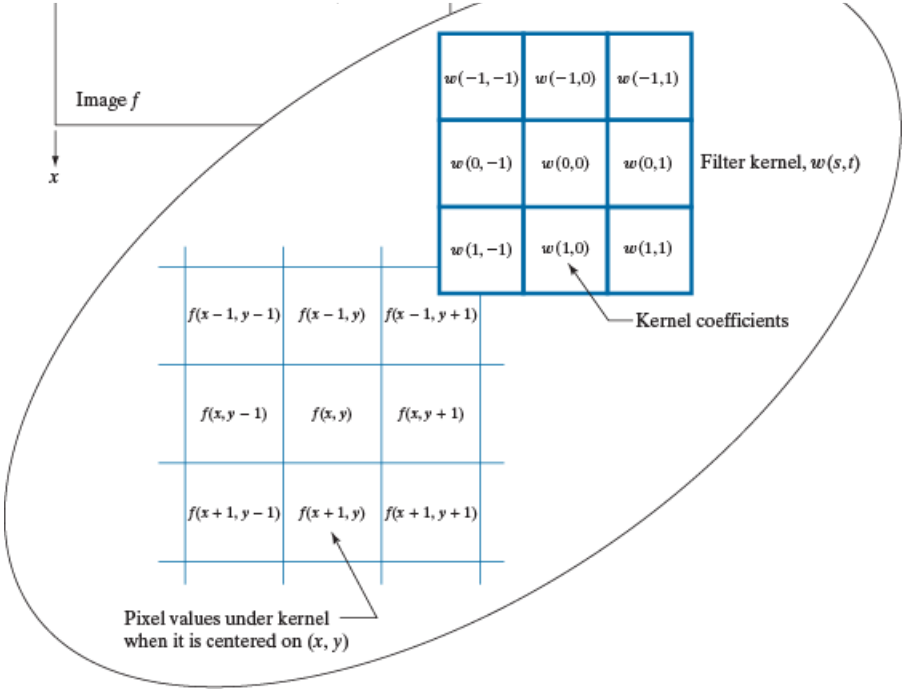
Input image

Output image

# 2D spatial filtering

- Example: 3 x 3 correlation

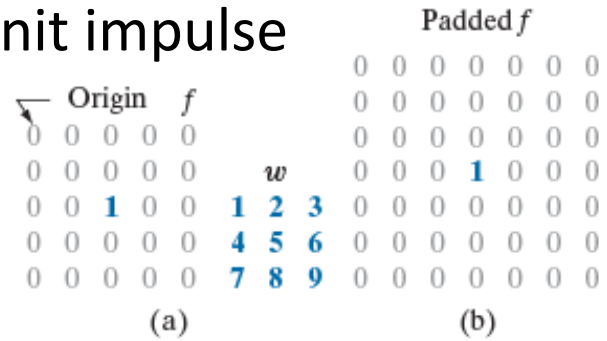
$$
\begin{aligned}
g(x,y) = \quad & w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + w(-1,1)f(x-1,y+1) \\
+ \quad & w(0,-1)f(x,y-1) + w(0,0)f(x,y) + w(0,1)f(x,y+1) \\
+ \quad & w(1,-1)f(x+1,y-1) + w(1,0)f(x+1,y) + w(1,1)f(x+1,y+1)
\end{aligned}
$$




Largest response when kernel coefficients are equal to corresponding image pixel values

# 2D correlation and convolution

Unit impulse




For convolution, kernel is rotated 180 degrees

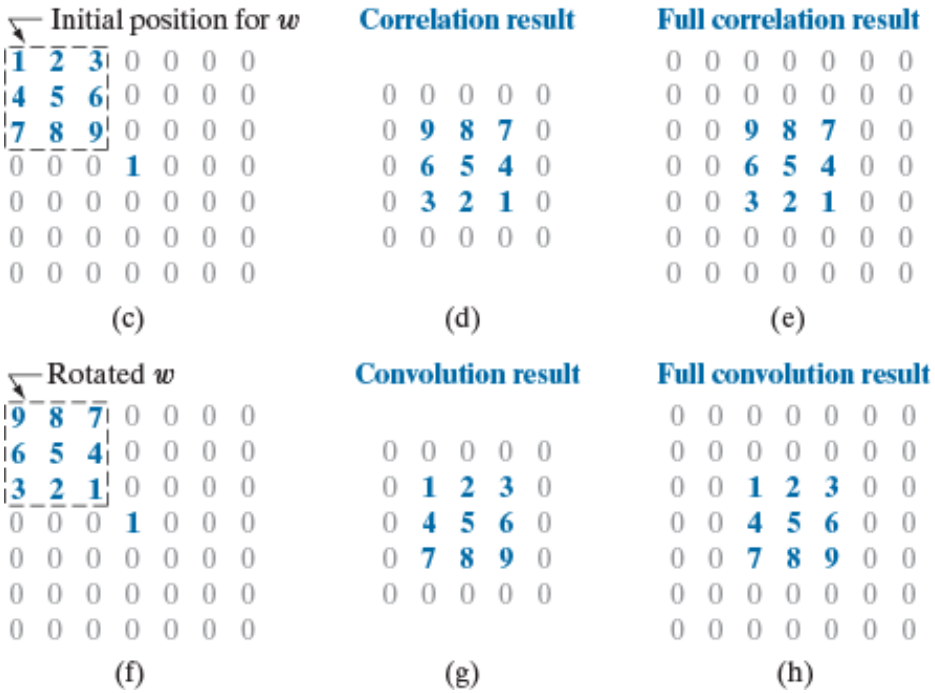



**FIGURE 3.36** Correlation (middle row) and convolution (last row) of a 2-D kernel with an image consisting of a discrete unit impulse. The 0's are shown in gray to simplify visual analysis. Note that correlation and convolution are functions of $x$ and $y$. As these variable change, they *displace* one function with respect to the other. See the discussion of Eqs. (3-45) and (3-46) regarding full correlation and convolution.

# Correlation and convolution

This is equivalent to applying one "combined" filter, but shape full and zero padding must be used

**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 | $f \star (g + h) = (f \star g) + (f \star h)$ | $f \star (g + h) = (f \star g) + (f \star h)$ |

- Convolution is commutative and associative, correlation is not

# Smoothing kernels

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

Average
(box kernel)

$$\frac{1}{4.8976} \times \begin{bmatrix} 0.3679 & 0.6065 & 0.3679 \\ 0.6065 & 1.0000 & 0.6065 \\ 0.3679 & 0.6065 & 0.3679 \end{bmatrix}$$

Weighted average
(Gaussian kernel)

Common that sum of filter coefficients is 1, which prevents (full) bias
Bias: sum of input pixel values not equal to sum of output pixel values

# Smoothing with box kernel

Input image





3x3

11x11



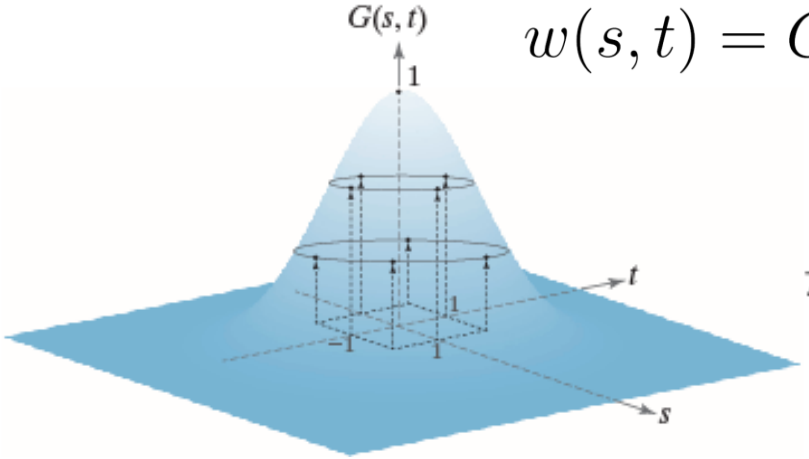


21x21

# Smoothing with Gaussian kernel

a b

**FIGURE 3.41**
(a) Sampling a Gaussian function to obtain a discrete Gaussian kernel. The values shown are for $K = 1$ and $\sigma = 1$. (b) Resulting $3 \times 3$ kernel [this is the same as Fig. 3.37(b)].




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

$$\frac{1}{4.8976} \times \begin{bmatrix} 0.3679 & 0.6065 & 0.3679 \\ 0.6065 & 1.0000 & 0.6065 \\ 0.3679 & 0.6065 & 0.3679 \end{bmatrix}$$

| Standard deviation σ | Percent of total volume under surface |
|---|---|
| 1 | 39.35 |
| 2 | 86.47 |
| 3 | 98.89 |

Volume under surface greater than 3σ is negligible

# Smoothing with Gaussian kernel

$$\text{minimum size is } \begin{cases} \lceil 6\sigma \rceil & \text{if } \lceil 6\sigma \rceil \text{ is odd} \\ \lceil 6\sigma \rceil + 1 & \text{otherwise} \end{cases}$$



$\sigma = 7$
43x43



$\sigma = 7$
85x85

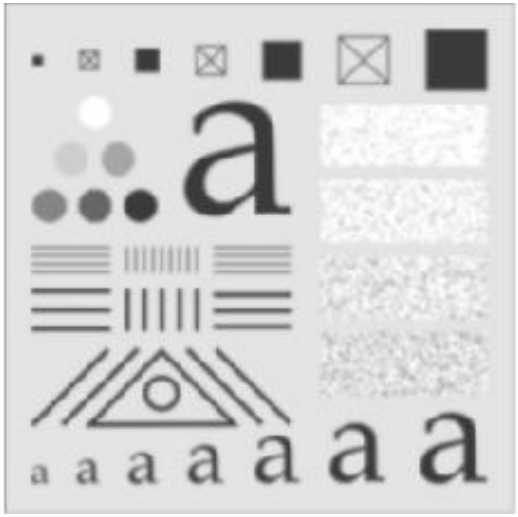


Difference

# Smoothing with Gaussian kernel



Input image



$\sigma = 3.5$
21x21



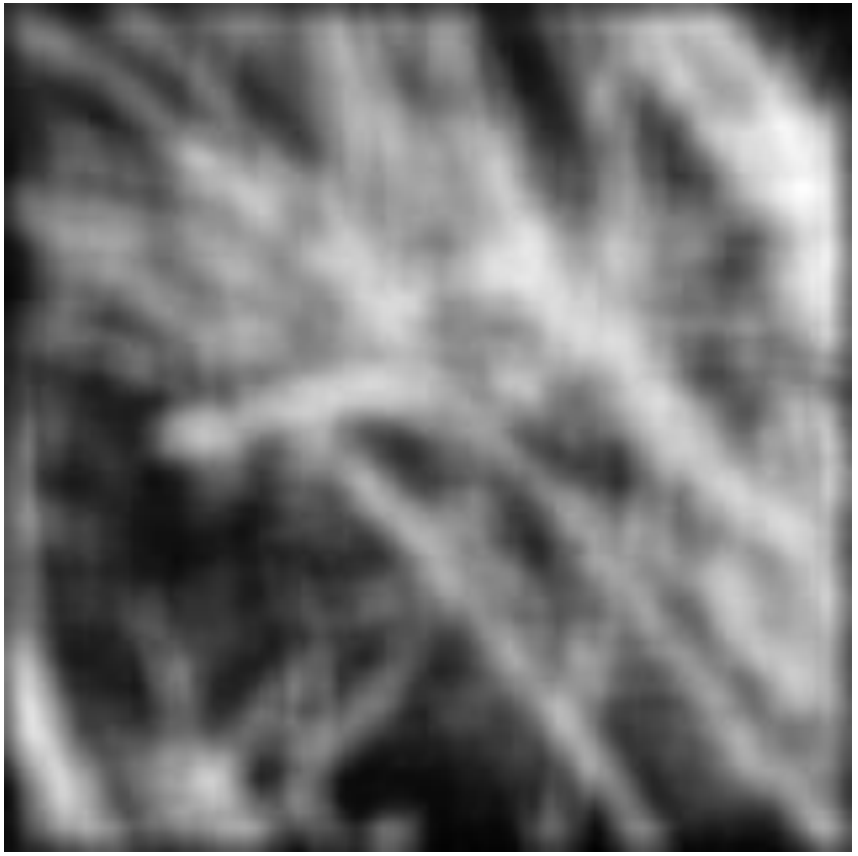
$\sigma = 7$
43x43

# Smoothing with box kernel

Input image



Box kernel

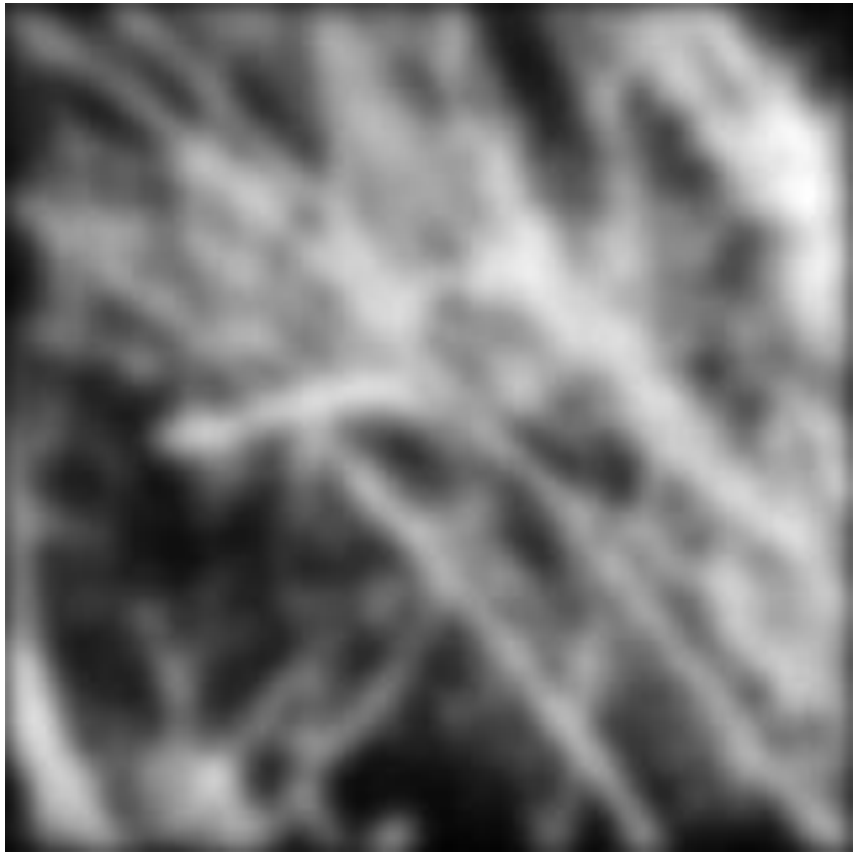


# Smoothing with Gaussian kernel

Input image

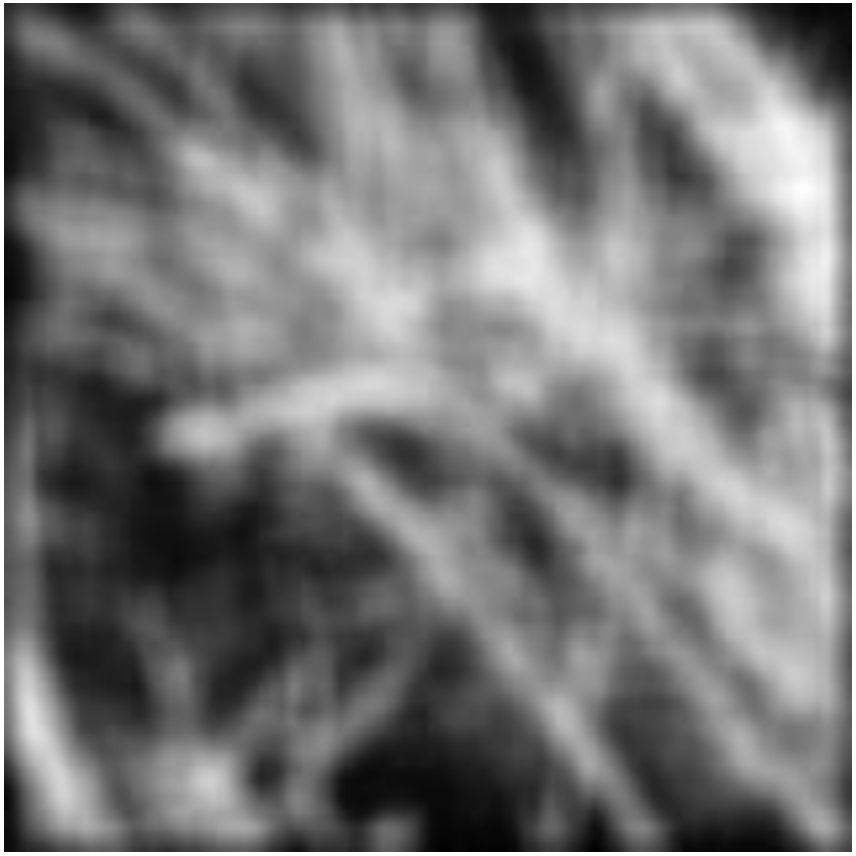


Gaussian kernel

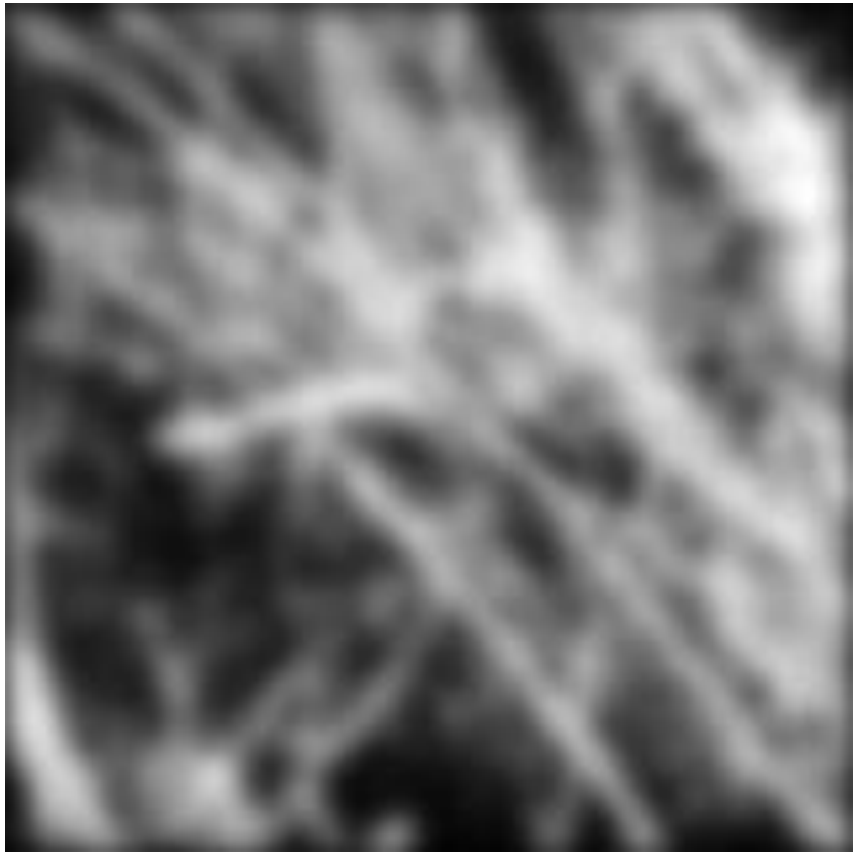


# Smoothing

Box kernel



Gaussian kernel



# Border padding

| v | v | v | v | v | v | v | v | v |
|---|---|---|---|---|---|---|---|---|
| v | v | v | v | v | v | v | v | v |
| v | v | 1 | 2 | 3 | 4 | 5 | v | v |
| v | v | 6 | 7 | 8 | 9 | 10 | v | v |
| v | v | 11 | 12 | 13 | 14 | 15 | v | v |
| v | v | 16 | 17 | 18 | 19 | 20 | v | v |
| v | v | v | v | v | v | v | v | v |

Constant padding

Zero padding
when v = 0

| 1 | 1 | 1 | 2 | 3 | 4 | 5 | 5 | 5 |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 2 | 3 | 4 | 5 | 5 | 5 |
| 1 | 1 | 1 | 2 | 3 | 4 | 5 | 5 | 5 |
| 6 | 6 | 6 | 7 | 8 | 9 | 10 | 10 | 10 |
| 11 | 11 | 11 | 12 | 13 | 14 | 15 | 15 | 15 |
| 16 | 16 | 16 | 17 | 18 | 19 | 20 | 20 | 20 |
| 16 | 16 | 16 | 17 | 18 | 19 | 20 | 20 | 20 |

Replicate padding

| 13 | 12 | 11 | 12 | 13 | 14 | 15 | 14 | 13 |
|---|---|---|---|---|---|---|---|---|
| 8 | 7 | 6 | 7 | 8 | 9 | 10 | 9 | 8 |
| 3 | 2 | 1 | 2 | 3 | 4 | 5 | 4 | 3 |
| 8 | 7 | 6 | 7 | 8 | 9 | 10 | 9 | 8 |
| 13 | 12 | 11 | 12 | 13 | 14 | 15 | 14 | 13 |
| 18 | 17 | 16 | 17 | 18 | 19 | 20 | 19 | 18 |
| 13 | 12 | 11 | 12 | 13 | 14 | 15 | 14 | 13 |

Mirror padding

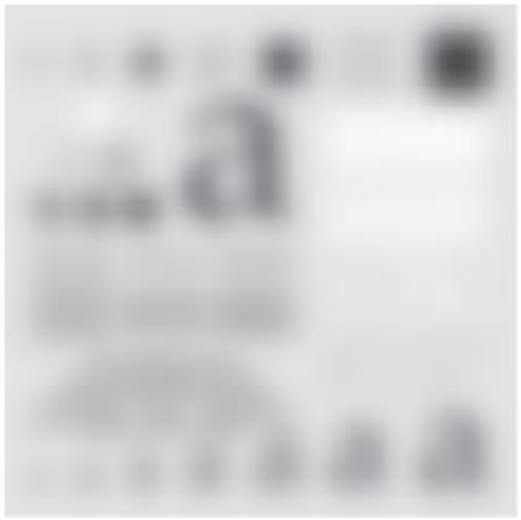
# Border padding



Zero padding

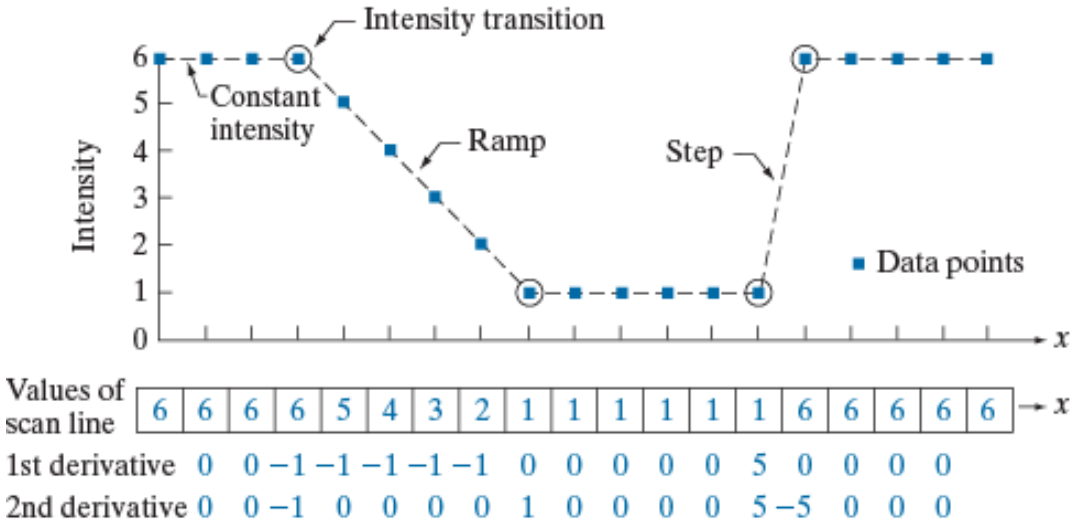


Mirror padding



Replicate padding

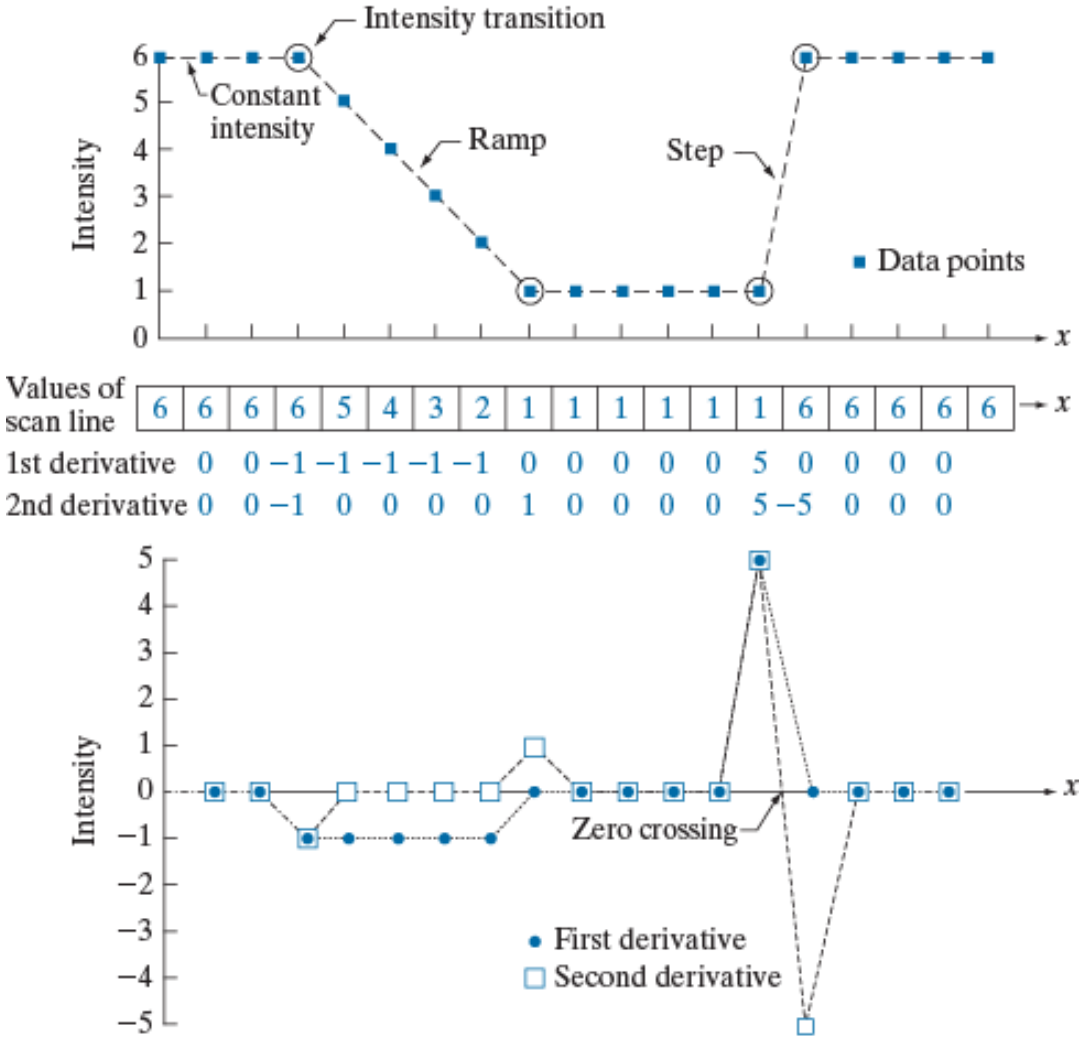
# 1D derivatives




$$\frac{\mathrm{d}f}{\mathrm{d}x} = f(x+1) - f(x) \qquad \text{First derivative (forward difference)}$$

$$\frac{\mathrm{d}^2 f}{\mathrm{d}x^2} = f(x+1) + f(x-1) - 2f(x) \qquad \text{Second derivative}$$

# 1D derivatives




a
b
c

**FIGURE 3.50**
(a) A section of a horizontal scan line from an image, showing ramp and step edges, as well as constant segments.
(b)Values of the scan line and its derivatives.
(c) Plot of the derivatives, showing a zero crossing. In (a) and (c) points were joined by dashed lines as a visual aid.

# 2D first derivatives

- Gradient vector $\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$

  Points in direction of greatest rate of change at $(x,y)$

  – Roberts operator

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

Filter kernels

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Even dimensions are not preferred

# 2D first derivatives

- Gradient vector $\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$ Points in direction of greatest rate of change at $(x,y)$
  - Sobel operator

$$\begin{aligned} \frac{\partial f}{\partial x} = & \ (f(x+1,y-1) + 2f(x+1,y) + f(x+1,y+1)) \\ & -(f(x-1,y-1) + 2f(x-1,y) + f(x-1,y+1)) \text{ and} \\ \frac{\partial f}{\partial y} = & \ (f(x-1,y+1) + 2f(x,y+1) + f(x+1,y+1)) \\ & -(f(x-1,y-1) + 2f(x,y-1) + f(x+1,y-1)) \end{aligned}$$

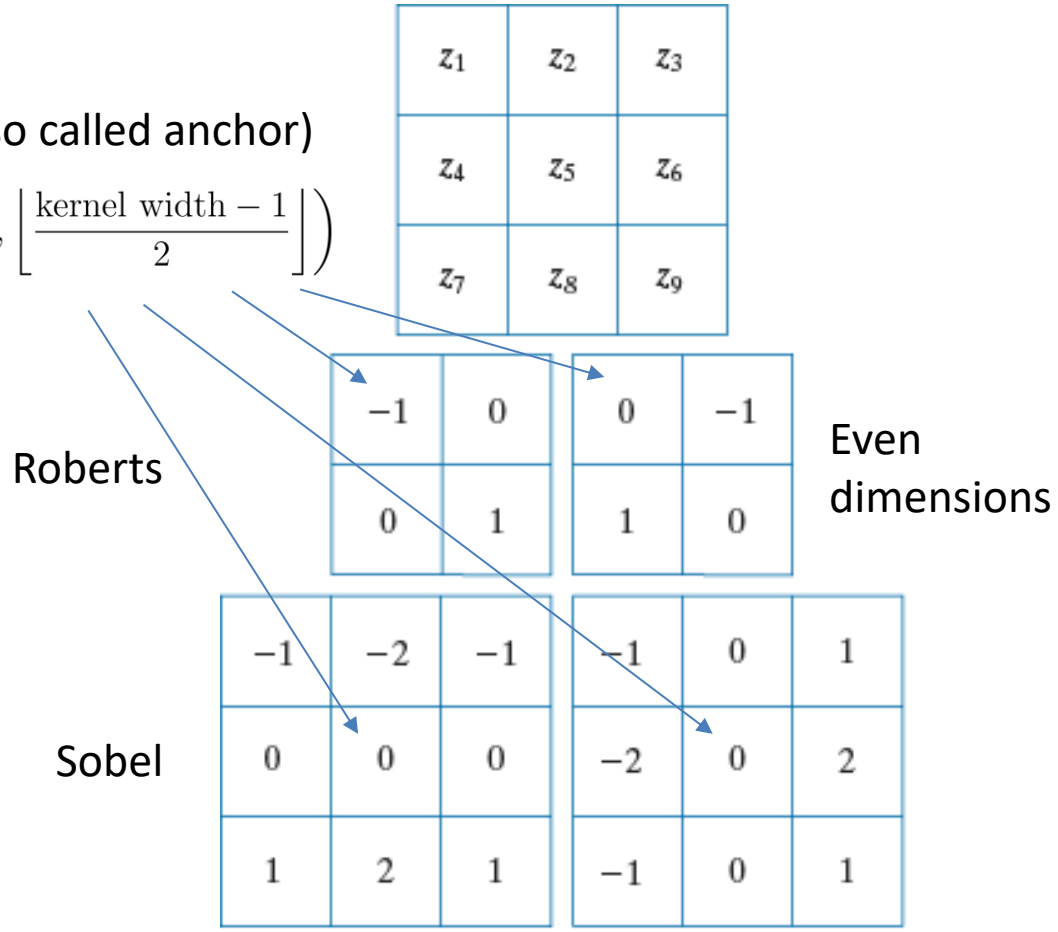
Filter kernels

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

# 2D first derivatives

Kernel origin (also called anchor)

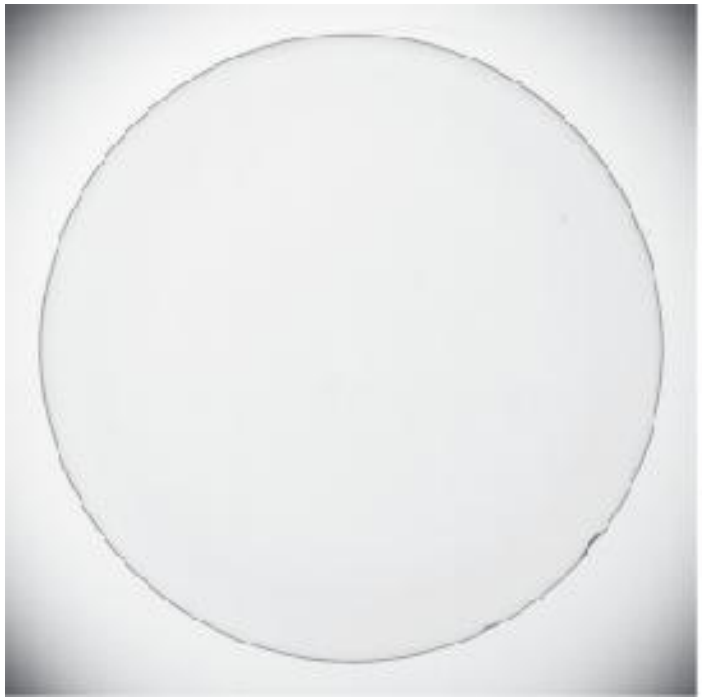
$$\left(\left\lfloor \frac{\text{kernel height} - 1}{2} \right\rfloor, \left\lfloor \frac{\text{kernel width} - 1}{2} \right\rfloor\right)$$

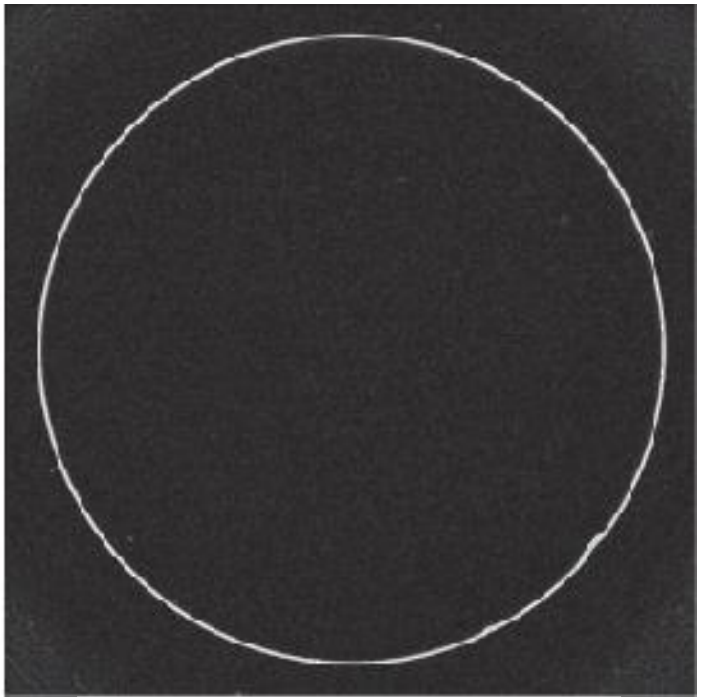



a
b c
d e

**FIGURE 3.56**
(a) A $3 \times 3$ region of an image, where the $z$s are intensity values. (b)–(c) Roberts cross-gradient operators. (d)–(e) Sobel operators. All the kernel coefficients sum to zero, as expected of a derivative operator.

# Magnitude of gradient vector





a b

**FIGURE 3.57**
(a) Image of a contact lens (note defects on the boundary at 4 and 5 o'clock).
(b) Sobel gradient.
(Original image courtesy of Perceptics Corporation.)

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

# 2D second derivatives

- Laplacian

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

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

$$\begin{aligned} \nabla^2 f = & \quad\quad\quad\quad f(x-1, y) \\ & + \quad f(x, y-1) \quad + \quad -4f(x, y) \quad + \quad f(x, y+1) \\ & \quad\quad\quad\quad + \quad f(x+1, y) \end{aligned}$$

Filter kernel

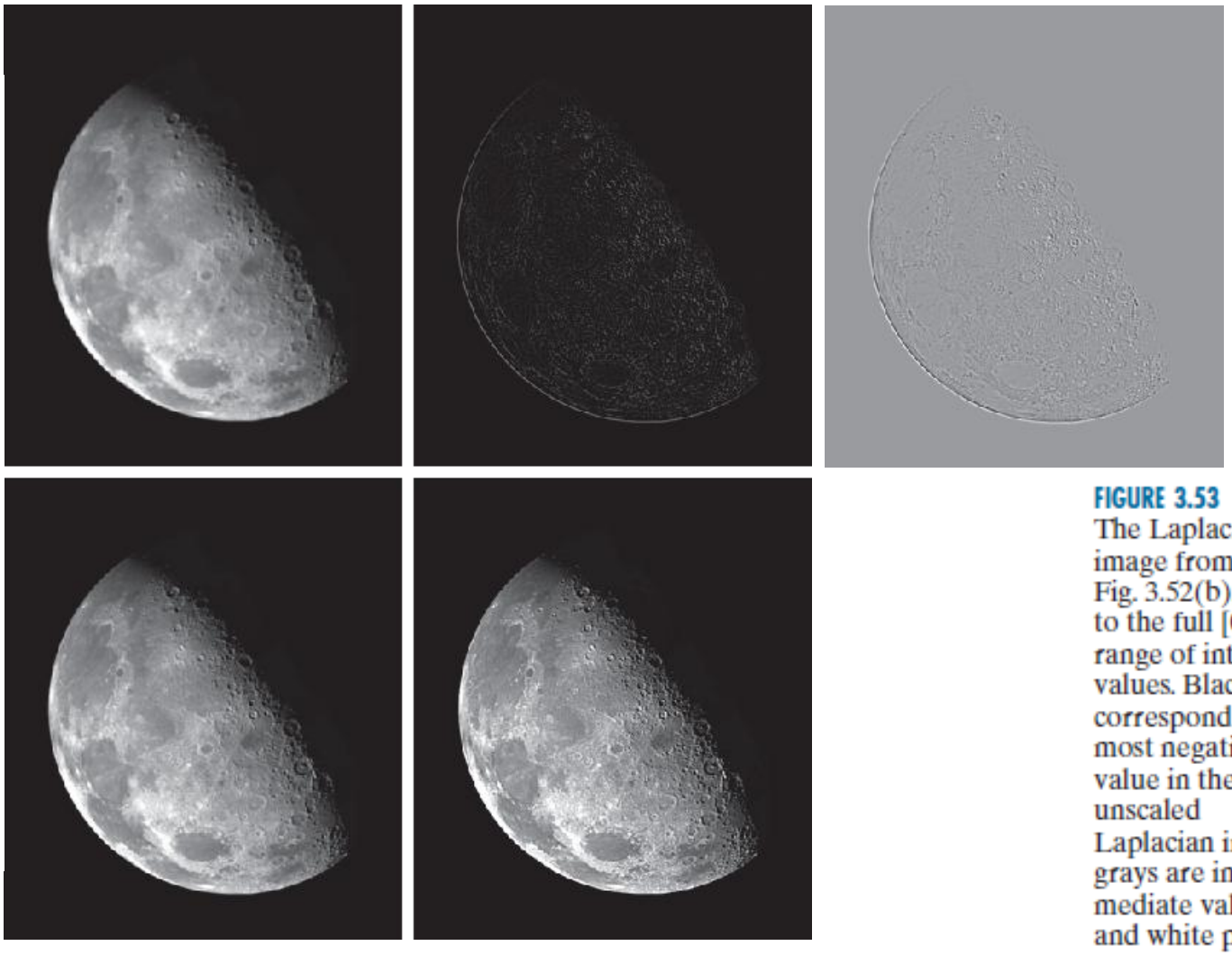
$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Note: output may be negative

  - Sharpening

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

# Sharpening

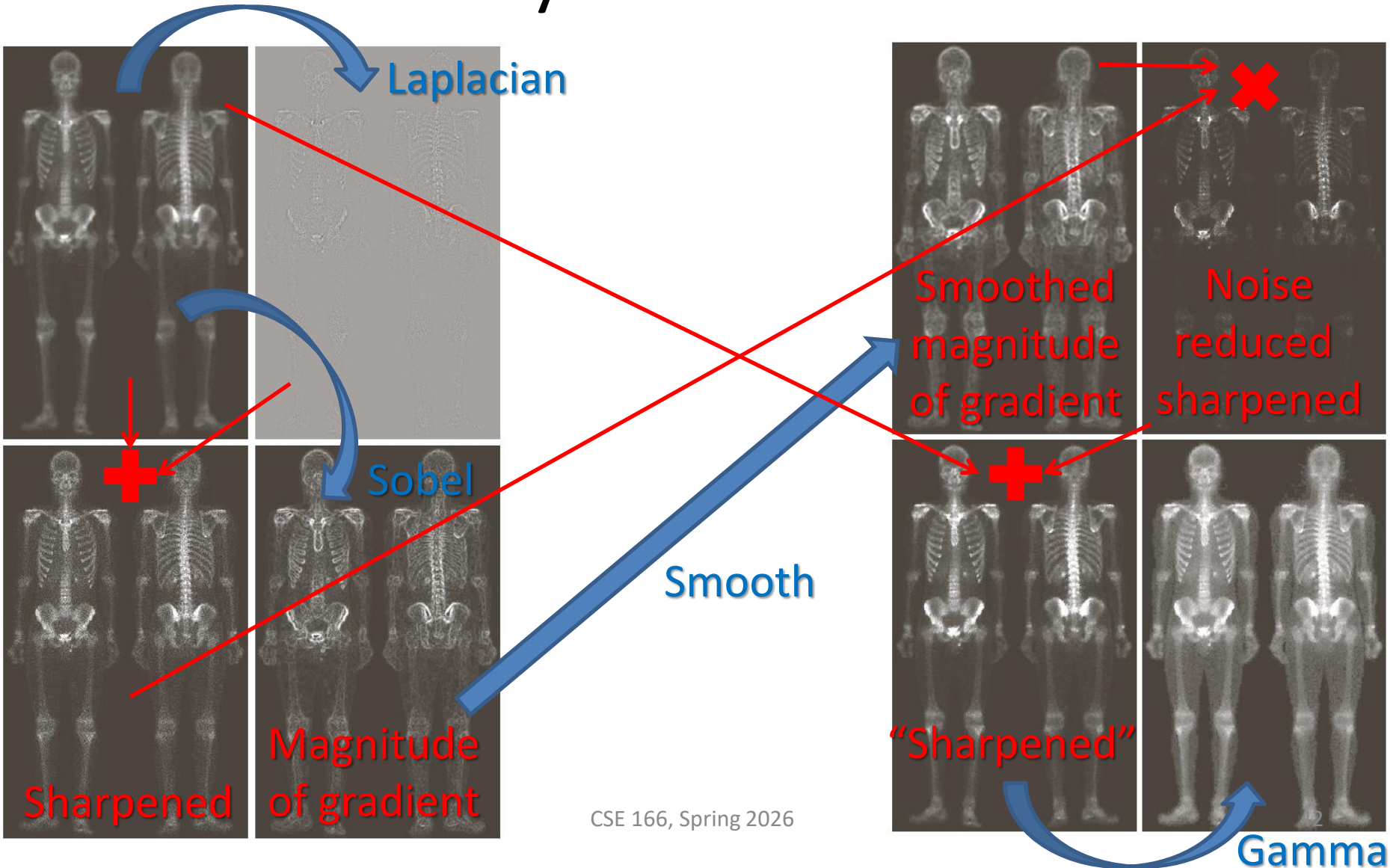


a b
c d

**FIGURE 3.52**
(a) Blurred image of the North Pole of the moon.
(b) Laplacian image obtained using the kernel in Fig. 3.51(a).
(c) Image sharpened using Eq. (3-63) with $c = -1$.
(d) Image sharpened using the same procedure, but with the kernel in Fig. 3.51(b).
(Original image courtesy of NASA.)

**FIGURE 3.53**
The Laplacian image from Fig. 3.52(b), 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.

# Combining spatial filtering and intensity transformations



# Next Lecture

- Continuous Fourier transform
- Reading
  - Chapter 4: Filtering in the Frequency Domain
    - Sections 4.1 and 4.2