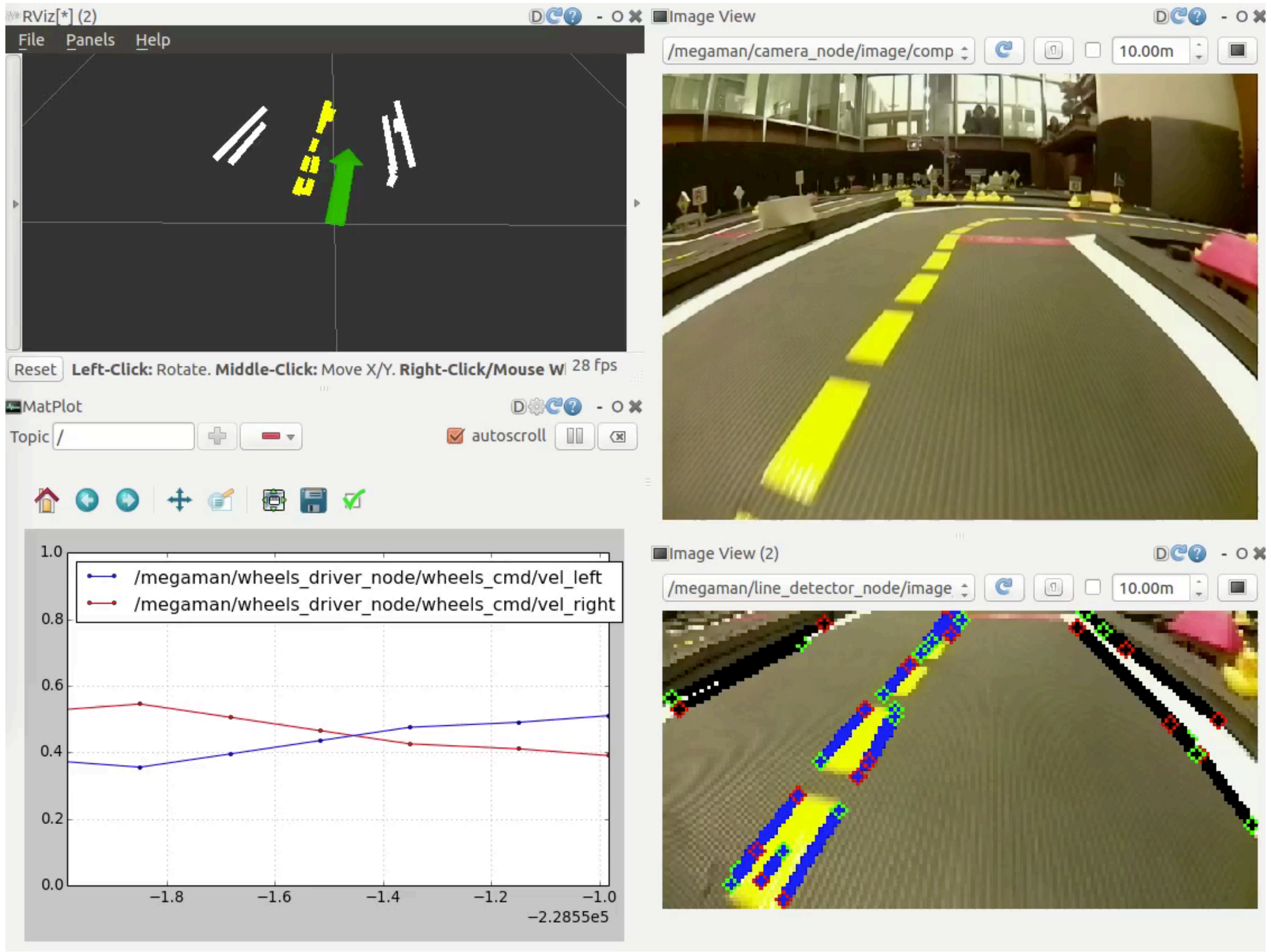
Computer vision: image gradients

Why edge detection?

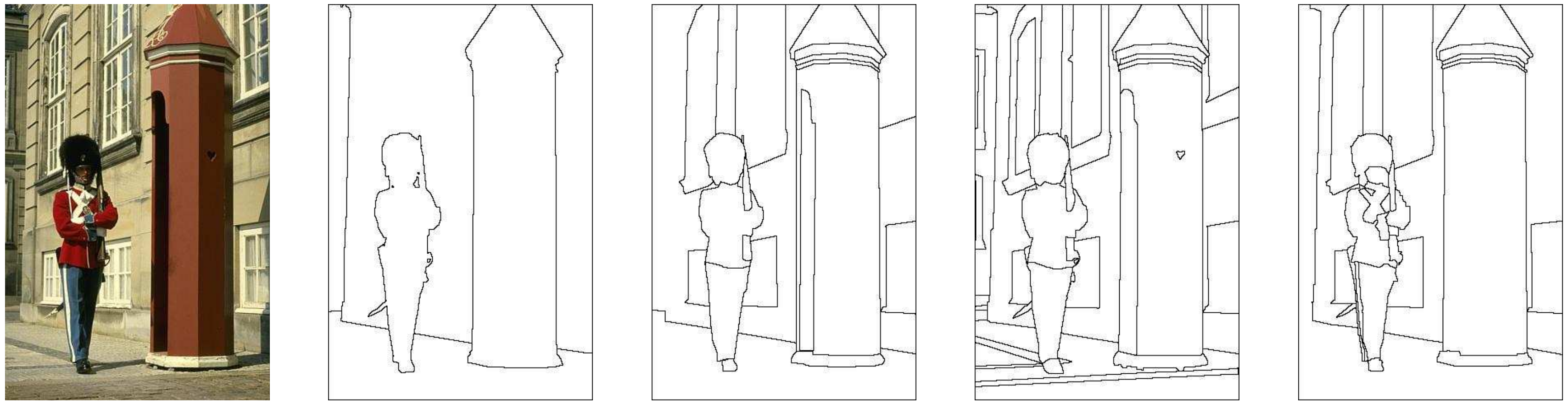
- **Goal**: Identify sudden changes (discontinuities) in an image
  - Most semantic and shape information is encoded in edges
  - More compact than raw intensities



Why edge detection?

- **Goal**: Identify sudden changes (discontinuities) in an image
  - Most semantic and shape information is encoded in edges
  - More compact than raw intensities
- **Edges correspond** to valid decompositions:
  - Surface normal discontinuity
  - Depth discontinuity
  - Different materials

What defines an edge is very subjective

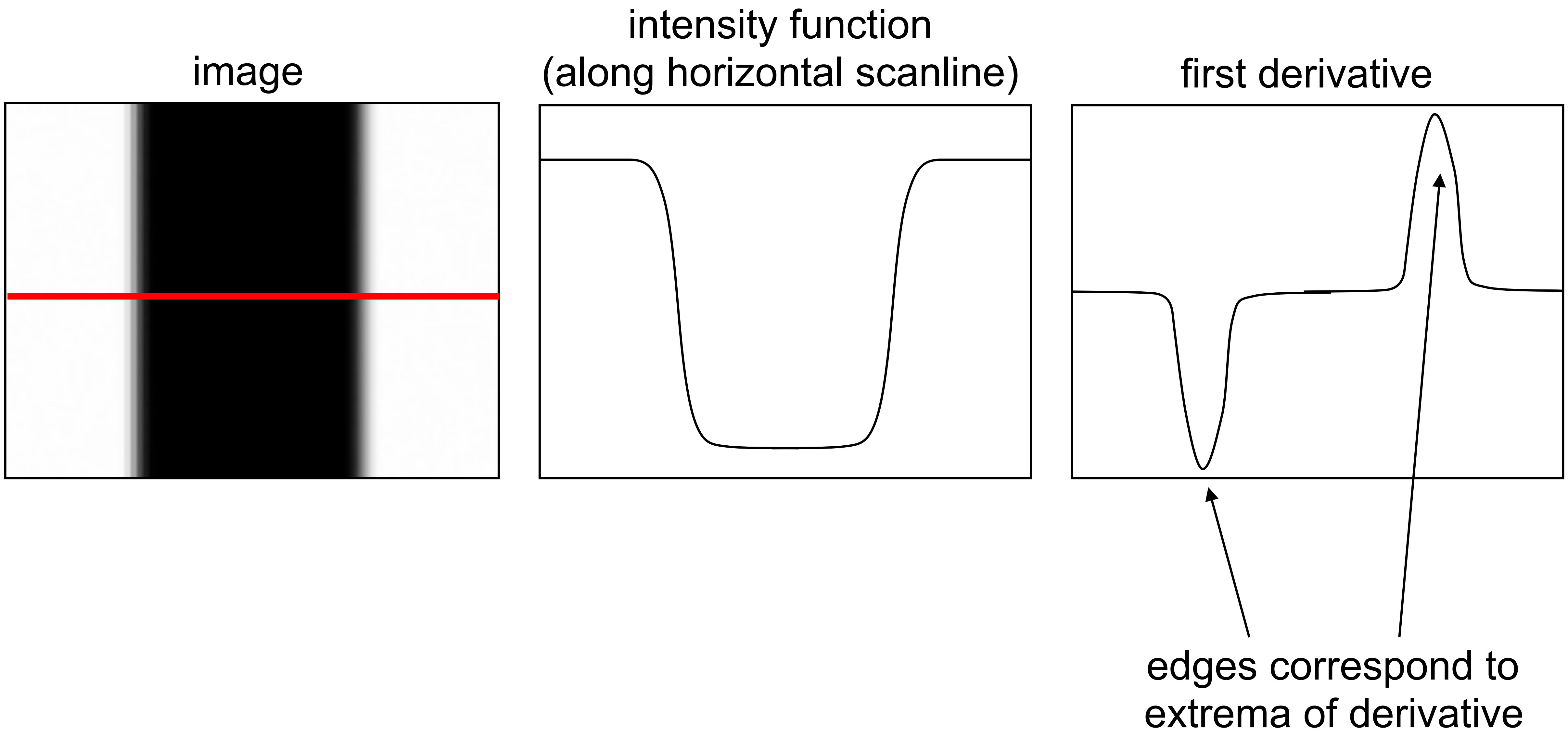


Why edge detection?

- **Goal**: Identify sudden changes (discontinuities) in an image
  - Most semantic and shape information is encoded in edges
  - More compact than raw intensities
- **Edges are caused** by several factors:
  - Changes in depth or surface normal
  - Surface color discontinuity
  - Illumination discontinuity

Change in image intensity

- **An edge in an image is a place of rapid change in the intensity function**



Edge detection: Partial derivatives

- **For a 2D function $f(x, y)$, the partial derivative is**

$$\frac{\partial f(x,y)}{\partial x} = \lim_{\epsilon \to 0} \frac{f(x+\epsilon, y) - f(x,y)}{\epsilon}$$

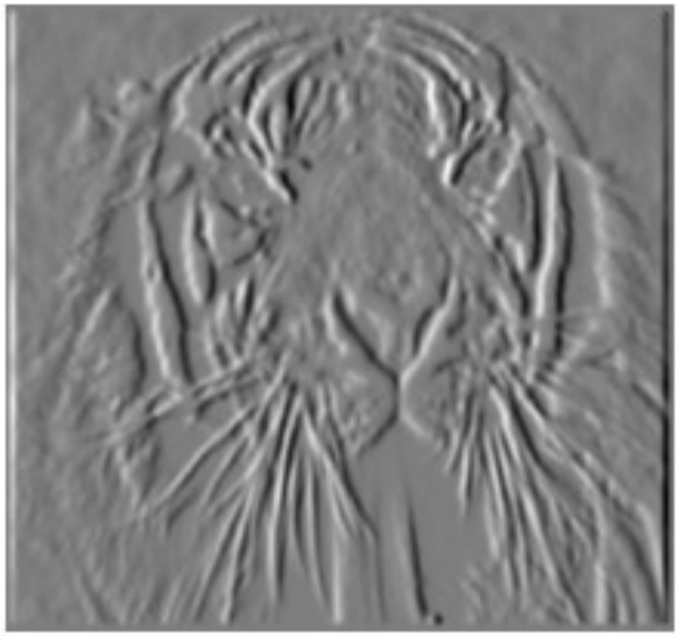
- **For discrete data, we can approximate this using finite differences**

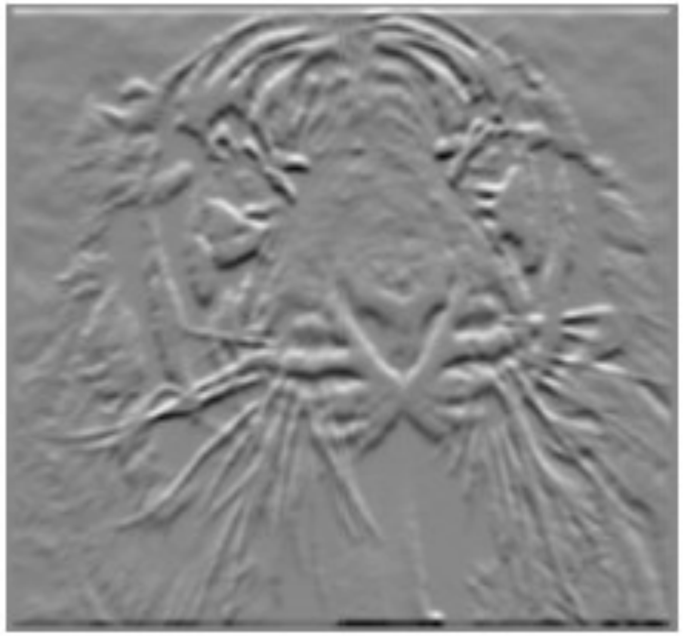
$$\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x+1, y) - f(x,y)}{1}$$

- **Finite difference filters are easy to implement**

Edge detection: Finite difference filters

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





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

Edge detection: Finite difference filters

Finite Difference Approximation

$$\frac{\partial}{\partial n_x} X[n_x, n_y] \quad \propto X[n_x + 1, n_y] - X[n_x - 1, n_y]$$

$$X * [1 \quad 0 \quad -1]$$

Derivative is a linear spatially invariant operation: Convolution

$$X * \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Smoothed in y direction
"Sobel" Operator

$$X * \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Y Derivative

Edge detection: Finite difference filters

Sobel operator

$$X * \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

$$X * \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

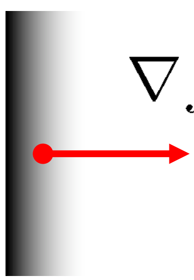
Derivatives have been scaled so that gray (0.5) corresponds to 0. Bright to positive derivative values, dark to negative.

Image gradients

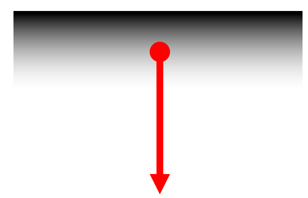
- The gradient of an image: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$
- Gradient points in direction of most rapid increase in intensity
    - How is this direction related to the direction of the edge?

Image gradients

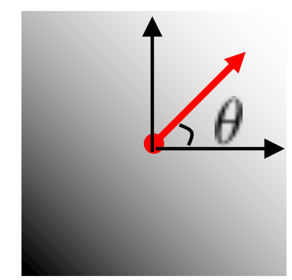
- The gradient of an image: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$
- Gradient points in direction of most rapid increase in intensity
  - How is this direction related to the direction of the edge?



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



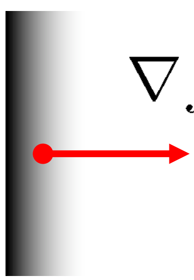
$$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$$



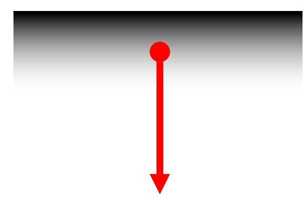
$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

Image gradients

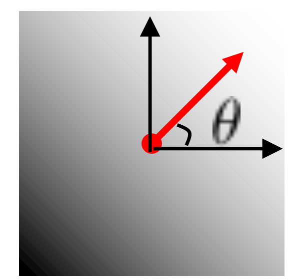
- The gradient of an image: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$
- Gradient points in direction of most rapid increase in intensity
    - How is this direction related to the direction of the edge?



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



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



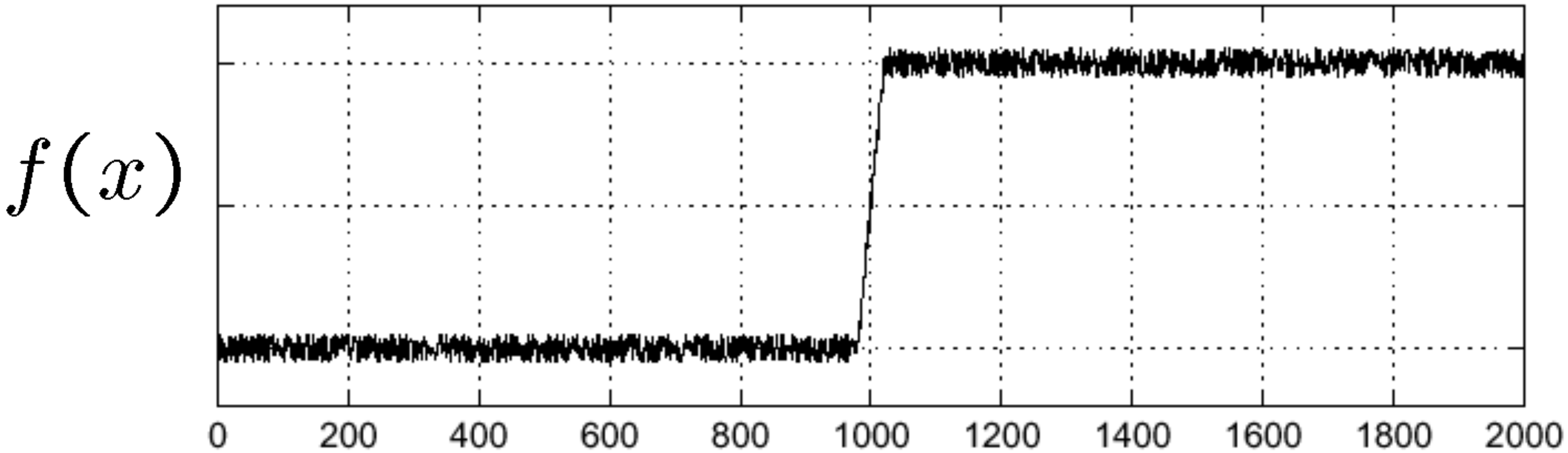
$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

- Gradient direction given by $\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$
- Edge strength given by gradient magnitude

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

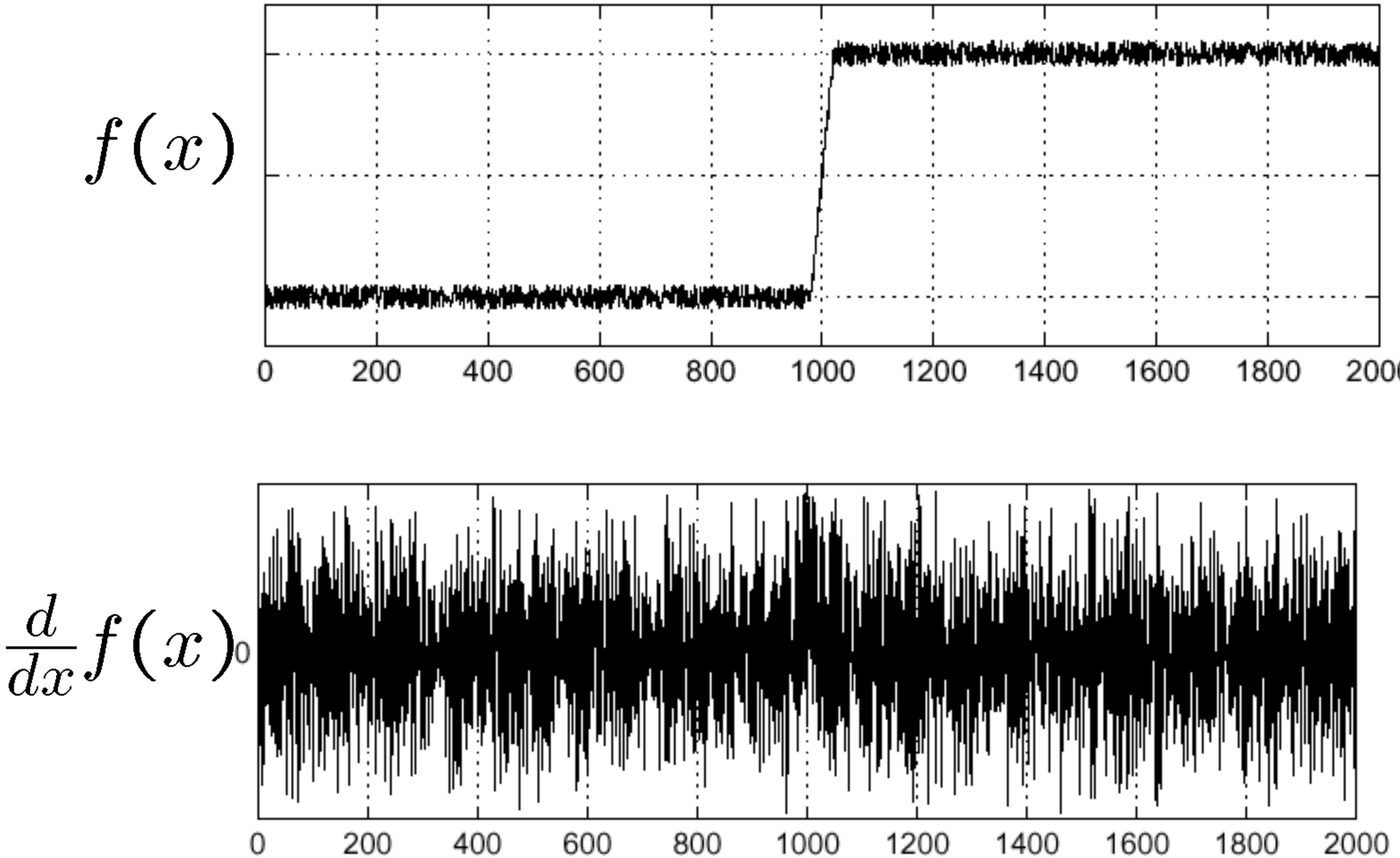
Effects of noise

Consider a 1D image



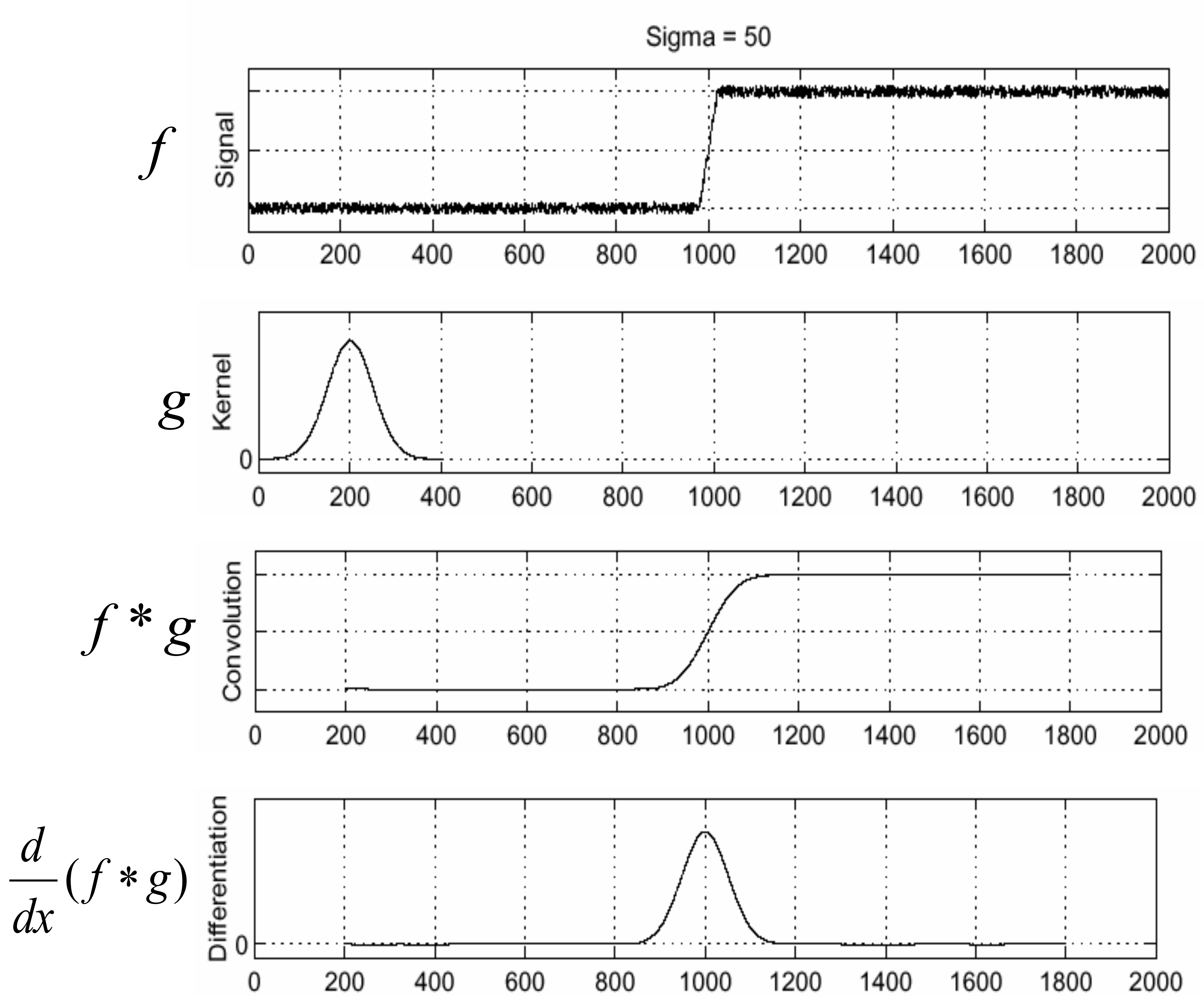
Effects of noise

Consider a 1D image



Where is the edge?

Solution: Apply a smoothing filter first



Derivative theorem of convolution

- Differentiation is shift-invariant and linear ⟶ there exists a corresponding kernel

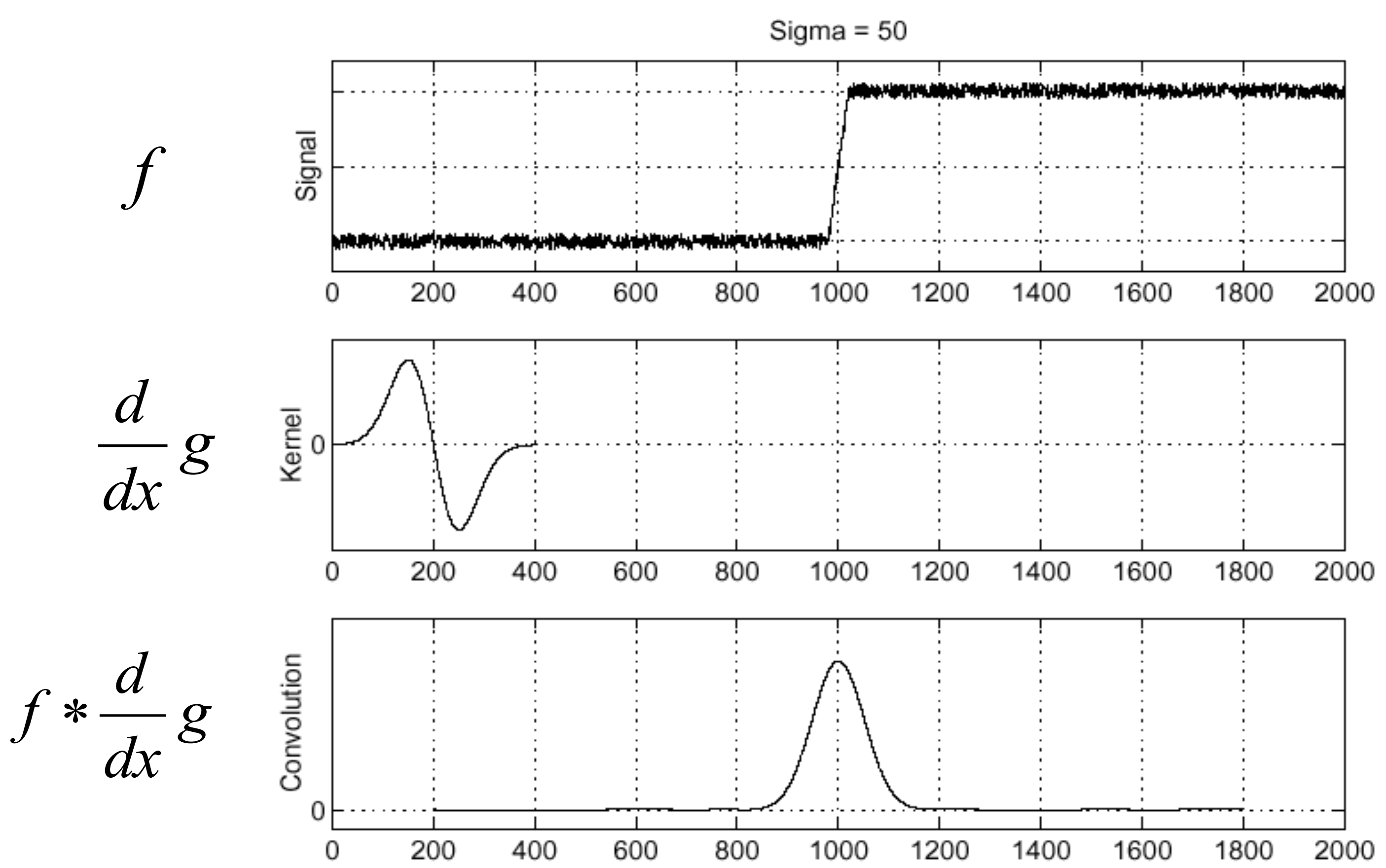
Derivative theorem of convolution

- Differentiation is shift-invariant and linear ⟶ there exists a corresponding kernel
- Differentiation is associative:

$$\frac{d}{dx}(g * f) = \left(\frac{dg}{dx}\right) * f$$

Derivative theorem of convolution

- Differentiation is shift-invariant and linear ⟶ there exists a corresponding kernel
- Differentiation is associative:
- This saves one operation $$\frac{d}{dx}(g * f) = \left(\frac{dg}{dx}\right) * f$$

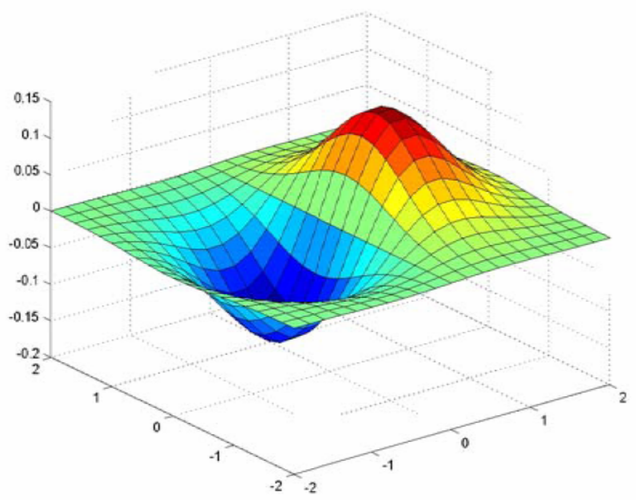
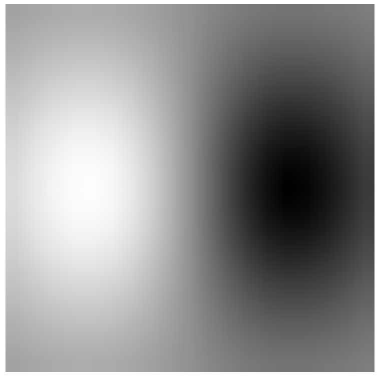


Derivative of Gaussian filters

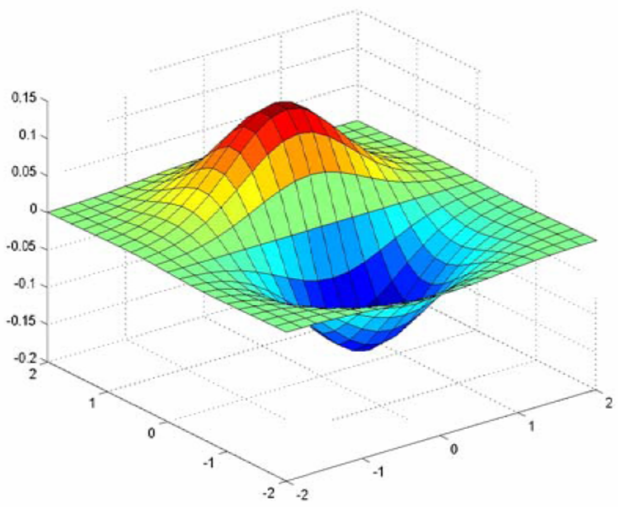
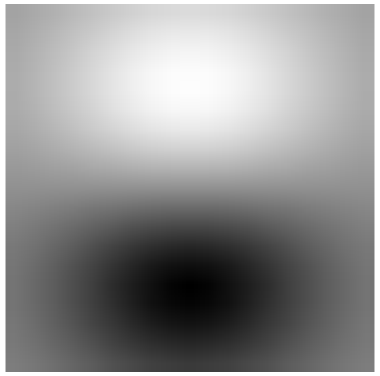
- In practice, we smooth using Gaussian filters (for reasons mentioned earlier)
  - Remember: Separability

$$I_x = \partial_x * (G_\sigma * X) = (\partial_x * G_\sigma) * X = G_{x:\sigma} * X$$

$$G_x = \frac{-x}{2\pi\sigma^4} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \qquad G_y = \frac{-y}{2\pi\sigma^4} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

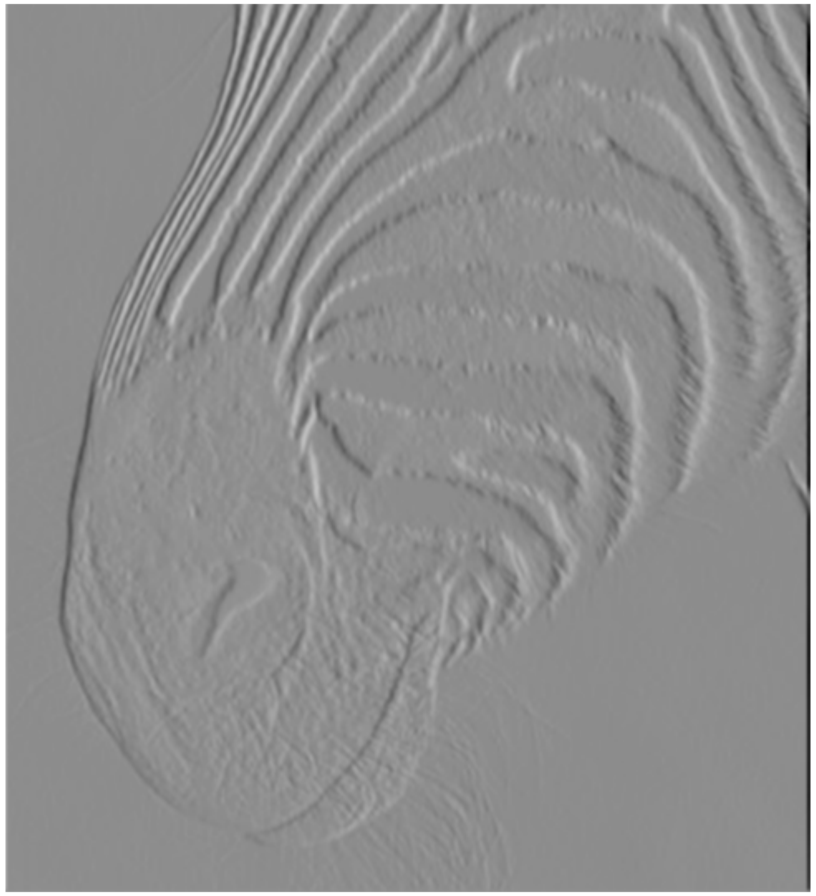



x-direction

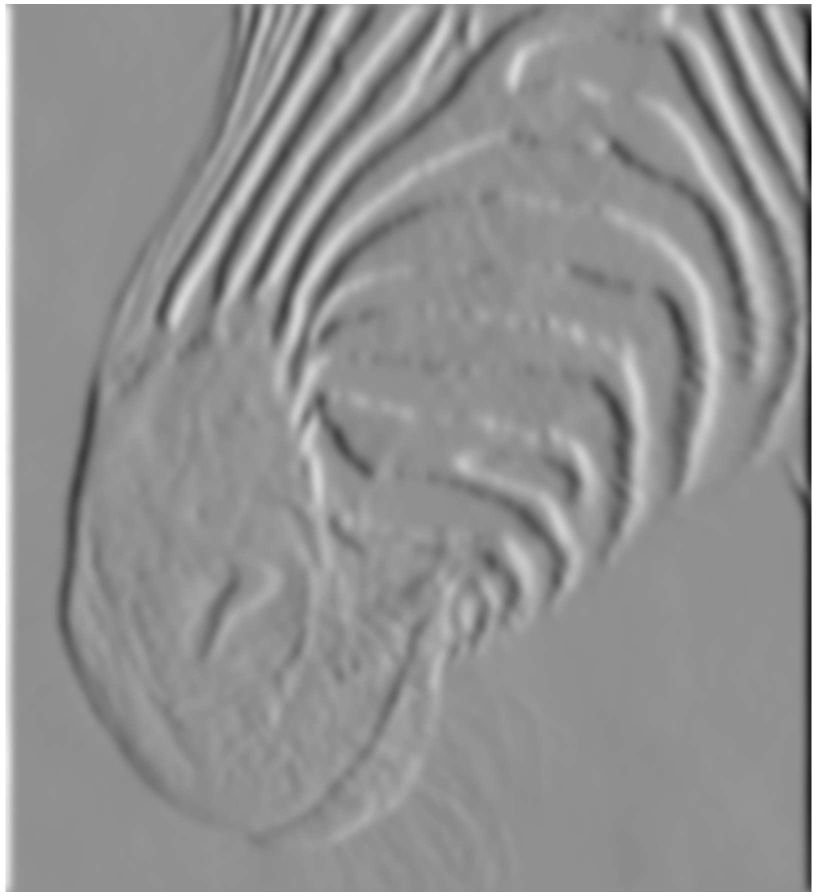



y-direction

Scale of Gaussian derivative filters



1 pixel



3 pixels

7 pixels

Smoothed derivative removes noise, but blurs edges.
Also finds edges at different "scales"