Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Image Processing 08
Edge Detection
Part 1

SS 2020

Prof. Dr. Simone Frintrop

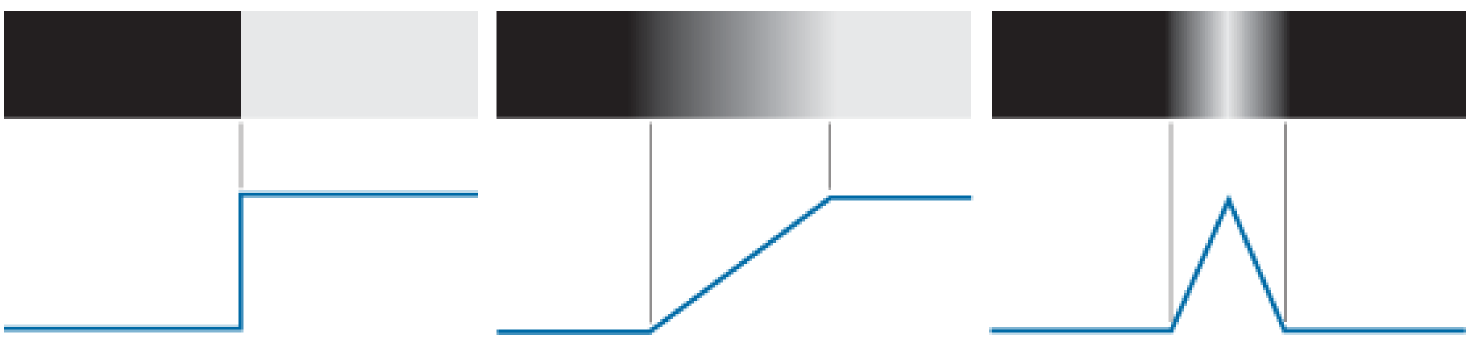
Computer Vision Group, Department of Informatics
University of Hamburg, Germany

Outline

- Part 1: Edge models and derivatives
- Part 2: Derivatives and edge detection in 1D
- Part 3: Partial derivatives, gradient, and edge detection in 2D
- Part 4: The effect of noise, Sobel & Prewitt filter
- Part 5: 2nd order derivatives and the Laplacian filter
- Part 6: Using the Laplacian for sharpening and line detection

Edges in Images

- Edges can be modeled according to their intensity profile
- We distinguish:
  – Step edges
  – Ramp edges
  – Roof edges



a b c

FIGURE 10.8 From left to right, models (ideal representations) of a step, a ramp, and a roof edge, and their corresponding intensity profiles.

[Gonzales/Woods]

Edge Detection

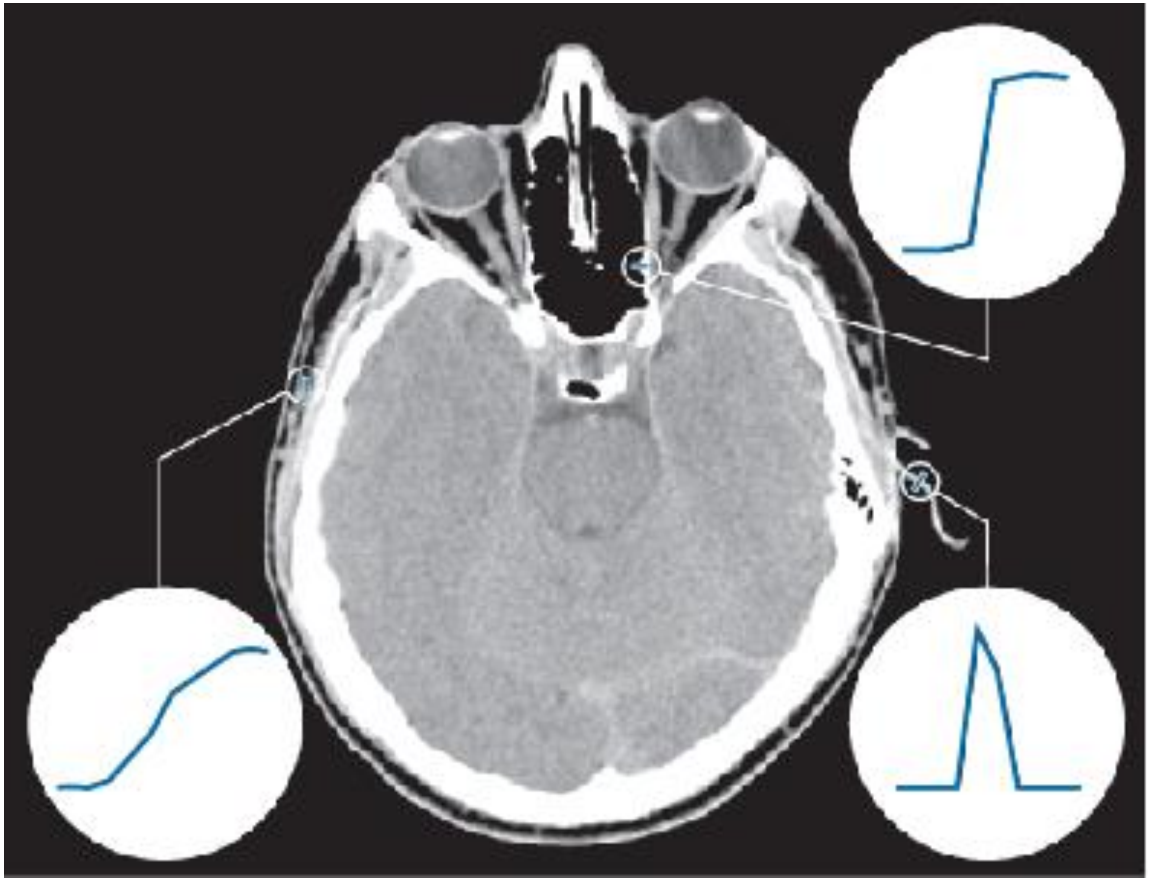
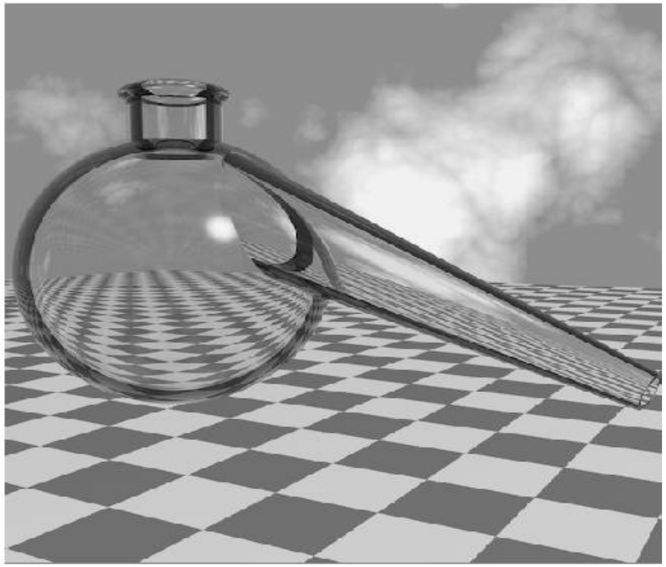


FIGURE 10.9 A 1508 × 1970 image showing (zoomed) actual ramp (bottom, left), step (top, right), and roof edge profiles. The profiles are from dark to light, in the areas enclosed by the small circles. The ramp and step profiles span 9 pixels and 2 pixels, respectively. The base of the roof edge is 3 pixels. (Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)

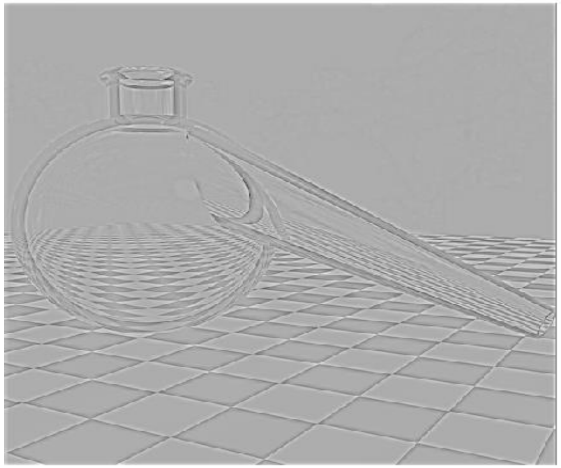
[Gonzales/Woods]

High-pass Filtering

- High-pass filtering lets the high frequencies pass
- → Edge detection corresponds to high-pass filtering



Original image

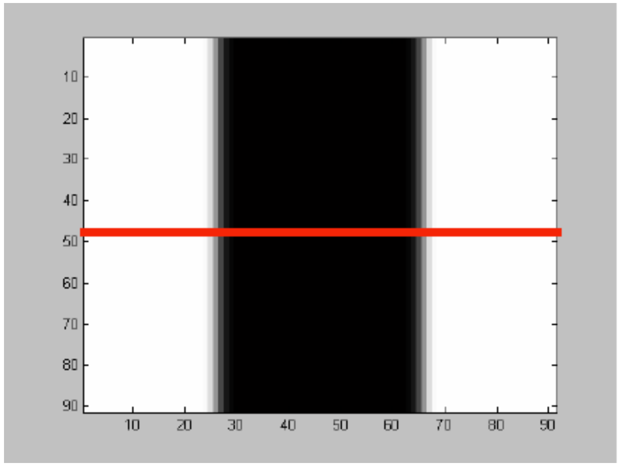


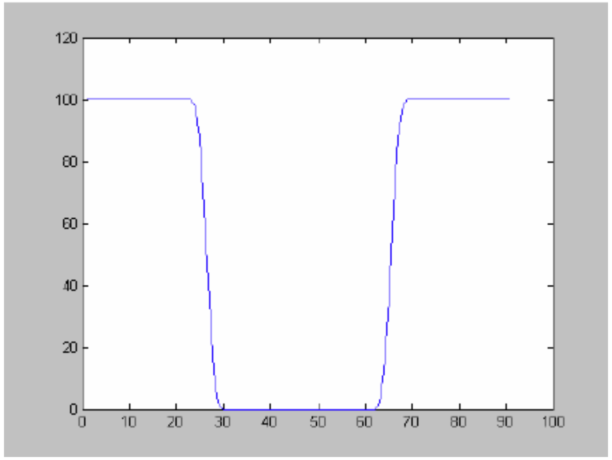
High-pass filtered

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Edges and Derivatives

- Remember: the image is a signal or function $f(x, y)$
- Edges are regions with a high slope
- What does the intensity profile of a slice through the below image at the red line look like?





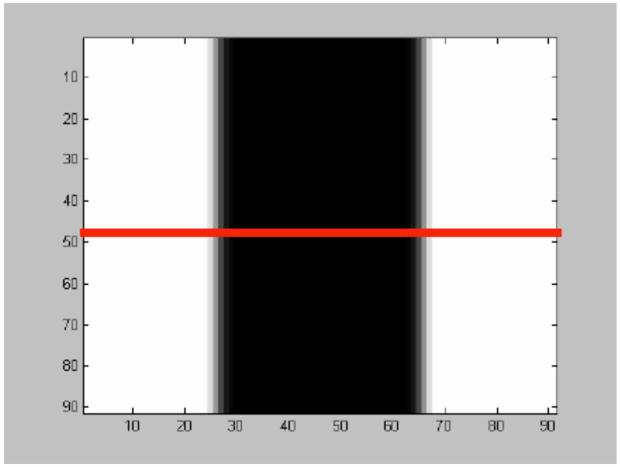
Intensity profile at red line

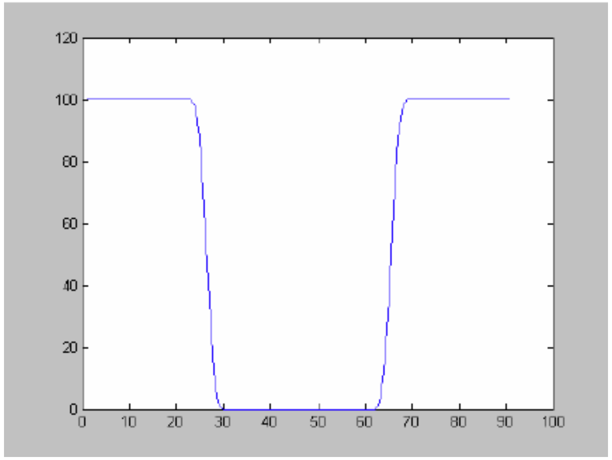
- High slopes in signals are found by derivatives
- Two approaches
  - Find maxima/minima in 1st derivative
  - Find zero-crossings in 2nd derivative

[Image: Bastian Leibe]

Question

What do the 1st and 2nd derivative of this function look like?

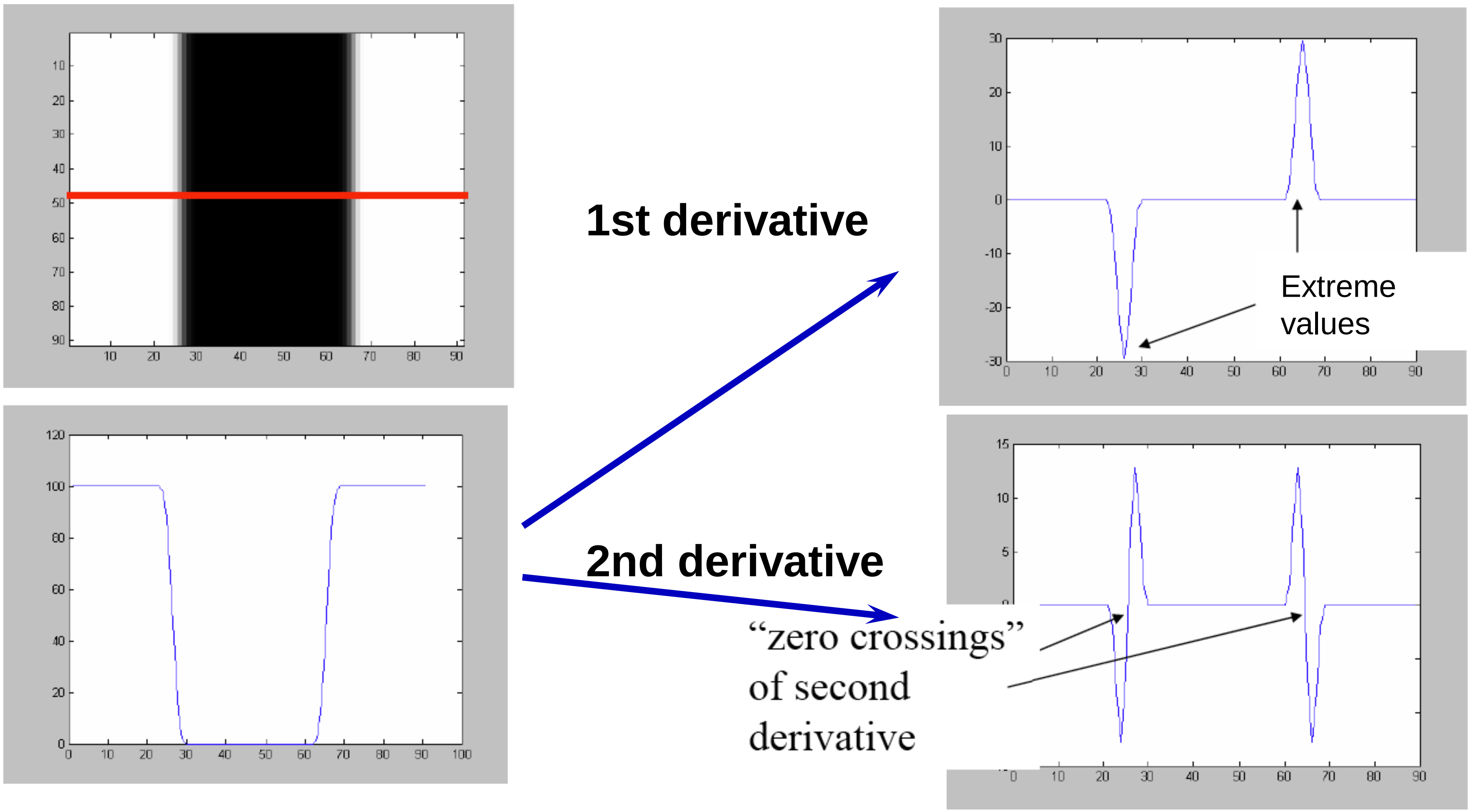




Intensity profile at red line

[Image: Bastian Leibe]

Edges and Derivatives



Outline

- Part 1: Edge models and derivatives
- Part 2: Derivatives and edge detection in 1D
- Part 3: Partial derivatives, gradient, and edge detection in 2D
- Part 4: The effect of noise, Sobel & Prewitt filter
- Part 5: 2nd order derivatives and the Laplacian filter
- Part 6: Using the Laplacian for sharpening and line detection

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Image Processing 08
Edge Detection
Part 2

SS 2020

Prof. Dr. Simone Frintrop

Computer Vision Group, Department of Informatics
University of Hamburg, Germany

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Outline

- Part 1: Edge models and derivatives
- Part 2: Derivatives and edge detection in 1D
- Part 3: Partial derivatives, gradient, and edge detection in 2D
- Part 4: The effect of noise, Sobel & Prewitt filter
- Part 5: 2nd order derivatives and the Laplacian filter
- Part 6: Using the Laplacian for sharpening and line detection

1D Derivatives

- Derivatives are defined in terms of differences
- Wikipedia: Differenzenquotient:

Definition [Bearbeiten | Quelltext bearbeiten]

Ist $f\colon D_f \to \mathbb{R}$ eine reellwertige Funktion, die im Bereich $D_f \subset \mathbb{R}$ definiert ist, und ist $[x_0; x_1] \subset D_f$, so nennt man den Quotienten

$$\varphi(x_1, x_0) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

Differenzenquotient von f im Intervall $[x_0; x_1]$.

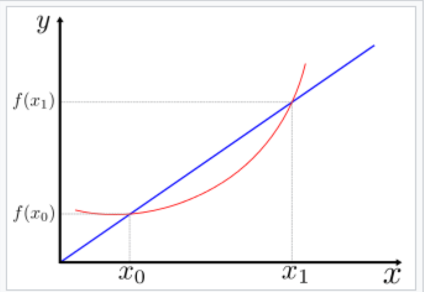
Schreibt man $\Delta x := x_1 - x_0$ und $\Delta y := f(x_1) - f(x_0)$, dann ergibt sich die alternative Schreibweise

$$\frac{\Delta y}{\Delta x} = \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$

Setzt man $h = x_1 - x_0$, also $x_1 = x_0 + h$, so erhält man die Schreibweise

$$\frac{f(x_0 + h) - f(x_0)}{h}.$$

Geometrisch entspricht der Differenzenquotient der Steigung der Sekante des Graphen von f durch die Punkte $(x_0, f(x_0))$ und $(x_1, f(x_1))$. Für $x_1 \to x_0$ bzw. $h \to 0$ wird aus der Sekante eine Tangente an der Stelle x_0.



Veranschaulichung des Differenzenquotienten: Er entspricht der Steigung der blauen Geraden

1D Derivatives

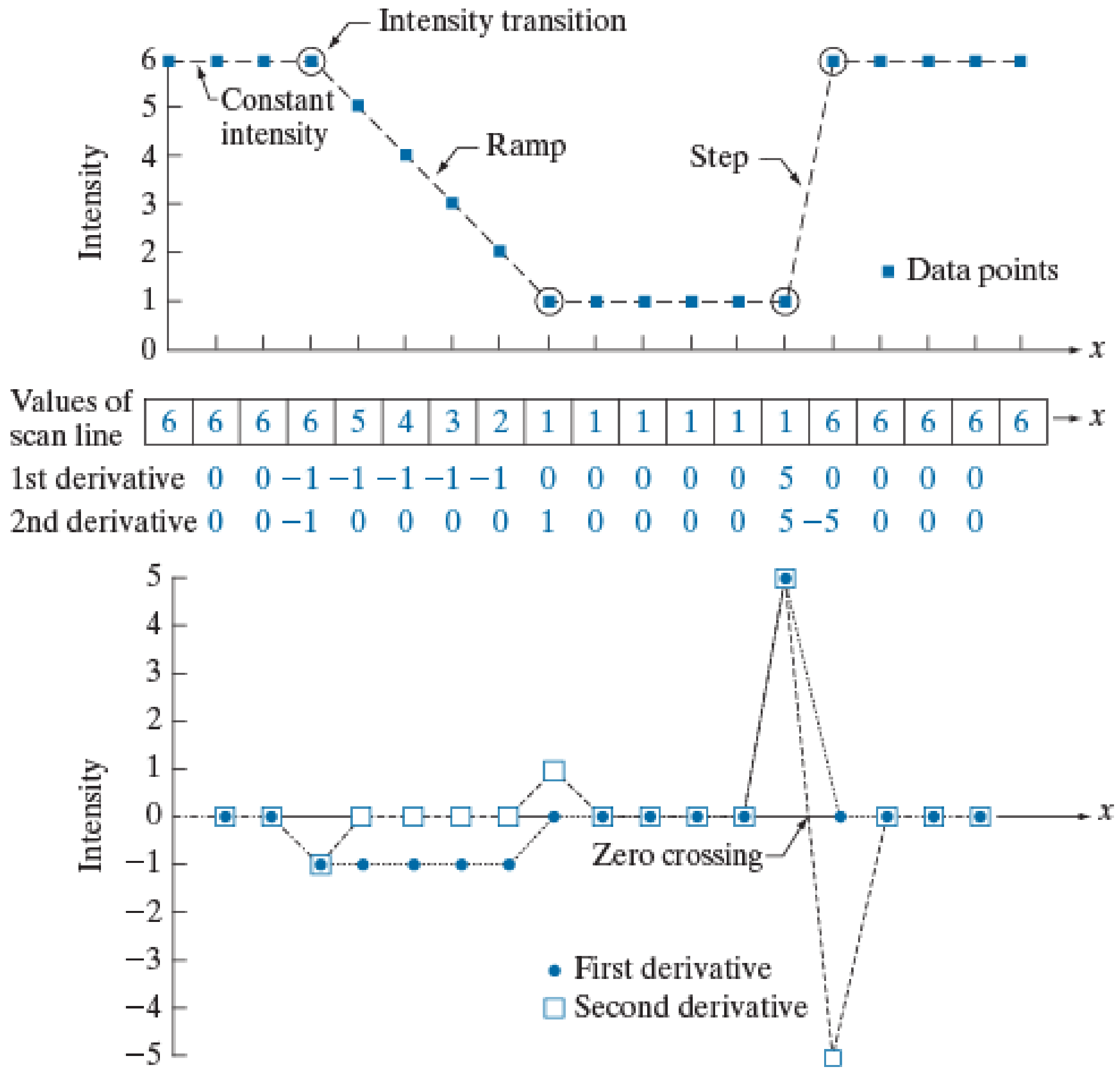
- Derivatives are defined in terms of differences
- Approximation of 1st order derivative with finite differences:
  - Forward difference: $f'(x) = f(x+1) - f(x)$
  - Backward difference: $f'(x) = f(x) - f(x-1)$
  - Central difference: $f'(x) = f(x+1) - f(x-1)$
    or: $f'(x) = f(x+0.5) - f(x-0.5)$

Image row	**0**	**0**	**0**	**0**	**0**	**255**	**255**	**255**	**255**	**255**
1st derivative, forward difference	**0**	**0**	**0**	**0**	**255**	**0**	**0**	**0**	**0**	**0**
1st derivative, backward difference:	**0**	**0**	**0**	**0**	**0**	**255**	**0**	**0**	**0**	**0**
1st derivative, central difference:	**0**	**0**	**0**	**0**	**255**	**255**	**0**	**0**	**0**	**0**

Edges and 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.

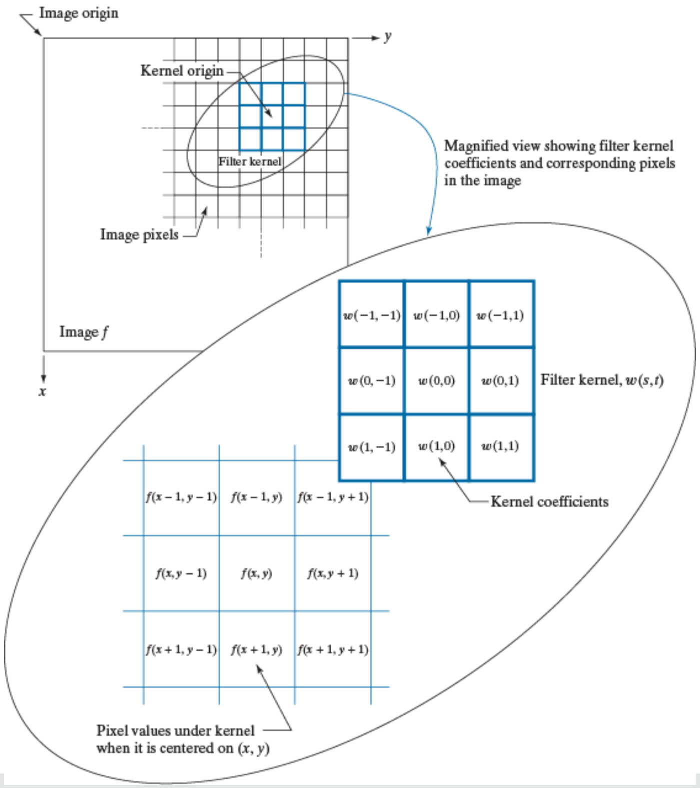


[Gonzales/Woods]

1D Derivatives

- How can we express the forward difference with correlation?

$$f'(x) = f(x+1) \quad - f(x)$$

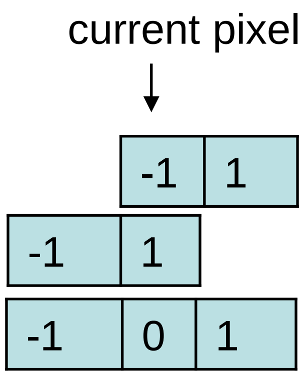


current pixel

-1	1

1D Derivatives

- Derivatives are defined in terms of differences
- Approximation of 1st order derivative with finite differences:
  - Forward difference: $f'(x) = f(x+1) - f(x)$
  - Backward difference: $f'(x) = f(x) - f(x-1)$
  - Central difference: $f'(x) = f(x+1) - f(x-1)$
    or: $f'(x) = f(x+0.5) - f(x-0.5)$



Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Derivatives

We find many different notations for derivatives:

First order derivative:

$$f'(x) = \frac{d}{dx} f(x) = \frac{df}{dx} = \frac{\partial}{\partial x} f(x) = D_x f(x) = D_x$$

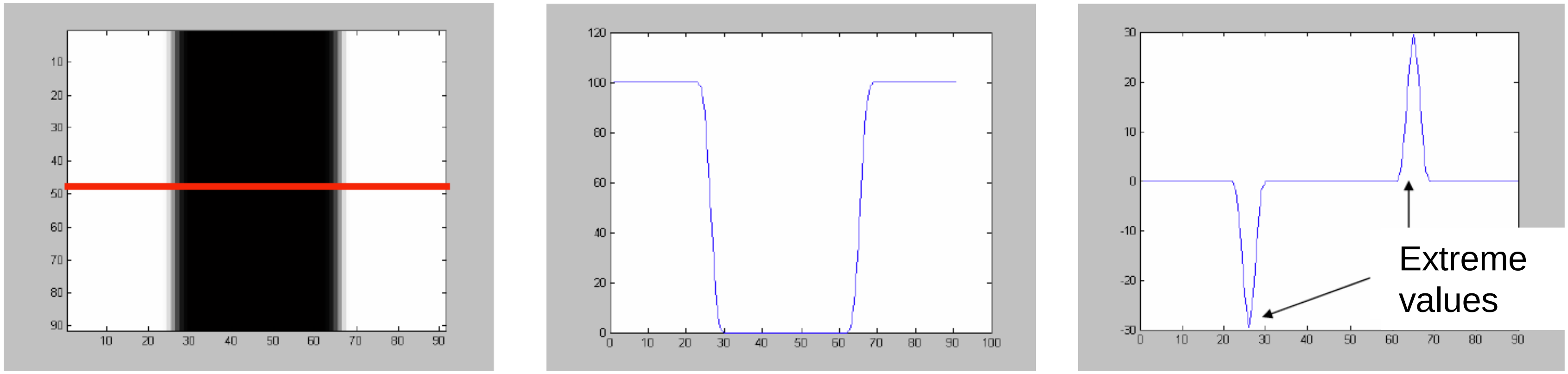
Second order derivative:

$$f''(x) = \frac{d^2}{dx^2} f(x) = \frac{d^2 f}{dx^2} = \frac{\partial^2}{\partial x^2} f(x) = D_x^2 f(x) = D_{xx}$$

Edge Detection

First ideas for edge detection with 1st derivative for a 1D signal without noise:

1. Take the derivative of each point (apply central difference filter)
2. Search for local extrema of the derivative value



[Image: Bastian Leibe]

Outline

- Part 1: Edge models and derivatives
- Part 2: Derivatives and edge detection in 1D
- Part 3: Partial derivatives, gradient, and edge detection in 2D
- Part 4: The effect of noise, Sobel & Prewitt filter
- Part 5: 2nd order derivatives and the Laplacian filter
- Part 6: Using the Laplacian for sharpening and line detection

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Image Processing 08
Edge Detection
Part 3

SS 2020

Prof. Dr. Simone Frintrop

Computer Vision Group, Department of Informatics
University of Hamburg, Germany

Outline

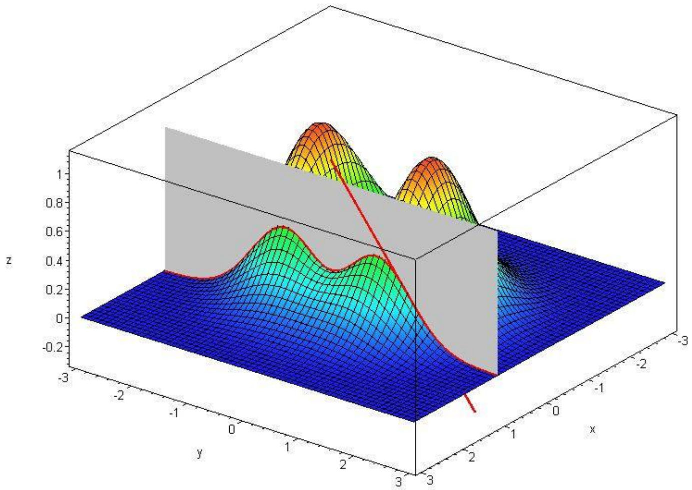
- Part 1: Edge models and derivatives
- Part 2: Derivatives and edge detection in 1D
- Part 3: Partial derivatives, gradient, and edge detection in 2D
- Part 4: The effect of noise, Sobel & Prewitt filter
- Part 5: 2nd order derivatives and the Laplacian filter
- Part 6: Using the Laplacian for sharpening and line detection

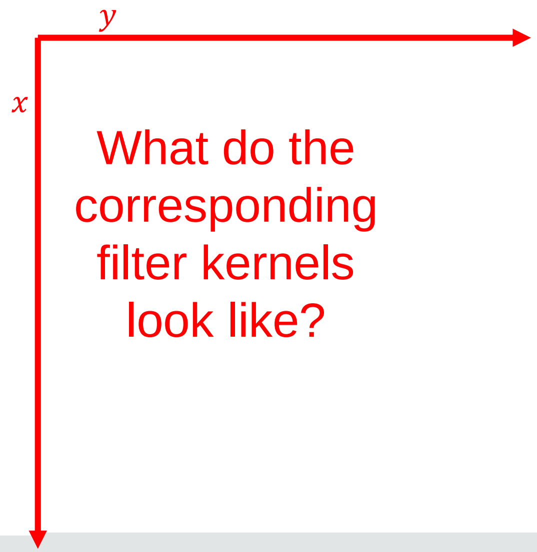
From 1D to 2D

Derivatives in images

Partial derivatives of function $f(x, y)$:

- In x-direction: $\dfrac{\partial f}{\partial x} = f(x+1, y) - f(x, y)$
- In y-direction: $\dfrac{\partial f}{\partial y} = f(x, y+1) - f(x, y)$





What do the corresponding filter kernels look like?

Derivatives in images

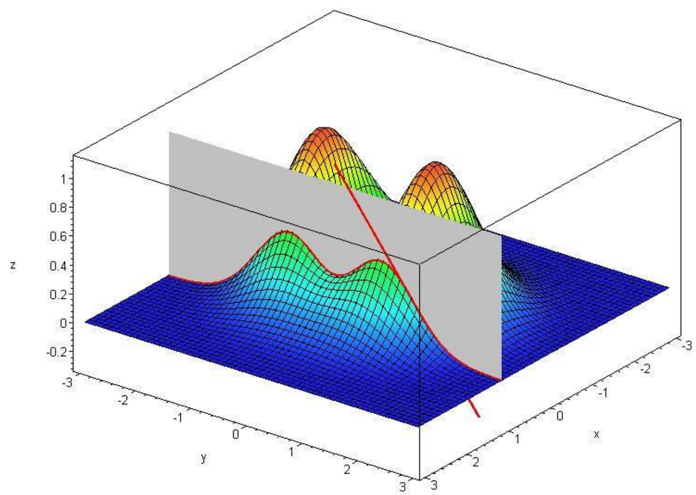
Partial derivatives of function $f(x, y)$:

- In x-direction: $$\frac{\partial f}{\partial x} = f(x+1, y) - f(x, y)$$

-1
1

- In y-direction: $$\frac{\partial f}{\partial y} = f(x, y+1) - f(x, y)$$

-1	1

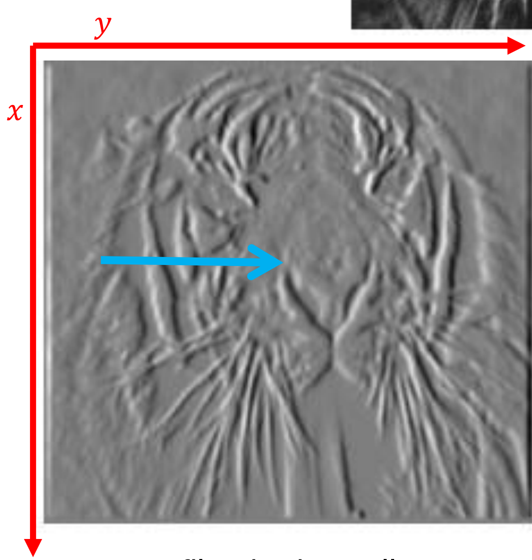


What do the corresponding filter kernels look like?

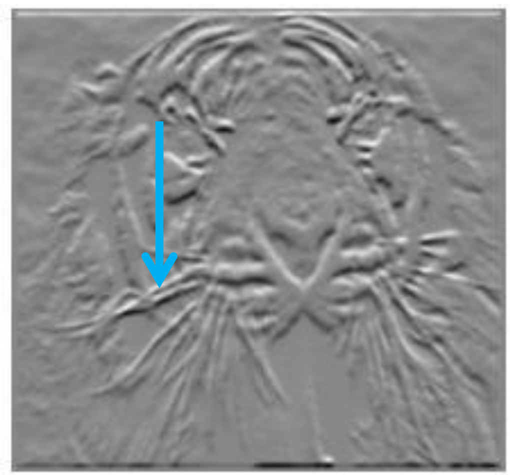
Partial Derivatives of an Image

$$\frac{\partial f(x,y)}{\partial y}$$

-1	1



Move filter horizontally,
Get vertical edges



Move filter vertically,
Get horizontal edges

$$\frac{\partial f(x,y)}{\partial x}$$

-1
1

Derivatives in images

Partial derivatives of function $f(x,y)$:

- In x-direction: $\frac{\partial f}{\partial x} = f(x+1,y) - f(x,y)$
- In y-direction: $\frac{\partial f}{\partial y} = f(x,y+1) - f(x,y)$
- Together, the partial derivatives form the gradient:

$$\nabla f = grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

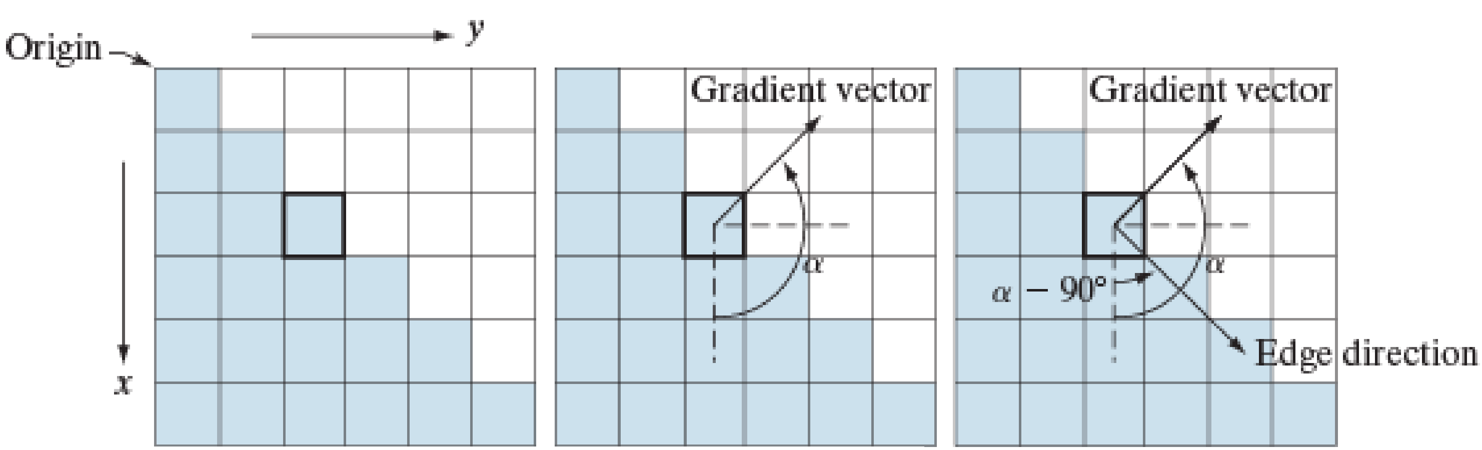
The gradient points into the direction of strongest increase of f

Derivatives in images

Many different notations for the gradient exist:

$$\nabla f = grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

$$\nabla f(x,y) \equiv gr[f(x,y)] \equiv \begin{bmatrix} g_x(x,y) \\ g_y(x,y) \end{bmatrix} = \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{bmatrix}$$



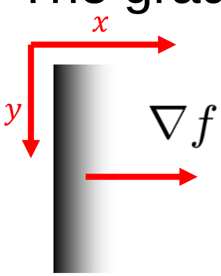
a b c

FIGURE 10.12 Using the gradient to determine edge strength and direction at a point. Note that the edge direction is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square represents one pixel. (Recall from Fig. 2.19 that the origin of our coordinate system is at the top, left.)

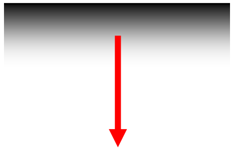
Image Gradient

Gradients in images: $\nabla f = grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]^T$

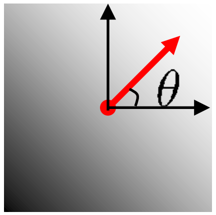
- The gradient points in the direction of most rapid intensity change



$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]^T$



$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]^T$



$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]^T$

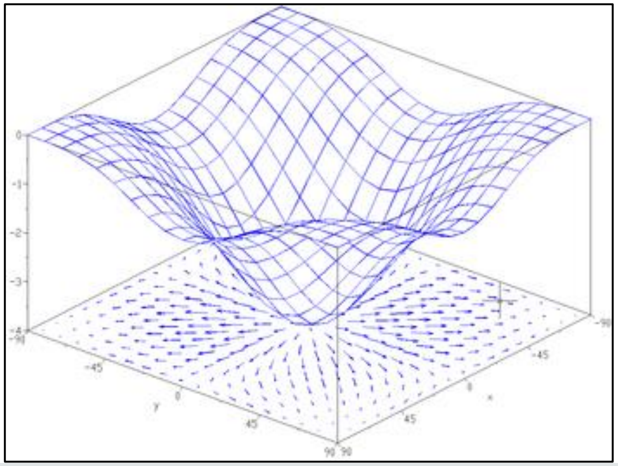


Image Gradient

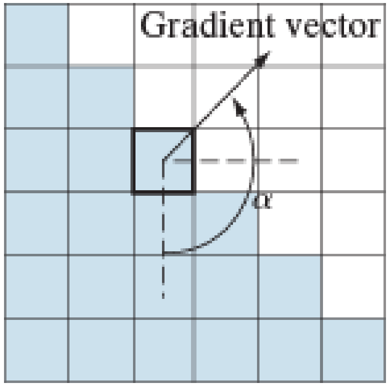
Gradients in images: $\nabla f = grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \end{bmatrix}^T$

- The gradient direction (orientation) is given by:

$$\alpha(x, y) = \tan^{-1}\left[\frac{g_y(x,y)}{g_x(x,y)}\right]$$

- The edge strength is given by the gradient magnitude

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



Edge Detection

First ideas for edge detection with 1st derivative for a 2D signal without noise:

1. Take the partial derivatives in x and y direction
2. Compute the gradient magnitude $\|\nabla f\| = \sqrt{(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2}$
3. Threshold on the gradient magnitude

Outline

- Part 1: Edge models and derivatives
- Part 2: Derivatives and edge detection in 1D
- Part 3: Partial derivatives, gradient, and edge detection in 2D
- Part 4: The effect of noise, Sobel & Prewitt filter
- Part 5: 2^{nd} order derivatives and the Laplacian filter
- Part 6: Using the Laplacian for sharpening and line detection

Image Processing 08
Edge Detection
Part 4

SS 2020

Prof. Dr. Simone Frintrop

Computer Vision Group, Department of Informatics
University of Hamburg, Germany

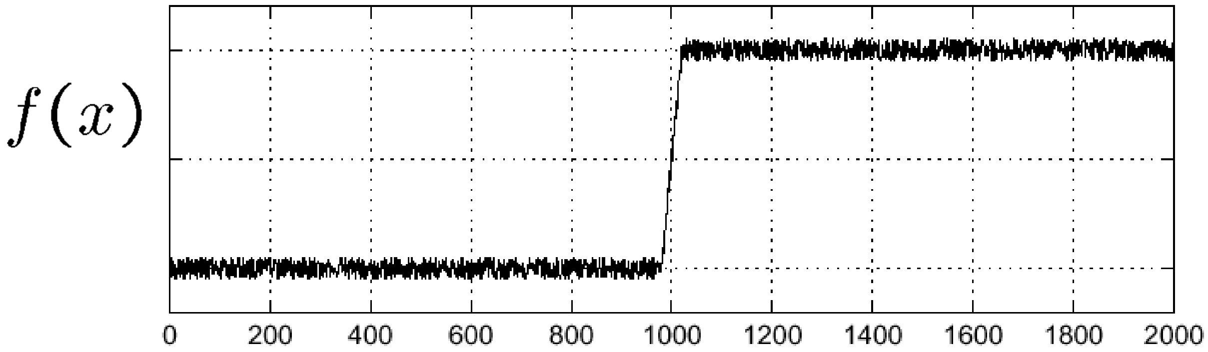
Outline

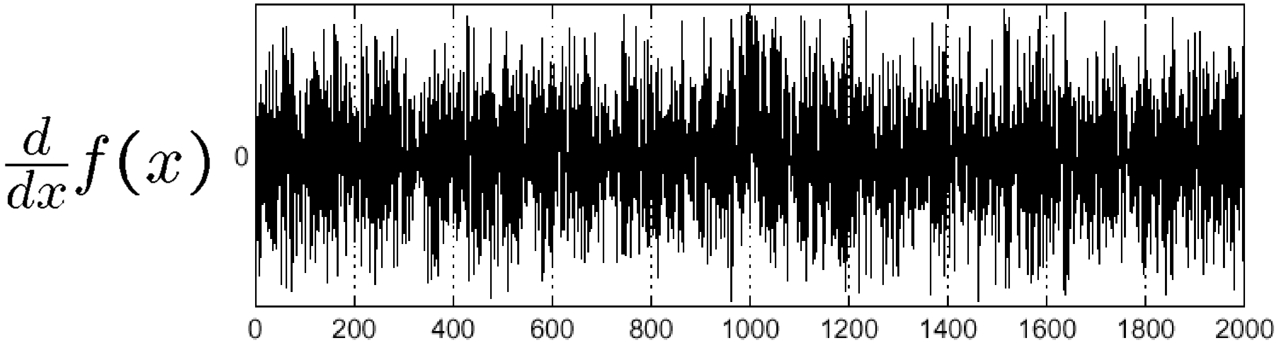
- Part 1: Edge models and derivatives
- Part 2: Derivatives and edge detection in 1D
- Part 3: Partial derivatives, gradient, and edge detection in 2D
- Part 4: The effect of noise, Sobel & Prewitt filter
- Part 5: 2nd order derivatives and the Laplacian filter
- Part 6: Using the Laplacian for sharpening and line detection

Effect of Noise

Noise affects the derivatives strongly:

Intensity function of a row of a noisy image:





Where is the edge?

Noise affects the derivatives strongly

Solution:

Smooth the image first or integrate smoothing into the edge filter

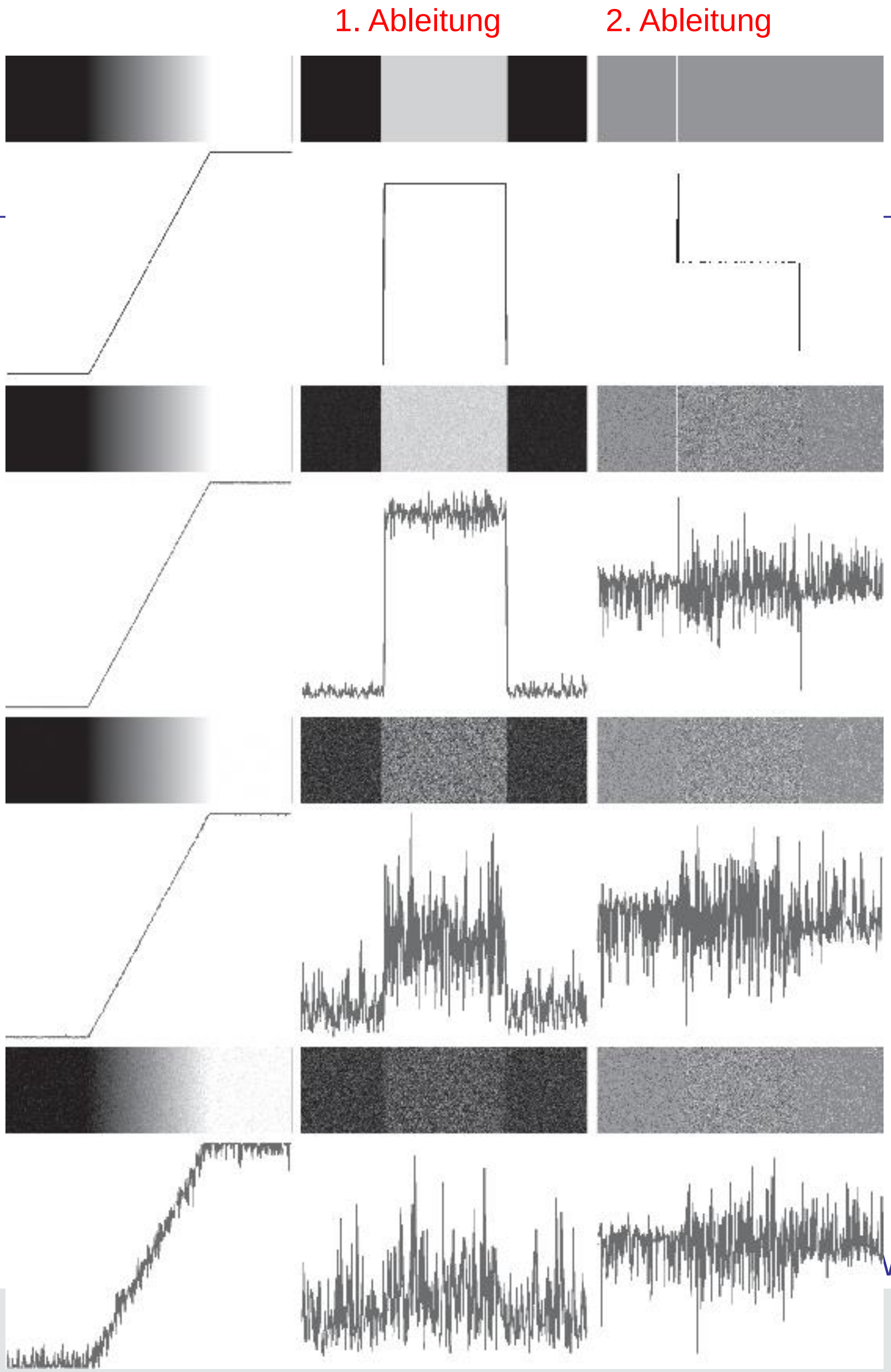


FIGURE 10.11 First column: 8-bit images with values in the range [0,255], and intensity profiles of a ramp edge corrupted by Gaussian noise of zero mean and standard deviations of 0.0, 0.1, 1.0, and 10.0 intensity levels, respectively. Second column: First-derivative images and intensity profiles. Third column: Second-derivative images and intensity profiles.

Voods]

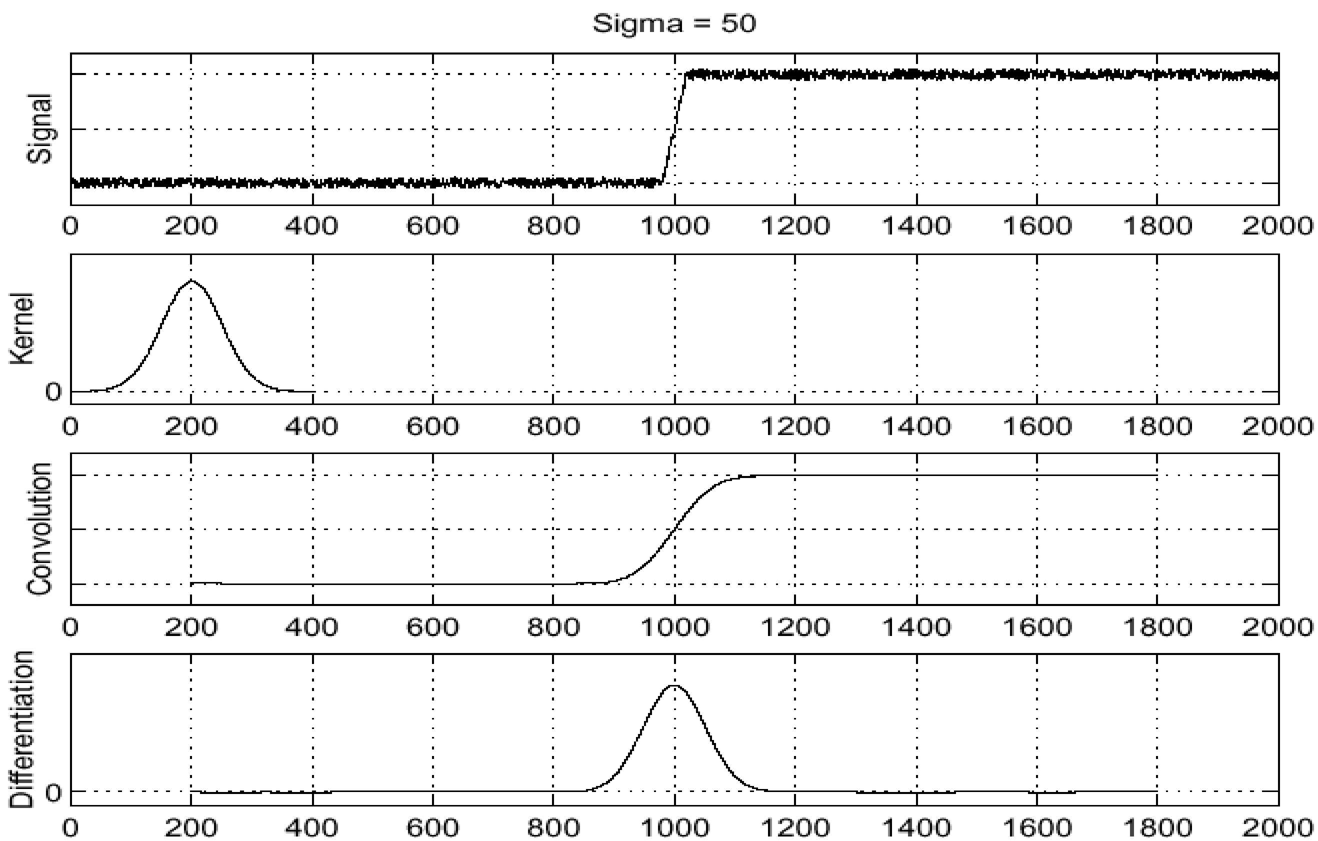
Solution: Smooth First

f

h

$h \star f$

$\frac{\partial}{\partial x}(h \star f)$



Where is the edge? Look for peaks in $\frac{\partial}{\partial x}(h \star f)$

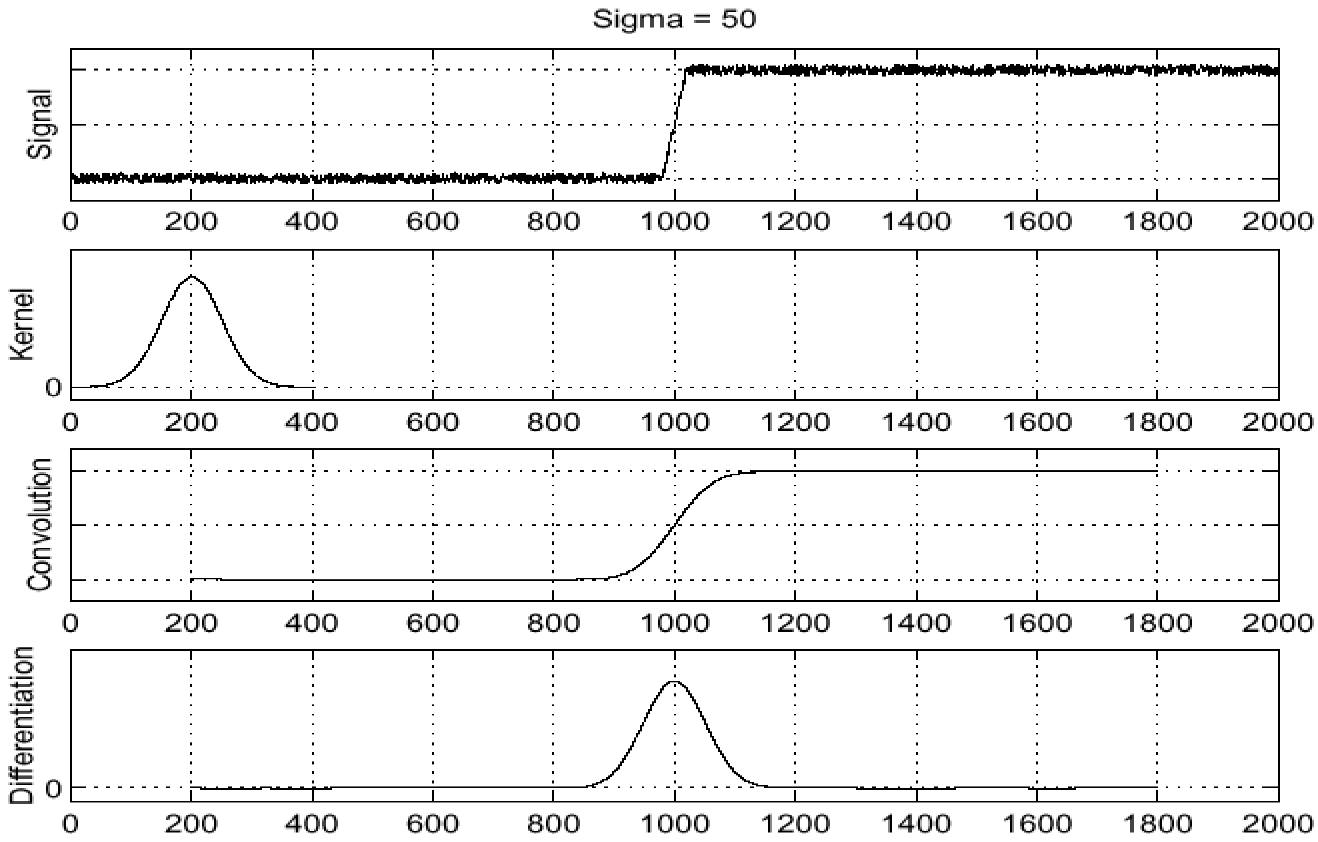
Solution: Smooth First

f

h

$h \star f$

$\frac{\partial}{\partial x}(h \star f)$



Where is the edge? Look for peaks in $\frac{\partial}{\partial x}(h \star f)$

Differentiation and Convolution

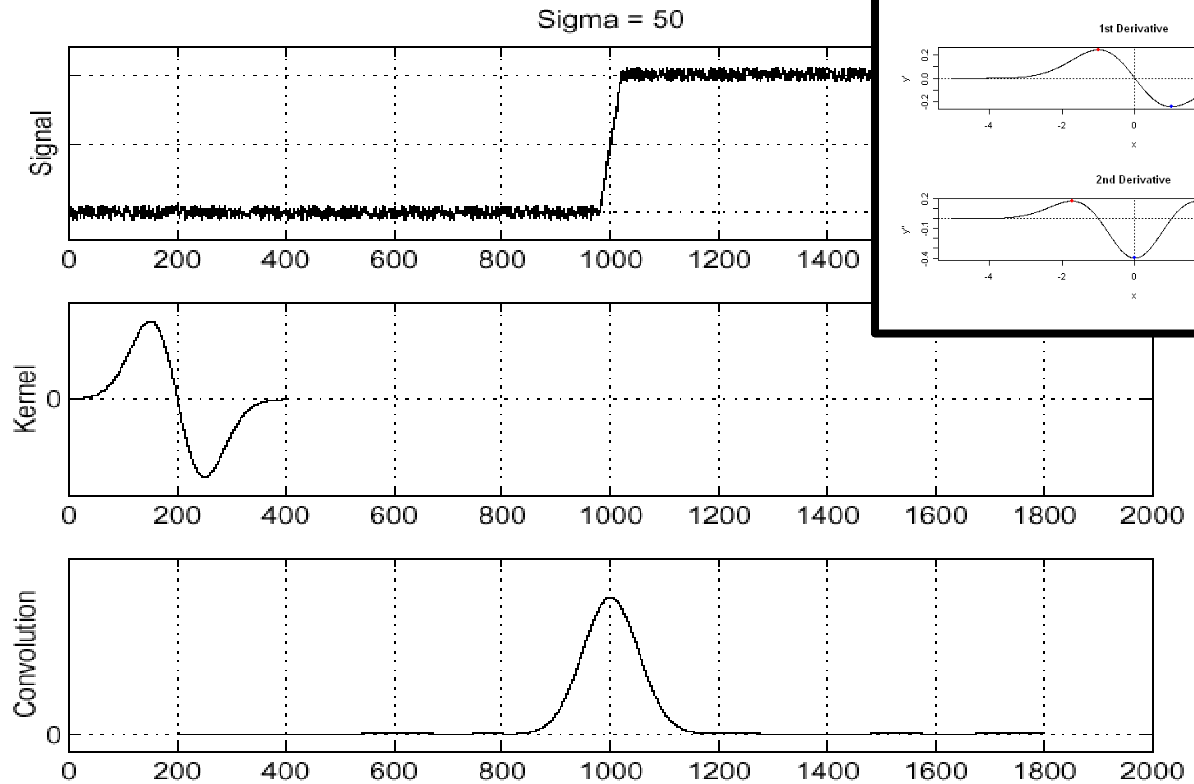
Remember: $\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$

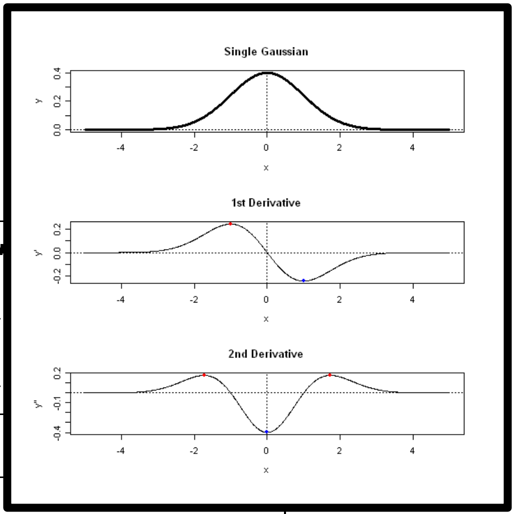
(Differentiation property of convolution)

$$f$$

$$\frac{\partial}{\partial x}h$$

$$(\frac{\partial}{\partial x}h) \star f$$





Edge Kernels

- A smoothing effect can be obtained by averaging over a 3x3 neighborhood, e.g. with the Prewitt operator:

-1	0	1
-1	0	1
-1	0	1

-1	-1	-1
0	0	0
1	1	1

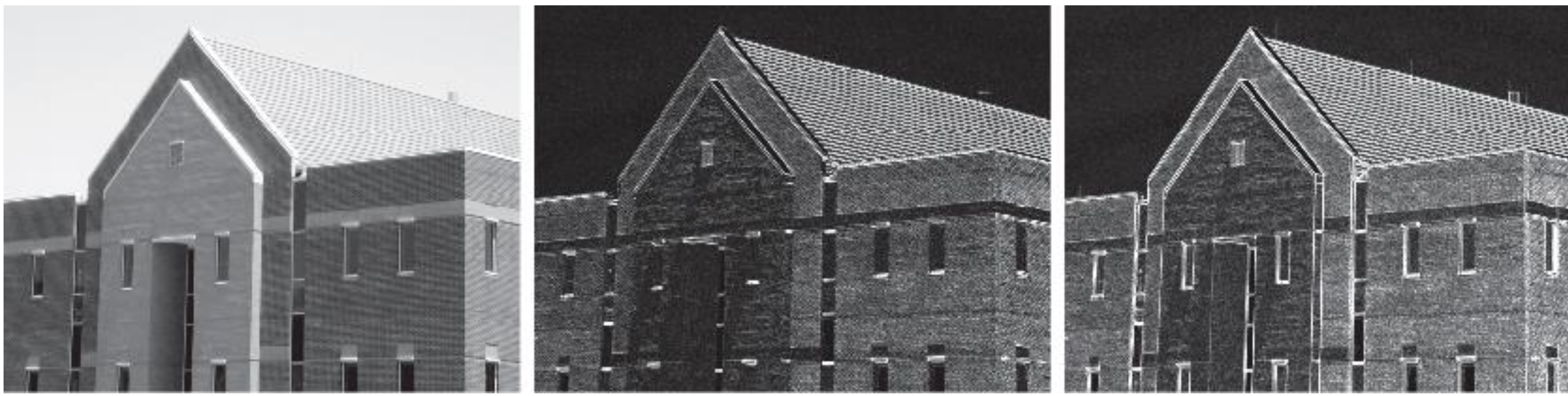
- or with the Sobel filter (even stronger smoothing effect):

-1	0	1
-2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

Applying Sobel

- Result from applying Sobel kernels to obtain the partial derivative images and the gradient magnitude image:



a b
c d

FIGURE 10.16 (a) Image of size 834×1114 pixels, with intensity values scaled to the range [0,1]. (b) $|g_x|$, the component of the gradient in the x-direction, obtained using the Sobel kernel in Fig. 10.14(f) to filter the image. (c) $|g_y|$, obtained using the kernel in Fig. 10.14(g).

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

[Gonzales/Woods]

Edge Kernels

- With these kernels, we can compute the partial derivatives g_x and g_y :

-1	0	1
-2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

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

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

Edge Detection

So, to summarize the 1st derivative approach for edge detection for a 2D signal with noise:

1. Smooth the image
2. Take the partial derivatives in x and y direction
3. Compute the gradient magnitude
4. Threshold on the gradient magnitude

1. and 2. can be combined into directly smoothing with derivatives of smoothing kernels, e.g., with the Sobel filters

Edge Detection

This results in this 1st derivative approach for edge detection for a 2D signal with noise:

1. Compute partial derivatives in x and y direction, e.g. by applying Sobel kernels
2. Compute the gradient magnitude
3. Threshold on the gradient magnitude

Applying Sobel

- Result from applying Sobel kernels to obtain the partial derivative images and the gradient magnitude image:

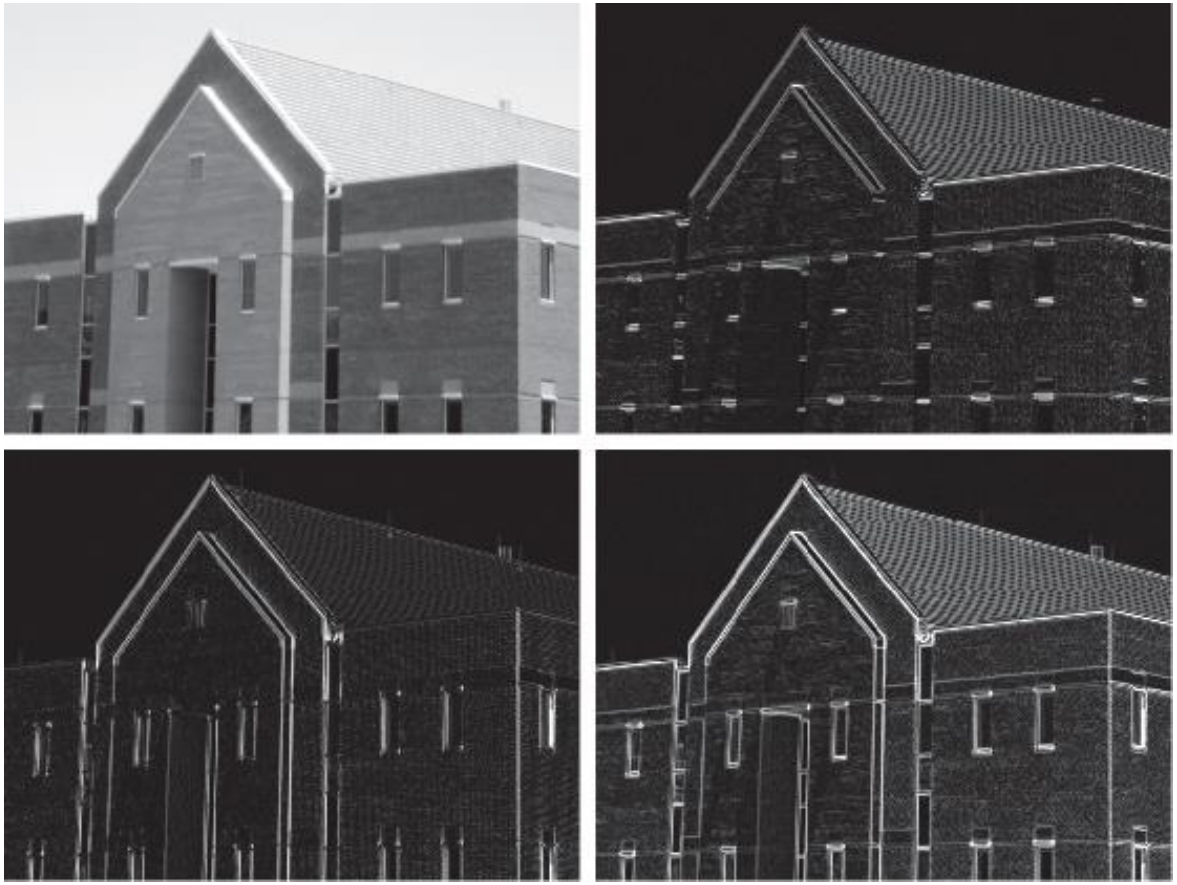
a b
c d

FIGURE 10.16 (a) Image of size 834×1114 pixels, with intensity values scaled to the range [0,1]. (b) $|g_x|$, the component of the gradient in the x-direction, obtained using the Sobel kernel in Fig. 10.14(f) to filter the image. (c) $|g_y|$, obtained using the kernel in Fig. 10.14(g). (d) The gradient image, $|g_x| + |g_y|$.

[Gonzales/Woods]

Applying Sobel

- To avoid responses to fine detail edges, it can be useful to smooth explicitely before applying Sobel:



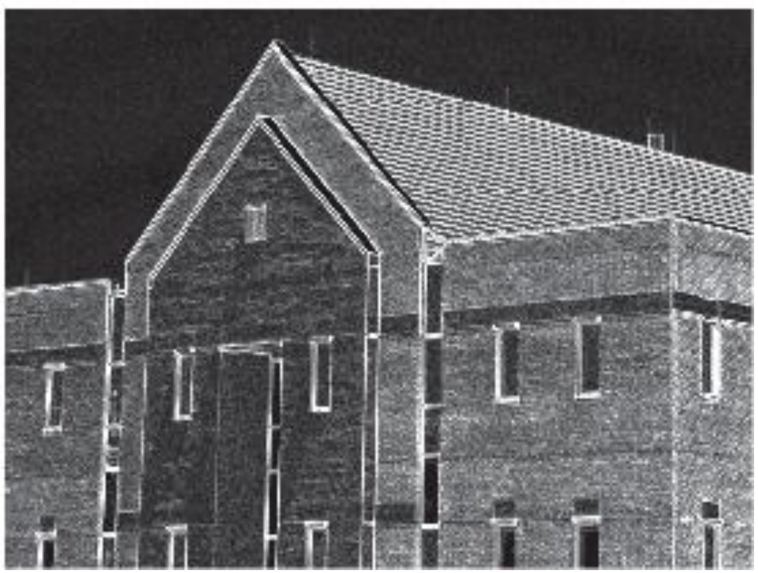
a b
c d

FIGURE 10.18 Same sequence as in Fig. 10.16, but with the original image smoothed using a 5 × 5 averaging kernel prior to edge detection.

[Gonzales/Woods]

Applying Sobel

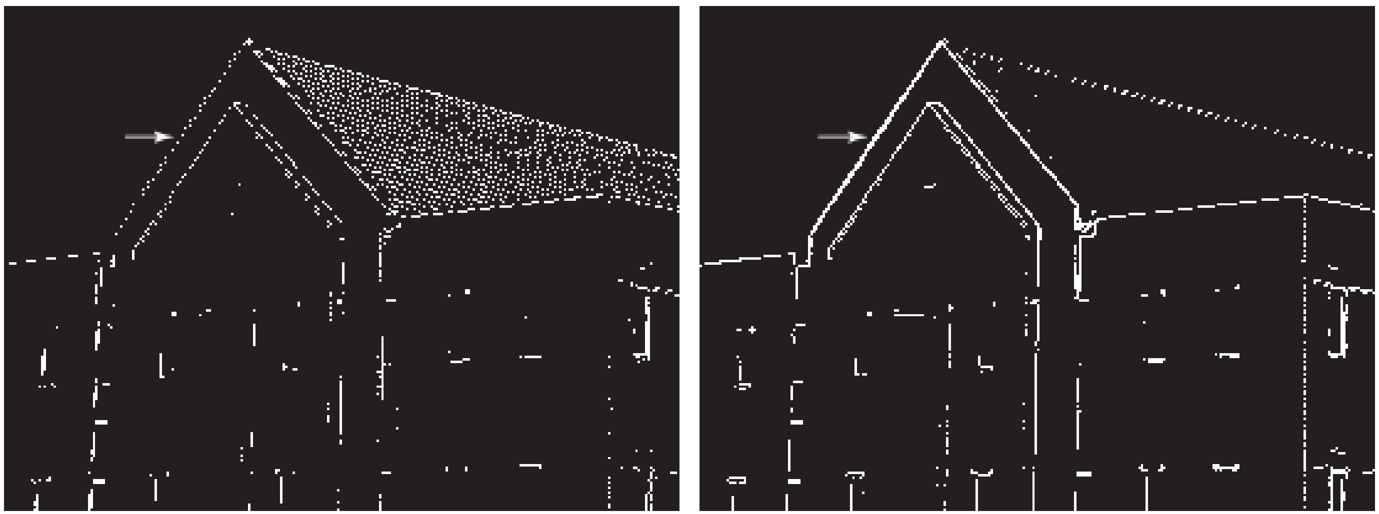
- Sobel gradient image without previous smooting (left) and with previous smoothing (right):



[Gonzales/Woods]

Applying Sobel

- After thresholding, we get a binary edge image:



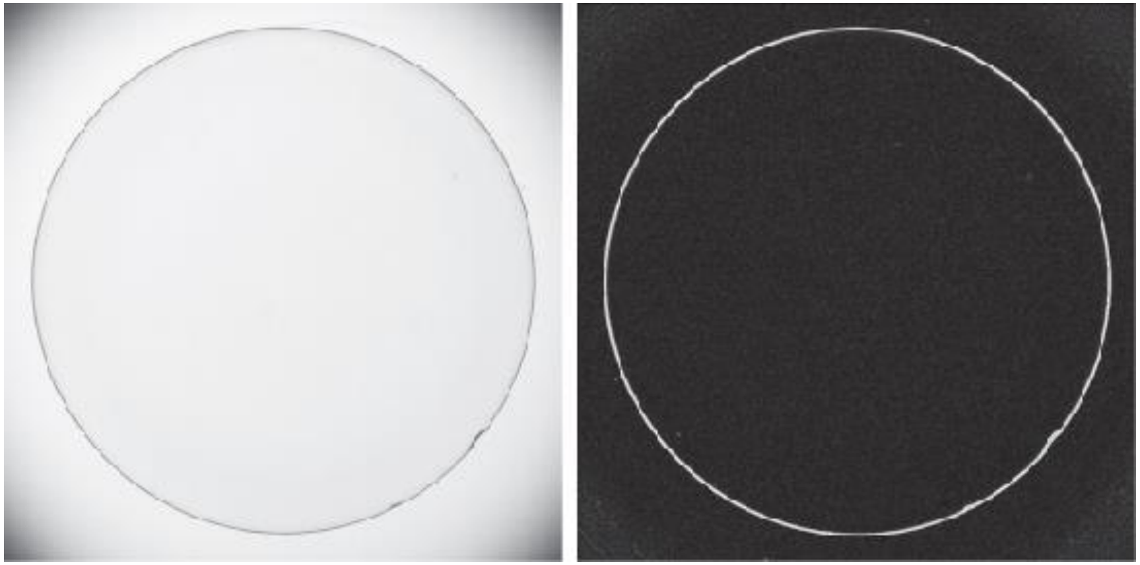
a b

FIGURE 10.20
(a) Result of thresholding Fig. 10.16(d), the gradient of the original image. (b) Result of thresholding Fig. 10.18(d), the gradient of the smoothed image.

[Gonzales/Woods]

Gradient for Edge Enhancement

- The gradient image is often used for industrial inspection:
  - To aid humans in detecting defects or
  - As preprocessing in automated inspections



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.)

[Gonzales/Woods]

Outline

- Part 1: Edge models and derivatives
- Part 2: Derivatives and edge detection in 1D
- Part 3: Partial derivatives, gradient, and edge detection in 2D
- Part 4: The effect of noise, Sobel & Prewitt filter
- Part 5: 2nd order derivatives and the Laplacian filter
- Part 6: Using the Laplacian for sharpening and line detection

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Image Processing 08
Edge Detection
Part 5

SS 2020

Prof. Dr. Simone Frintrop

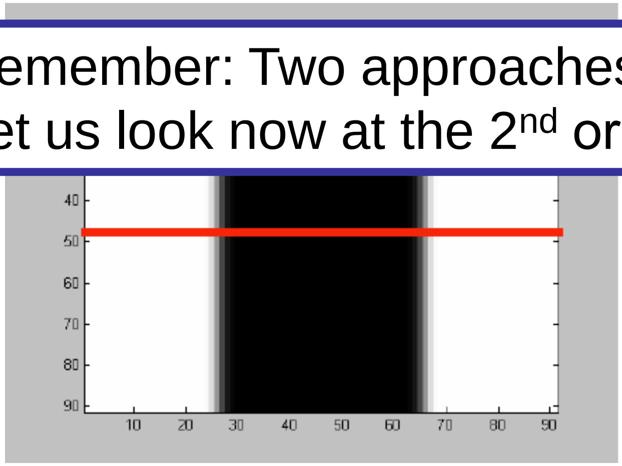
Computer Vision Group, Department of Informatics
University of Hamburg, Germany

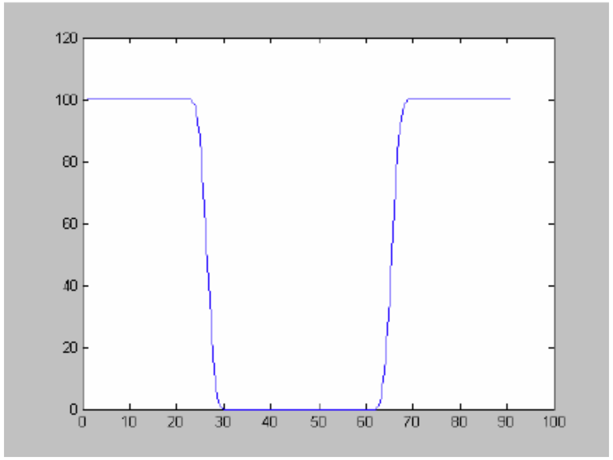
Outline

- Part 1: Edge models and derivatives
- Part 2: Derivatives and edge detection in 1D
- Part 3: Partial derivatives, gradient, and edge detection in 2D
- Part 4: The effect of noise, Sobel & Prewitt filter
- Part 5: 2nd order derivatives and the Laplacian filter
- Part 6: Using the Laplacian for sharpening and line detection

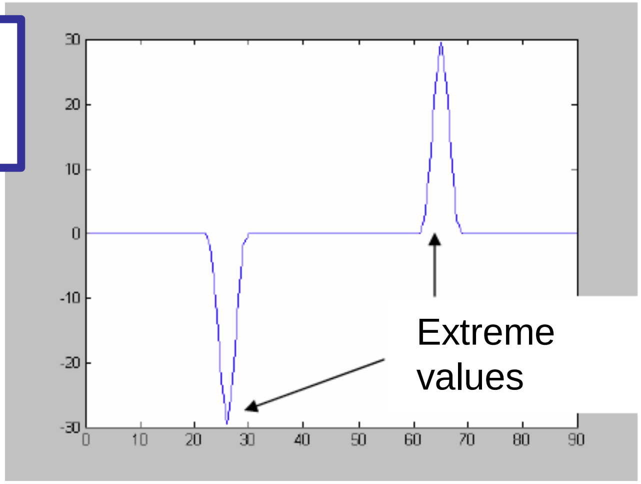
Derivatives and Edges…

Remember: Two approaches for edge detection
Let us look now at the 2nd order derivatives…



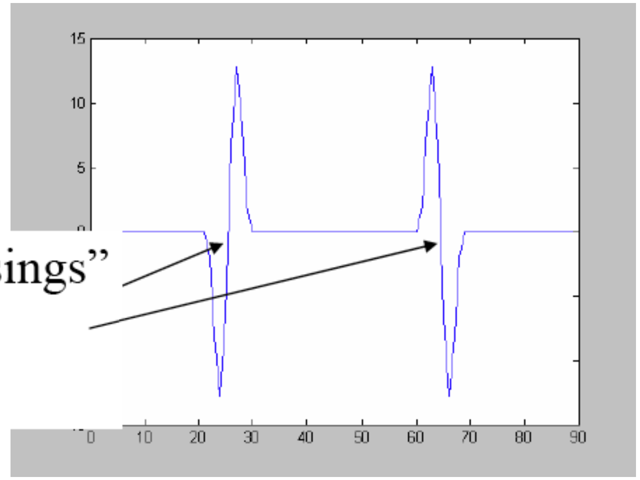


1st derivative



Extreme values

2nd derivative

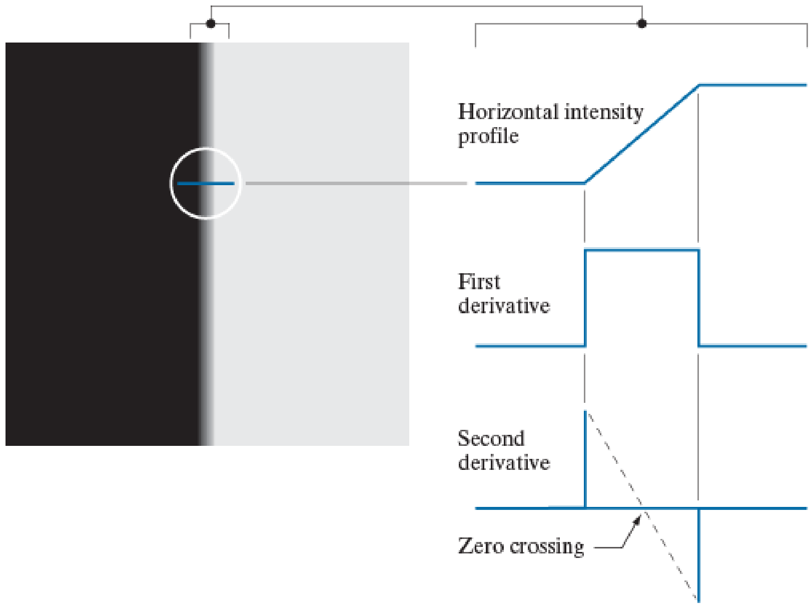


"zero crossings" of second derivative

Edge Detection

a b

FIGURE 10.10
(a) Two regions of constant intensity separated by an ideal ramp edge. (b) Detail near the edge, showing a horizontal intensity profile, and its first and second derivatives.



[Gonzales/Woods]

1D Derivatives

- 1^{st} order derivative (central difference):

$$f'(x) = f(x + 0.5) - f(x - 0.5)$$

2^{nd} order derivatives – 1D

- 1^{st} order derivative (central difference):

$$f'(x) = f(x+0.5) - f(x-0.5)$$

- 2^{nd} order derivative:

$$\begin{aligned} f''(x) &= (f(x+0.5) - f(x-0.5))' \\ &= f'(x+0.5) - f'(x-0.5) \\ &= (f(x+1) - f(x)) - (f(x) - f(x-1)) \\ &= f(x+1) - f(x) - f(x) + f(x-1) \\ &= f(x+1) - 2f(x) + f(x-1) \end{aligned}$$

What does the kernel look like?

- Kernel:

1	-2	1

2nd order derivatives – 2D

1D case, 2nd order derivative:

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

1	-2	1

Extended to a 2nd order partial derivative (2D case):

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

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

1
-2
1

1	-2	1

The Laplacian

2nd order partial derivative (2D case):

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

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

The **Laplacian** is a 2nd order differential operator

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

$$= f(x+1,y) - 2f(x,y) + f(x-1,y) + f(x,y+1) - 2f(x,y) + f(x,y-1)$$

$$= f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

The Laplacian

The **Laplacian**

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

What does the corresponding filter mask look like?

The Laplacian

The **Laplacian**

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

What does the corresponding filter mask look like?

0	1	0
1	-4	1
0	1	0

Note that this results from

$$\begin{array}{|c|c|c|}\hline 1 & -2 & 1 \\ \hline \end{array} + \begin{array}{|c|}\hline 1 \\ \hline -2 \\ \hline 1 \\ \hline \end{array}$$

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

The Laplacian

There are different variants of the Laplacian kernel:

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

a b c d

FIGURE 3.51 (a) Laplacian kernel used to implement Eq. (3-62). (b) Kernel used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other Laplacian kernels.

[Gonzales/Woods]

The Laplacian

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).

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.

[Gonzales/Woods]

Edge Detection

The 2nd derivative approach for edge detection for a 2D signal with noise:

1. Smooth the image
2. Apply the Laplacian operator
3. Search for zero crossings
   (zero crossing at x: at least two opposing neighbors have different signs)

1\. and 2. can be combined into directly smoothing with a 2nd derivative of a smoothing kernel, e.g., a Laplacian of Gaussian

Outline

- Part 1: Edge models and derivatives
- Part 2: Derivatives and edge detection in 1D
- Part 3: Partial derivatives, gradient, and edge detection in 2D
- Part 4: The effect of noise, Sobel & Prewitt filter
- Part 5: 2nd order derivatives and the Laplacian filter
- Part 6: Using the Laplacian for sharpening and line detection

Image Processing 08
Edge Detection
Part 6

SS 2020

Prof. Dr. Simone Frintrop

Computer Vision Group, Department of Informatics
University of Hamburg, Germany

Outline

- Part 1: Edge models and derivatives
- Part 2: Derivatives and edge detection in 1D
- Part 3: Partial derivatives, gradient, and
  edge detection in 2D
- Part 4: The effect of noise, Sobel & Prewitt filter
- Part 5: 2^{nd} order derivatives and the Laplacian filter
- Part 6: Using the Laplacian for sharpening and line
  detection

The Laplacian

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).

[Gonzales/Woods]

Sharpening with the Laplacian

- Images can be sharpened using the Laplacian:
- Given an input image $f(x,y)$, we obtained a sharpened image $g(x,y)$ by:

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

where c is a constant with

$c = -1$ for:

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

and $c = -1$ for:

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

Sharpening with the Laplacian

- Images can be sharpened using the Laplacian:
- Given an input image $f(x, y)$, we obtained a sharpened image $g(x, y)$ by:

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

where c is a constant with

c = -1 for:

0	1	0
1	−4	1
0	1	0

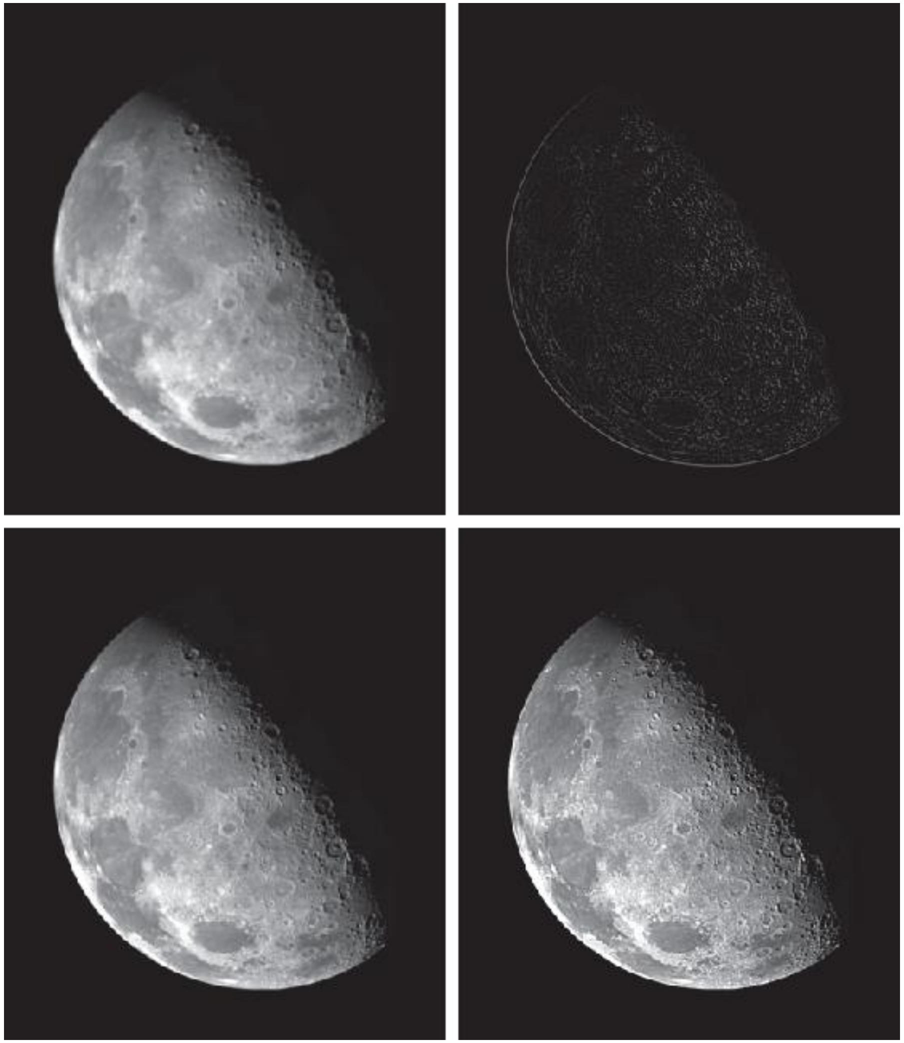
1	1	1
1	−8	1
1	1	1

and c = 1 for:

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

Sharpening with the Laplacian



a b
c d

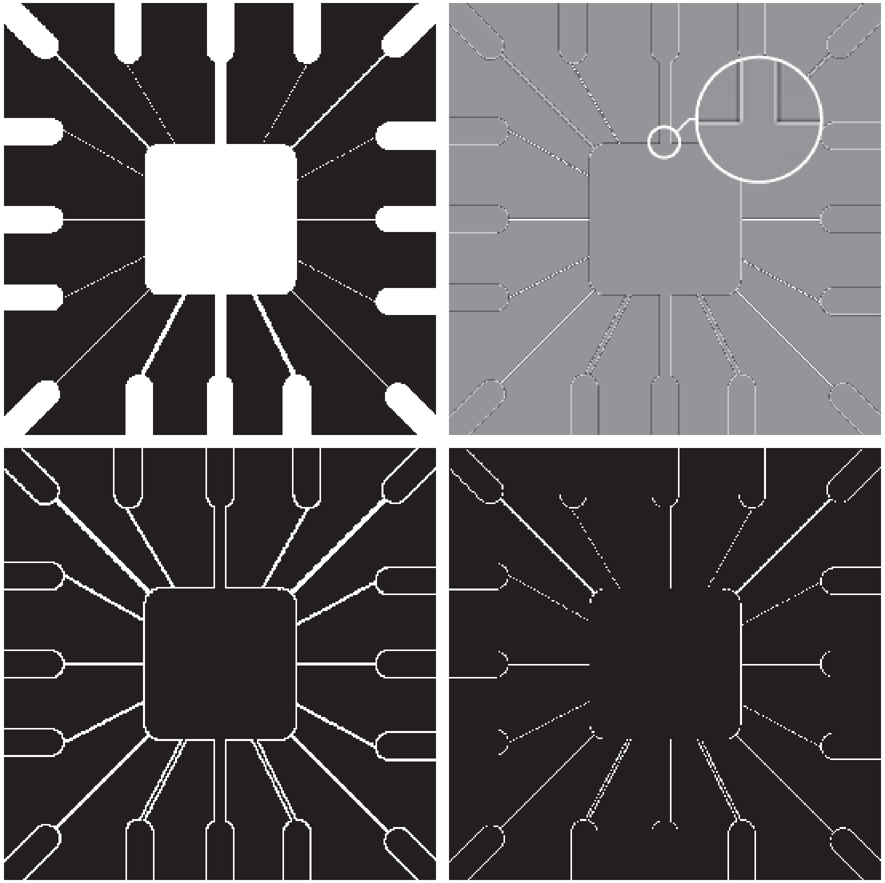
FIGURE 3.52
(a) Blurred image of the North Pole of the moon.
(b) Laplacian image obtained using the kernel in Fig. 3.51(a).
(c) Image sharpened using Eq. (3-63) with $c = -1$.
(d) Image sharpened using the same procedure, but with the kernel in Fig. 3.51(b).
(Original image courtesy of NASA.)

[Gonzales/Woods]

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Line Detection

- Lines can be also detected with the Laplacian operator (Example 10.2):



a b
c d

FIGURE 10.5 (a) Original image. (b) Laplacian image; the magnified section shows the positive/negative double-line effect characteristic of the Laplacian. (c) Absolute value of the Laplacian. (d) Positive values of the Laplacian.

[Gonzales/Woods]

Summary

- Edges can be found with derivatives
- Derivative operators use finite differences
- In 2D, the gradient is used to determine edges. It is a vector of partial derivatives. The gradient magnitude shows the edge strength
- Partial derivatives can be computed with kernels such as the Sobel operator
- A 2nd order derivative operator is the Laplacian
- The Laplacian is useful for image sharpening or line detection

Outlook

What else is out there?

- The Laplacian of Gaussian filter
- Canny Edge Detector
- Filtering in the frequency domain (see next lecture)

Literature

- Gonzales/Woods:
  - chapter 3.6 and
  - chapter 10.2
- Seliski
  - chapter 3.2.2
  - chapter 3.2.3

Outline

- Part 1: Edge models and derivatives
- Part 2: Derivatives and edge detection in 1D
- Part 3: Partial derivatives, gradient, and edge detection in 2D
- Part 4: The effect of noise, Sobel & Prewitt filter
- Part 5: 2^{nd} order derivatives and the Laplacian filter
- Part 6: Using the Laplacian for sharpening and line detection