Università degli Studi di Padova

# Spatial filtering – linear filters

Stefano Ghidoni

Dipartimento di Ingegneria dell'Informazione

Intelligent Autonomous Systems Lab

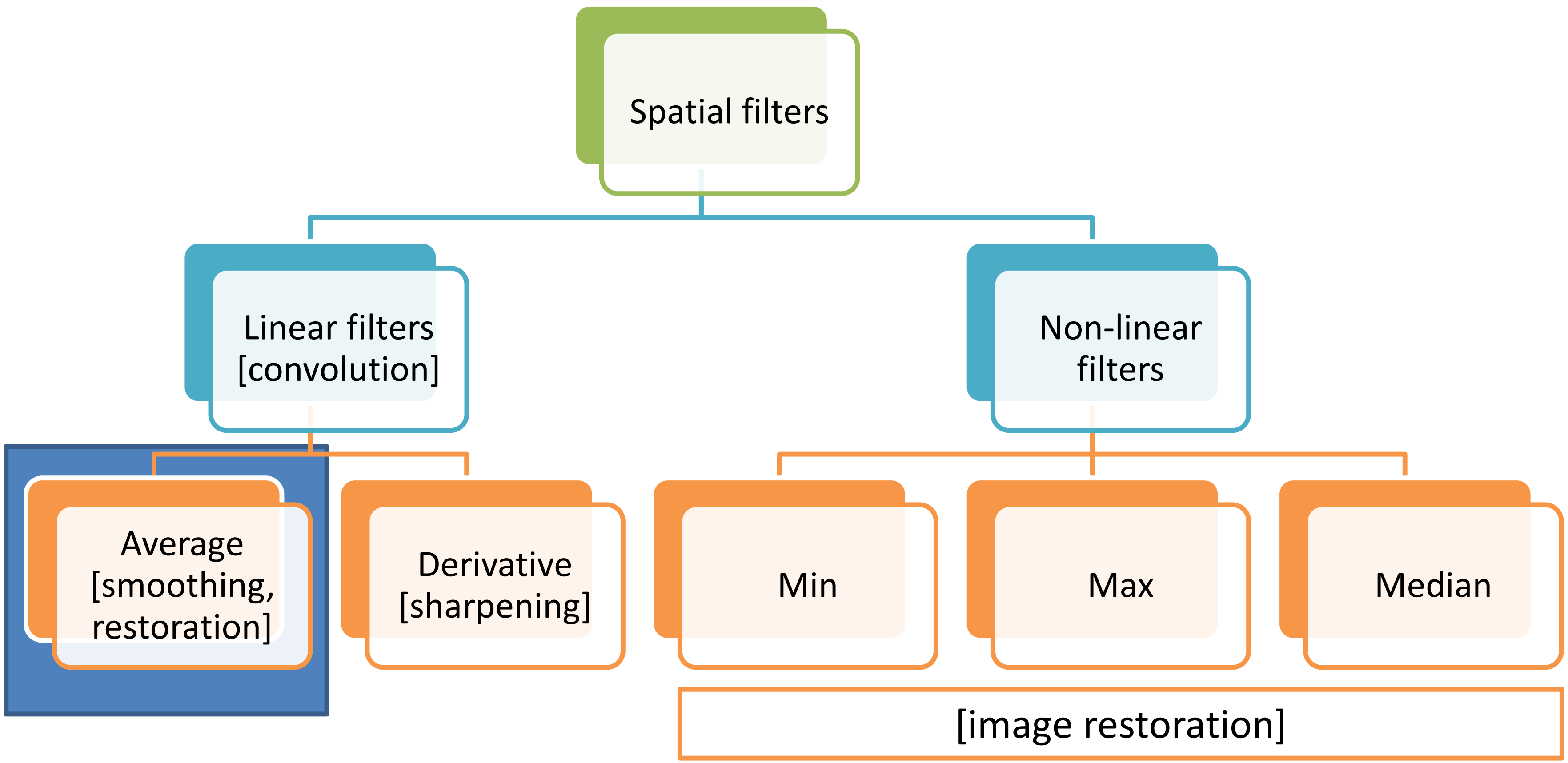
# Agenda



- Averaging filter
- Derivative filters
- Image sharpening

# Spatial filters overview





# Averaging filter



- Consider a kernel 3x3
- How would you evaluate average over such neighborhood?
- What is the related mask?

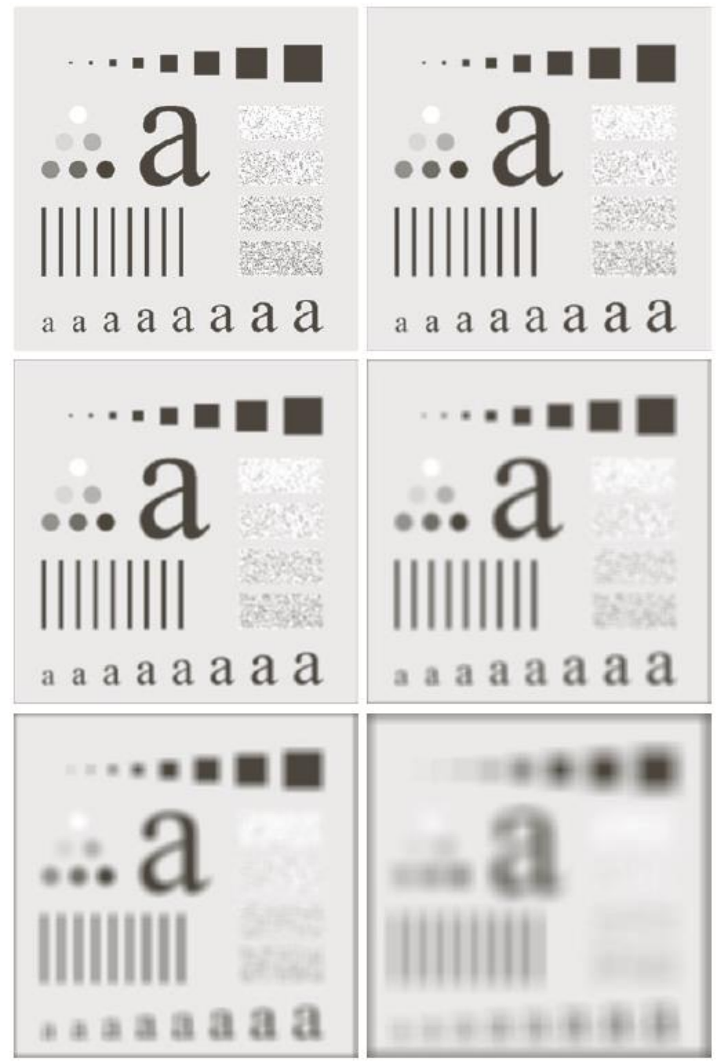
- No spoiler ☺

# Averaging filter

## Averaging filters

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

- The size of the filter can be increased
- Larger filters: stronger smoothing



a b
c d
e f

**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.

# Separable filters



- A square filter of size $n \times n$ may be separated in
  - A filter of size $n \times 1$
  - A filter of size $1 \times n$
- Separable filters: $w(x, y) \rightarrow w_x(x)w_y(y)$
  - Can be applied on rows, then columns (or vv)
  - O(MN(a+b)) instead of O(MNab) – faster

# 2D convolution



Original
*Cameraman*

*Cameraman* blurred by convolution
Filter impulse response

$$\frac{1}{25}\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & [1] & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

# Horizontal convolution



Original
*Cameraman*

*Cameraman* blurred horizontally
Filter impulse response

$$\frac{1}{5}\begin{pmatrix}1 & 1 & [1] & 1 & 1\end{pmatrix}$$

The filter doesn’t have to be squre

# Vertical convolution



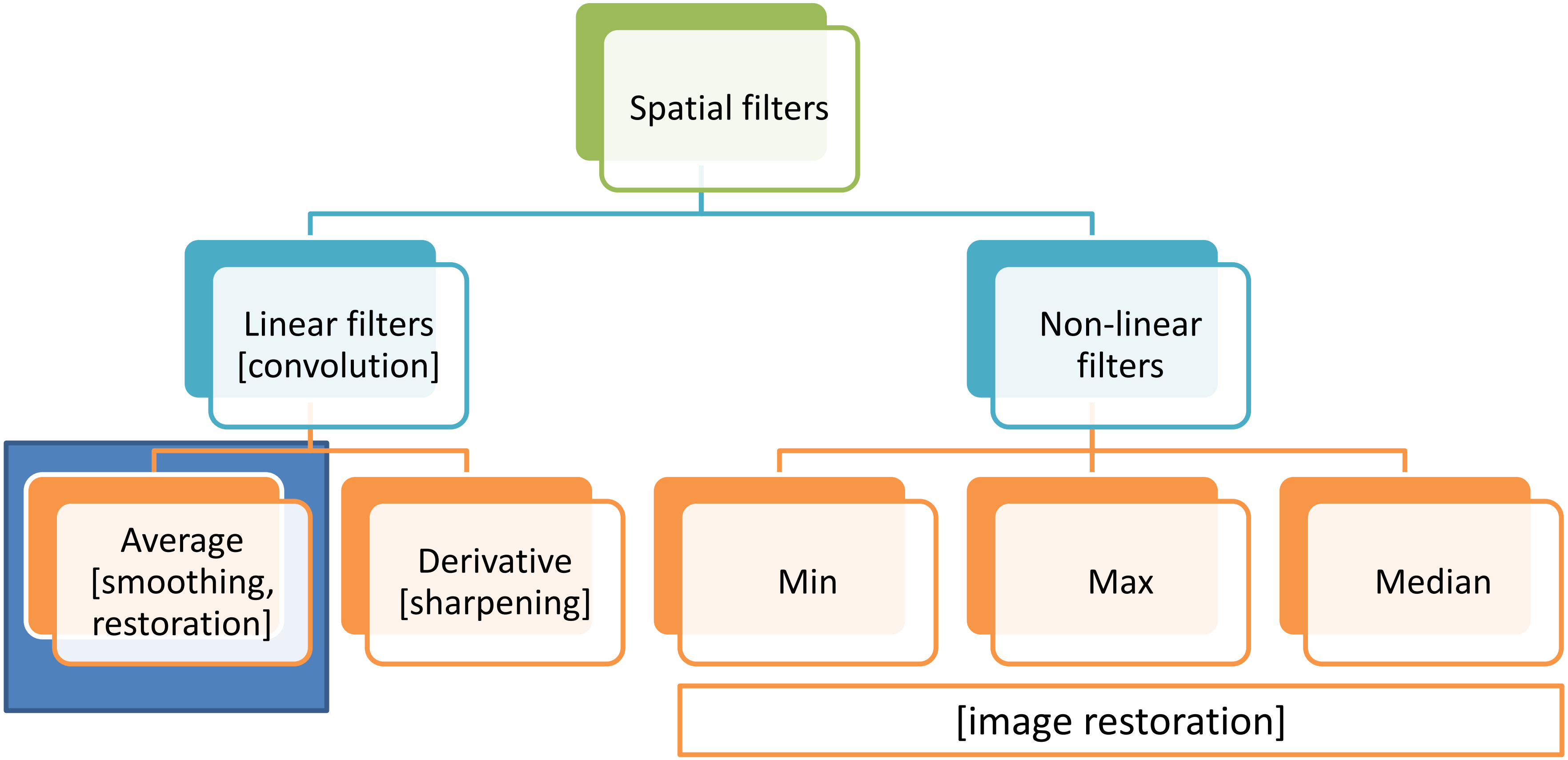
Original
*Cameraman*

*Cameraman* blurred vertically
Filter impulse response

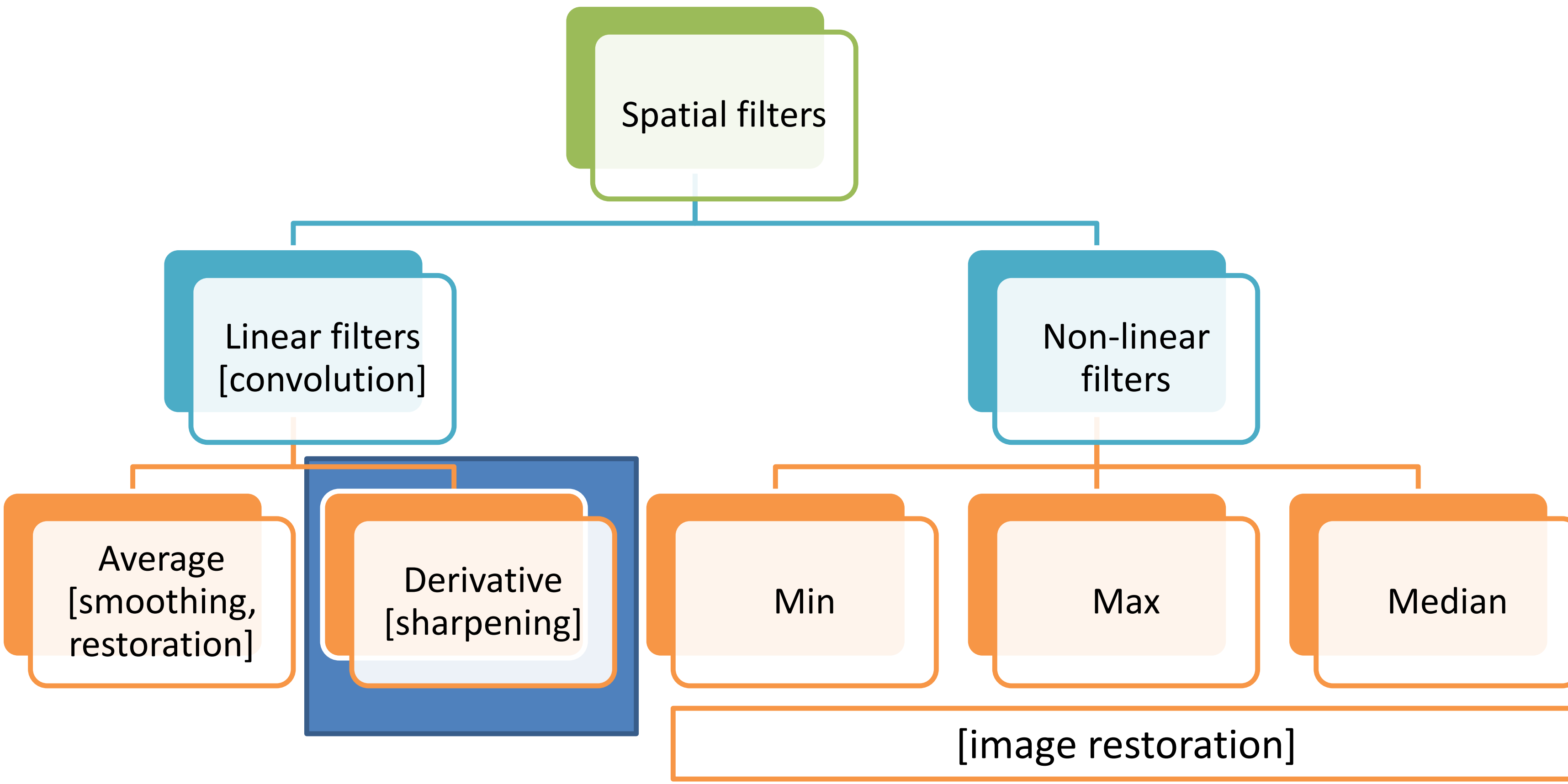
$$\frac{1}{5}\begin{pmatrix} 1 \\ 1 \\ [1] \\ 1 \\ 1 \end{pmatrix}$$

# Spatial filters overview





# Spatial filters overview



# Derivative operators



First order derivative operator

1. Is zero in flat segments
2. Is non-zero on the onset of a step/ramp
3. Is non-zero along ramps

# Derivative operators



First order derivative operator

1. Is zero in flat segments
2. Is non-zero on the onset of a step/ramp
3. Is non-zero along ramps

Second order derivative operator

1. Is zero in flat segments
2. Is non-zero on the onset and at the end of a step/ramp
3. Is zero along ramps of constant slope

# Derivative operators properties

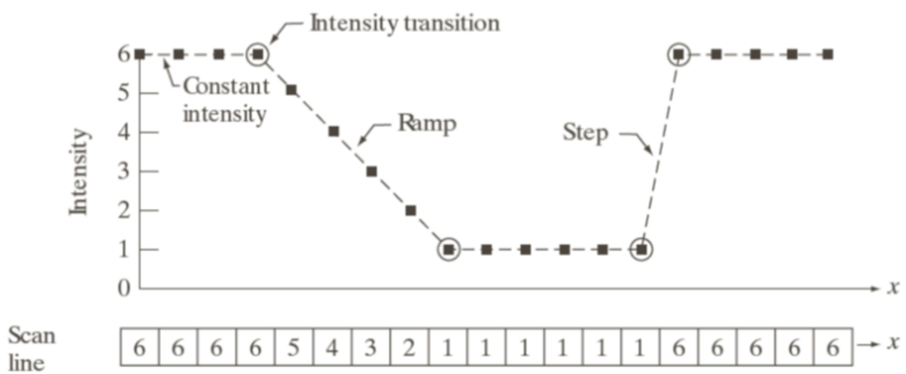


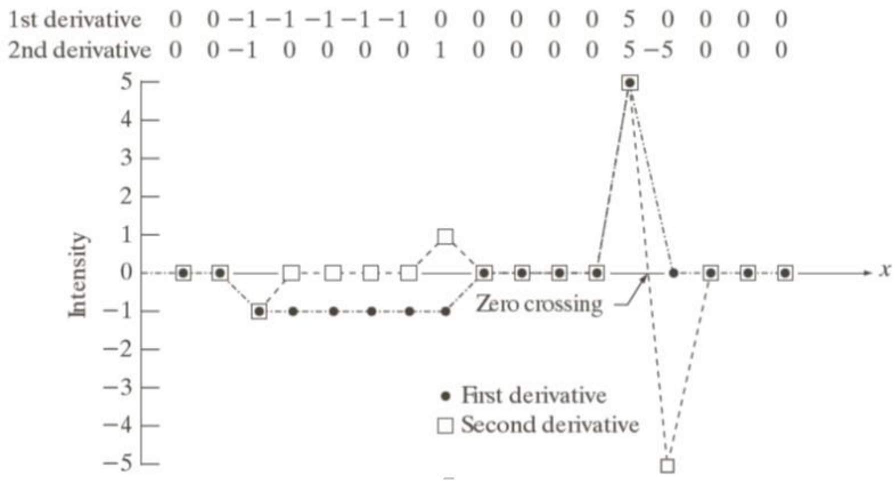
First order derivative operator

1. Is zero in flat segments
2. Is non-zero on the onset of a step/ramp
3. Is non-zero along ramps

Second order derivative operator

1. Is zero in flat segments
2. Is non-zero on the onset and at the end of a step/ramp
3. Is zero along ramps of constant slope



# Derivative of a discrete signal



- First order:

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

  - Computed at x + 1/2

- First order (alternative):

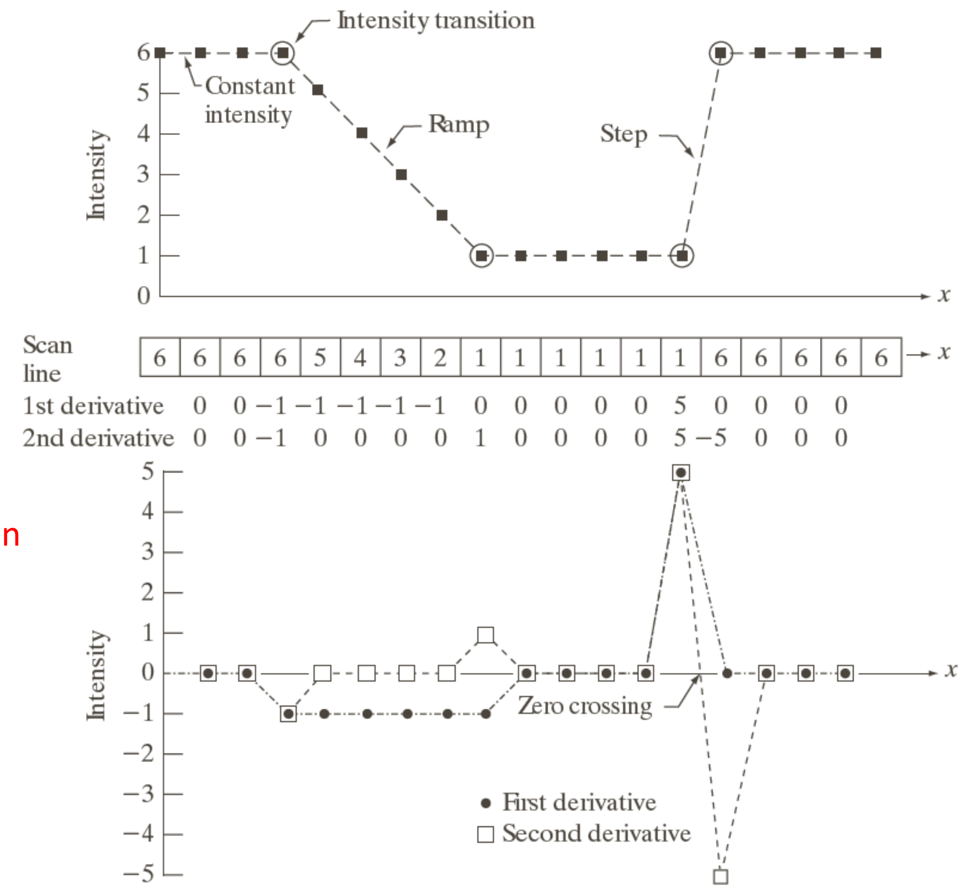
$$\frac{\partial f}{\partial x} = \frac{f(x+1)-f(x-1)}{2}$$

  - Computed at x, lower precision

- Second order:

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

  - Computed at x




a
b
c

**FIGURE 3.36** Illustration of the first and second derivatives of a 1-D digital function representing a section of a horizontal intensity profile from an image. In (a) and (c) data points are joined by dashed lines as a visualization aid.

# 1st order derivative

# Differential operators – 1st order

| Operator | Derivative order | Output | Linear | Isotropic |
|---|---|---|---|---|
| Gradient<br>$\nabla = [f_x, f_y]$ | 1st | 2D vector | Yes | No |
| Gradient module<br>$M = \sqrt{f_x^2 + f_y^2}$ | 1st | Scalar | No | Yes (ideally) |
| Approx. grad module<br>$M' = \lvert f_x \rvert + \lvert f_y \rvert$ | 1st | Scalar | No | No<br>(only 90°rot) |
| *Laplacian*<br>$\nabla^2 = f_{xx} + f_{yy}$ | *2nd* | *Scalar* | *Yes* | *Yes* |
| *Laplacian*<br>*(appr. with a mask)* | *2nd* | *Scalar* | *Yes* | *No*<br>*(only 90° rot)* |

# Gradient computation



- The derivative operations can be implemented using appropriate filters
  - Direct application
    - 2 elements for 1st order derivative
    - 3 elements for 2nd order derivative
  - Often, filters operate on larger areas
    - Different from a direct implementation of the derivative definition
    - More stable

# Gradient computation



*Simple differences*

| |
|---|
| 1 |
| -1 |

| | |
|---|---|
| 1 | -1 |

| | | |
|---|---|---|
| $z_1$ | $z_2$ | $z_3$ |
| $z_4$ | $z_5$ | $z_6$ |
| $z_7$ | $z_8$ | $z_9$ |

*Roberts*

| | |
|---|---|
| −1 | 0 |
| 0 | 1 |

| | |
|---|---|
| 0 | −1 |
| 1 | 0 |

*Sobel*

| | | |
|---|---|---|
| −1 | −2 | −1 |
| 0 | 0 | 0 |
| 1 | 2 | 1 |

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

| | | |
|---|---|---|
| −1 | 0 | 1 |
| −2 | 0 | 2 |
| −1 | 0 | 1 |

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

| | |
|---|---|
| | a |
| b | c |
| d | e |

**FIGURE 3.41** A 3 × 3 region of an image (the $z$s are intensity values). (b)–(c) Roberts cross gradient operators. (d)–(e) Sobel operators. All the mask coefficients sum to zero, as expected of a derivative operator.

Simple differences issues:

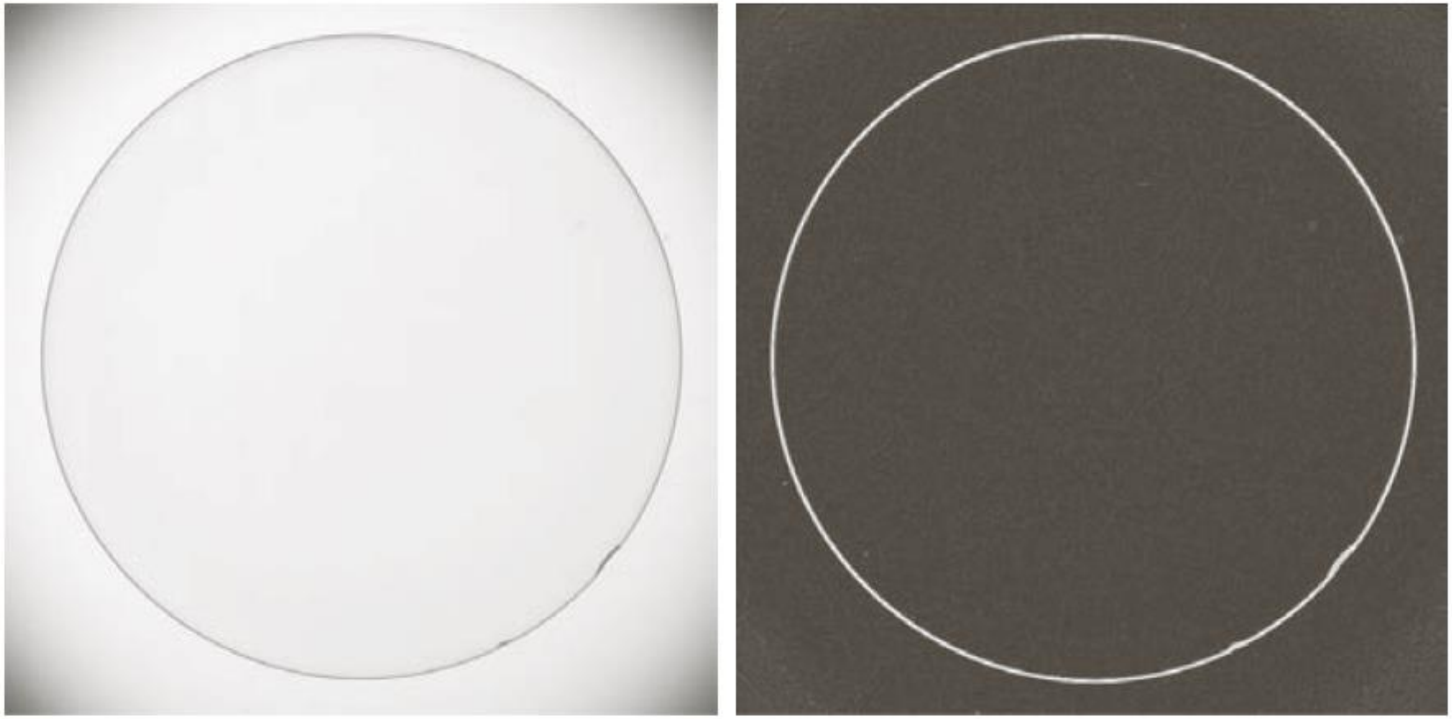
- Centered half-way between pixels
- Not robust to noise

Larger filters approximating the gradient are commonly used

What's the sum of the elements in the Sobel filters?

# Gradient computation – example





a b

**FIGURE 3.42** (a) Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Pete Sites, Perceptics Corporation.)

# 2nd order derivative

# Differential operators – 2nd order



| Operator | Derivative order | Output | Linear | Isotropic |
|---|---|---|---|---|
| *Gradient* $\nabla = [f_x, f_y]$ | *1st* | *2D vector* | *Yes* | *No* |
| *Gradient module* $M = \sqrt{f_x^2 + f_y^2}$ | *1st* | *Scalar* | *No* | *Yes (ideally)* |
| *Approx. grad module* $M' = \lvert f_x \rvert + \lvert f_y \rvert$ | *1st* | *Scalar* | *No* | *No (only 90°rot)* |
| Laplacian $\nabla^2 = f_{xx} + f_{yy}$ | 2nd | Scalar | Yes | Yes |
| Laplacian (appr. with a mask) | 2nd | Scalar | Yes | No (only 90° rot) |

# Laplacian filter



- The laplacian filter implements the equation

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

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

# Laplacian filter



- Laplacian filters

a b
c d

(a)

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | −4 | 1 |
| 0 | 1 | 0 |

(b)

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | −8 | 1 |
| 1 | 1 | 1 |

(c)

| | | |
|---|---|---|
| 0 | −1 | 0 |
| −1 | 4 | −1 |
| 0 | −1 | 0 |

(d)

| | | |
|---|---|---|
| −1 | −1 | −1 |
| −1 | 8 | −1 |
| −1 | −1 | −1 |

**FIGURE 3.37** (a) Filter mask used to implement Eq. (3.6-6). (b) Mask used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other implementations of the Laplacian found frequently in practice.

a
b c
d e

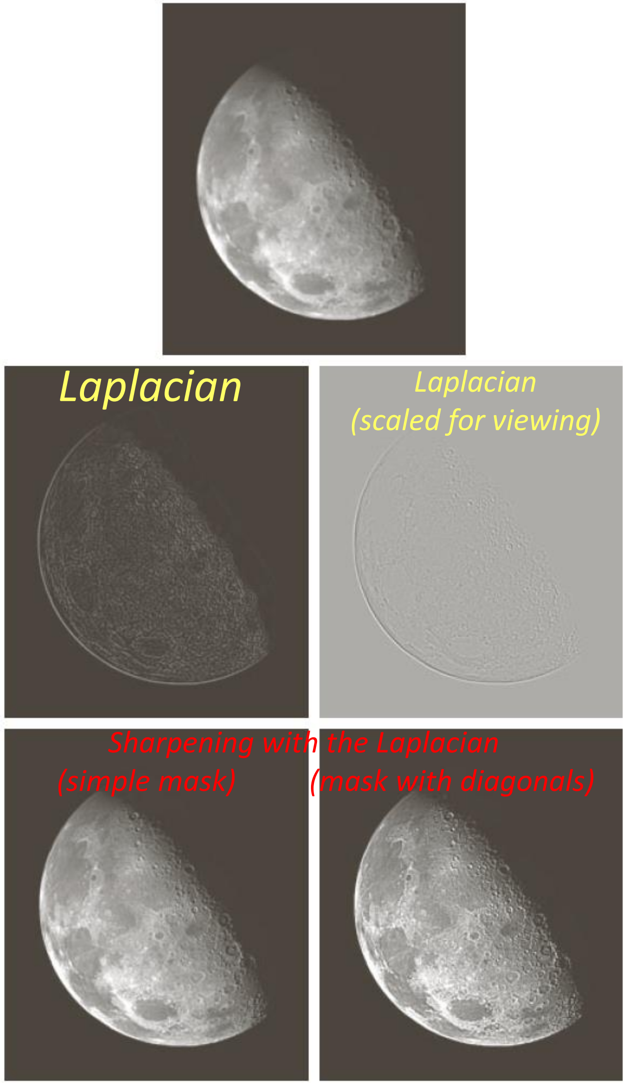
**FIGURE 3.38** (a) Blurred image of the North Pole of the moon. (b) Laplacian without scaling. (c) Laplacian with scaling. (d) Image sharpened using the mask in Fig. 3.37(a). (e) Result of using the mask in Fig. 3.37(b). (Original image courtesy of NASA.)

# Sharpening



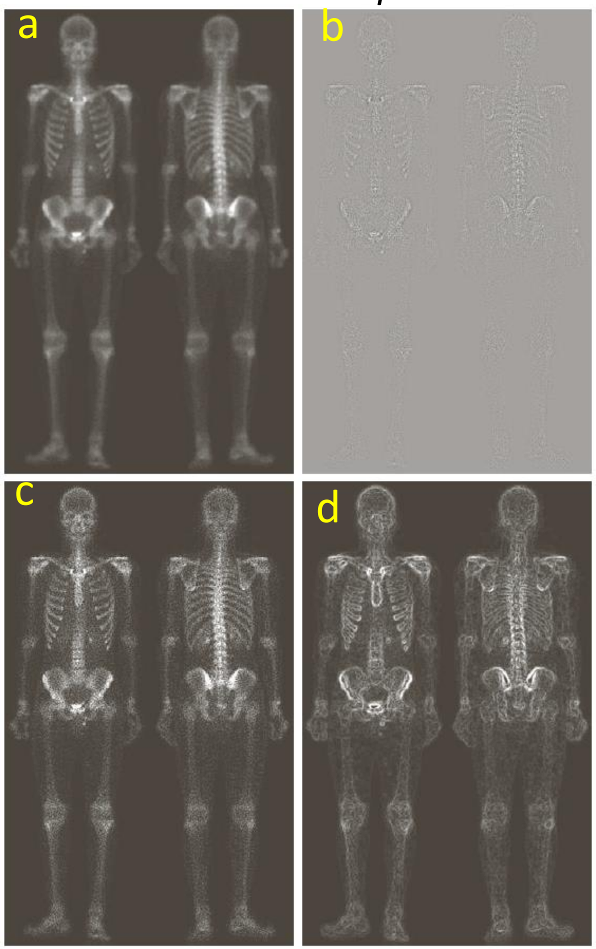
- The laplacian can be used to enhance the transitions in the image
- Combination of the image and the laplacian
  - Sharpening
  - Laplacian is subtracted if the center weight is negative

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



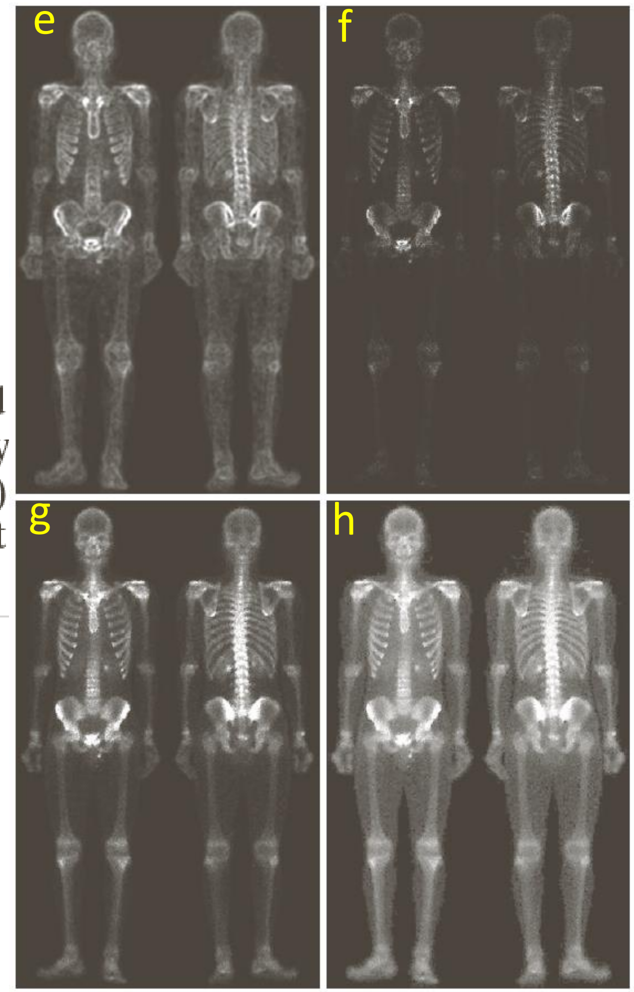
# Combination of multiple techniques






a b
c d

**FIGURE 3.43**
(a) Image of whole body bone scan. (b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b). (d) Sobel gradient of (a).




e f
g h

**FIGURE 3.43**
*(Continued)*
(e) Sobel image smoothed with a 5 × 5 averaging filter. (f) Mask image formed by the product of (c) and (e). (g) Sharpened image obtained by the sum of (a) and (f). (h) Final result obtained by applying a power-law transformation to (g). Compare (g) and (h) with (a). (Original image courtesy of G.E. Medical Systems.)

*Gonzalez 3.7*

UNIVERSITÀ DEGLI STUDI DI PADOVA

# Spatial filtering – linear filters

Stefano Ghidoni