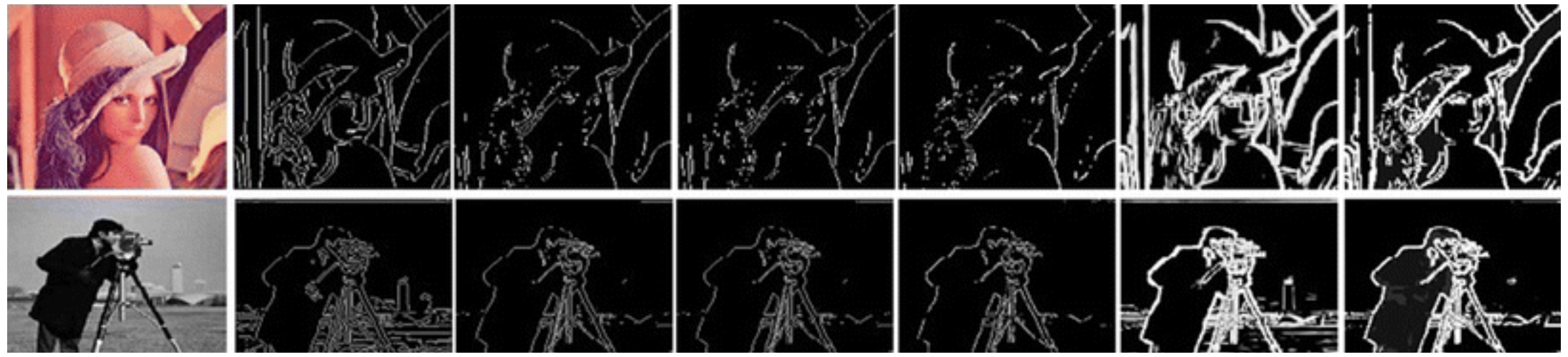
THE UNIVERSITY OF BRITISH COLUMBIA

CPSC 425: Computer Vision



Lecture 11: Edge Detection (cont.)

(unless otherwise stated slides are taken or adopted from **Bob Woodham, Jim Little** and **Fred Tung**)

Menu for Today (**October 2, 2020**)

Topics:

— Edge **Detection**
— **Marr / Hildreth** and **Canny** Edges
— Image **Boundaries**

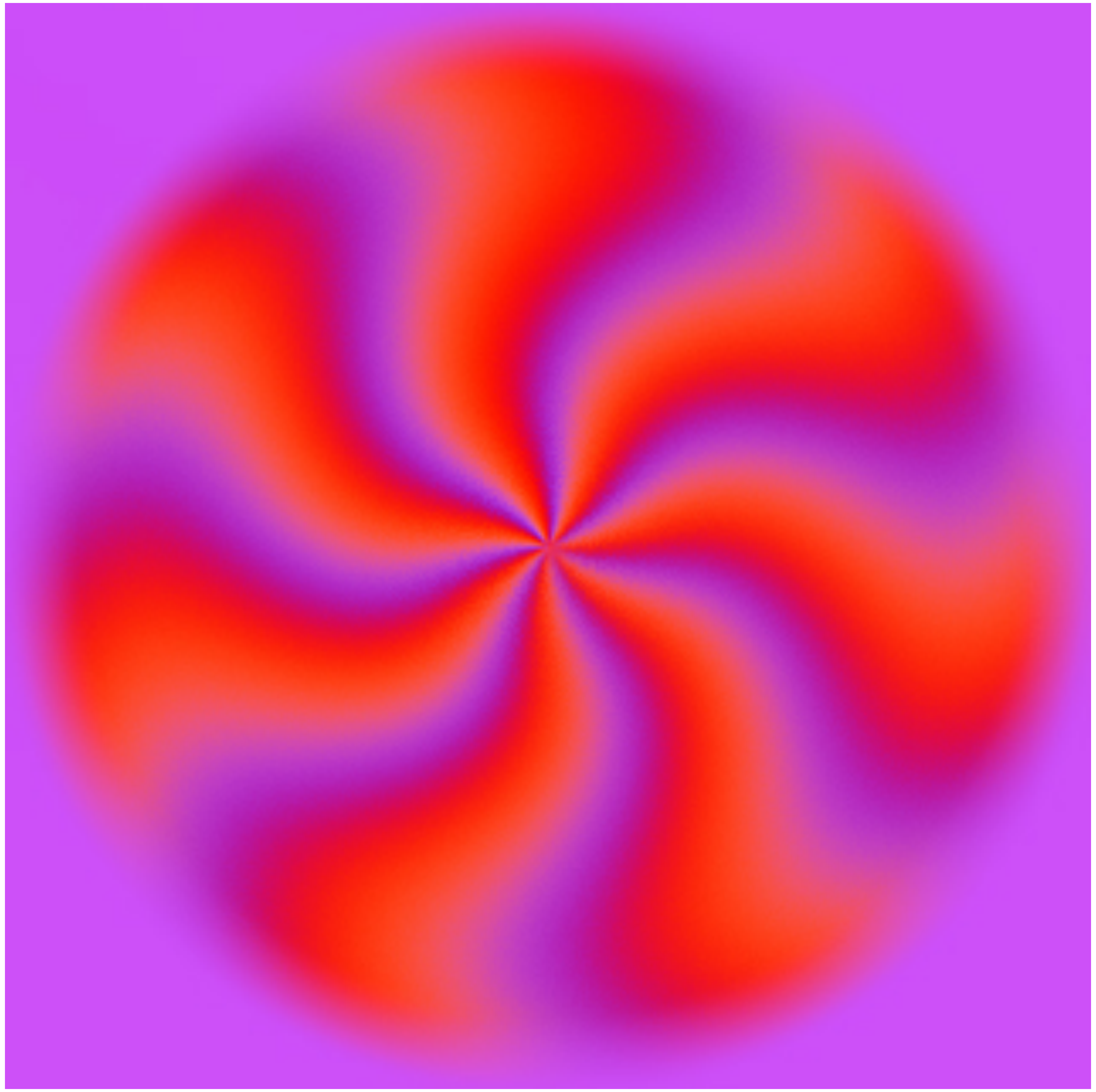
Redings:

— **Today's** Lecture: Forsyth & Ponce (2nd ed.) 5.1 - 5.2
— **Next** Lecture: Forsyth & Ponce (2nd ed.) 5.3.0 - 5.3.1

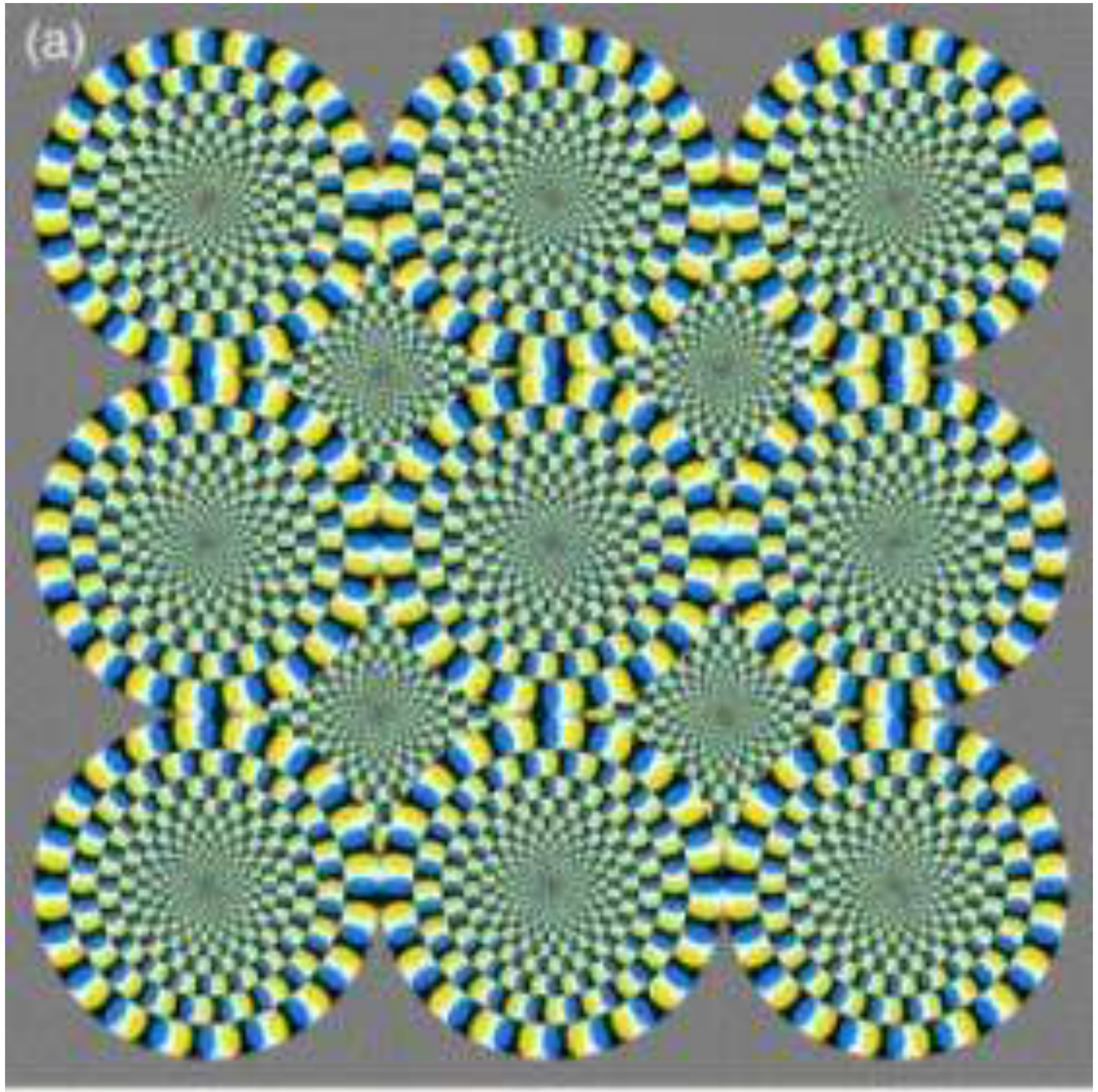
Reminders:

— **Assignment 2**: Scaled Representations, Face Detection and Image Blending

Today’s “**fun**” Example #1: Motion Illusion



Today's "**fun**" Example #1: Rotating Snakes Illusion



Lecture 10: Re-cap

A (**discrete**) approximation is

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

— "First forward difference"

— Can be implemented as a **convolution**

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

Lecture 10: Re-cap

A (**discrete**) approximation is

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

"**forward** difference" implemented as

correlation

-1	1

from **left**

convolution

1	-1

"**backward** difference" implemented as

correlation

-1	1

from **right**

convolution

1	-1

Lecture 10: Re-cap

Use the "first forward difference" to compute the image derivatives in X and Y directions.

(Compute two arrays, one of $\frac{\partial f}{\partial x}$ values and one of $\frac{\partial f}{\partial y}$ values.)

1	1	0.6	0.3	0	0
1	1	0.6	0.3	0	0
0	0	0	0	0	0
0	0	0	0	0	0

A Sort **Exercise**: Derivative in Y Direction

Use the “first forward difference" to compute the image derivatives in X and Y directions.

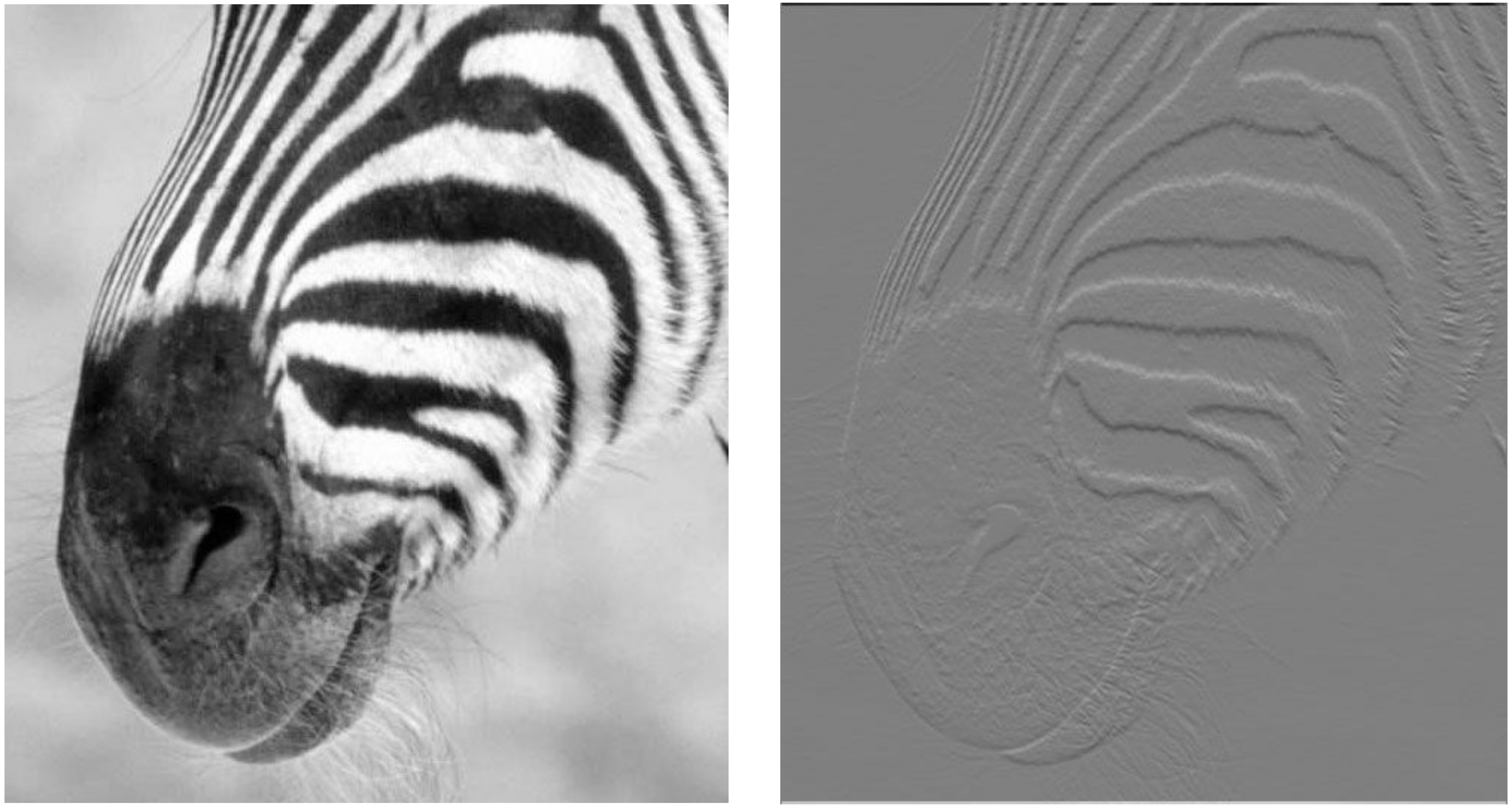
(Compute two arrays, one of $\frac{\partial f}{\partial x}$ values and one of $\frac{\partial f}{\partial y}$ values.)

1	1	0.6	0.3	0	0
1	1	0.6	0.3	0	0
0	0	0	0	0	0
0	0	0	0	0	0

1					
0	0	0	0	0	0

Lecture 10: Re-cap

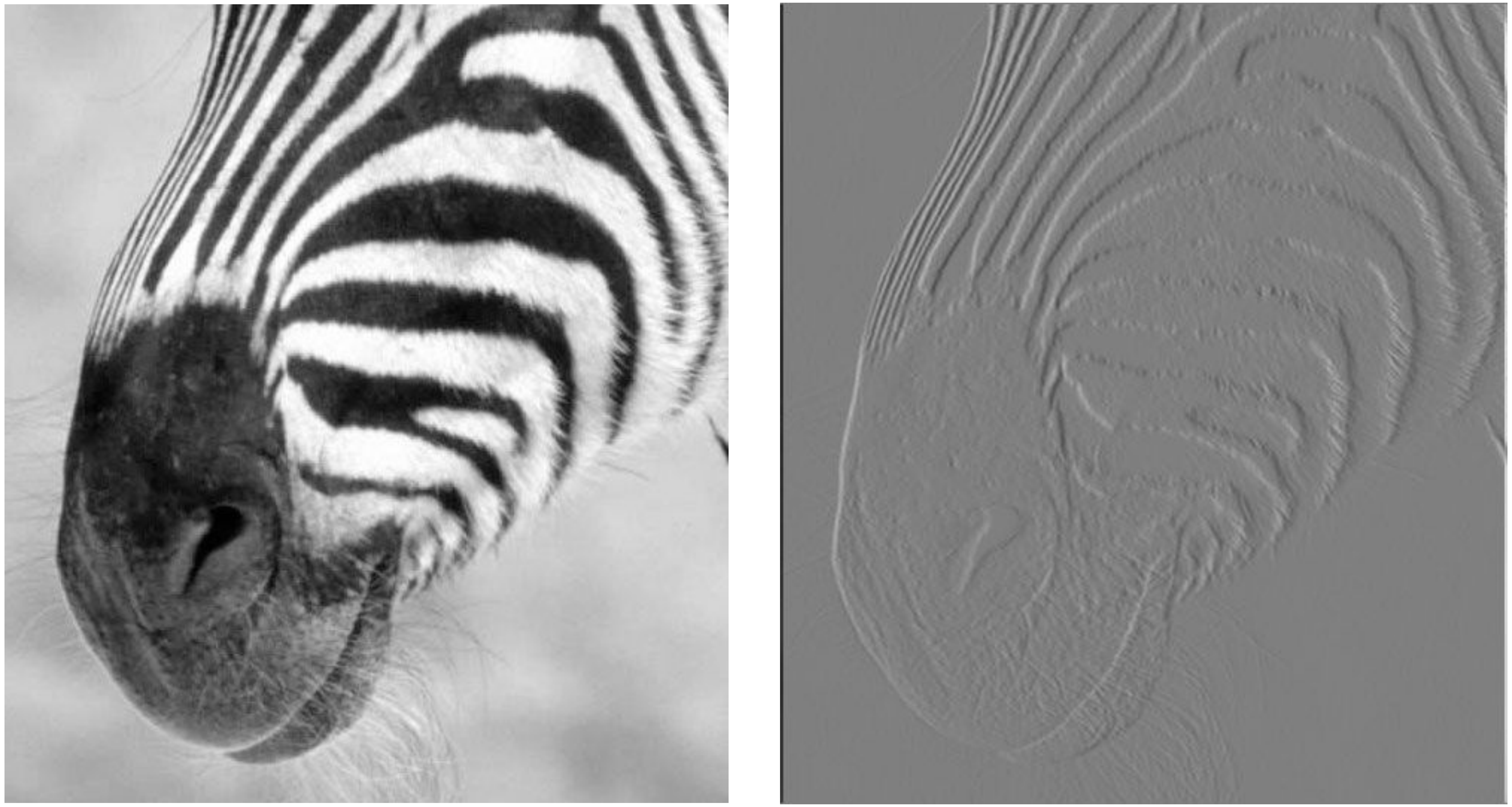
Derivative in Y (i.e., vertical) direction



Forsyth & Ponce (1st ed.) Figure 7.4 (top left & top middle)

Lecture 10: Re-cap

Derivative in X (i.e., horizontal) direction



Forsyth & Ponce (1st ed.) Figure 7.4 (top left & top right)

Estimating **Derivatives**

-1	1

Question: Why, in general, should the weights of a filter used for differentiation sum to 0?

Estimating **Derivatives**

-1	1

Question: Why, in general, should the weights of a filter used for differentiation sum to 0?

Answer: Think of a constant image, $I(X, Y) = k$. The derivative is 0. Therefore, the weights of any filter used for differentiation need to sum to 0.

Estimating **Derivatives**

-1	1

Question: Why, in general, should the weights of a filter used for differentiation sum to 0?

Answer: Think of a constant image, $I(X,Y) = k$. The derivative is 0. Therefore, the weights of any filter used for differentiation need to sum to 0.

$$\sum_{i=1}^{N} f_i \cdot k = k \sum_{i=1}^{N} f_i = 0 \implies \sum_{i=1}^{N} f_i = 0$$

Edge Detection

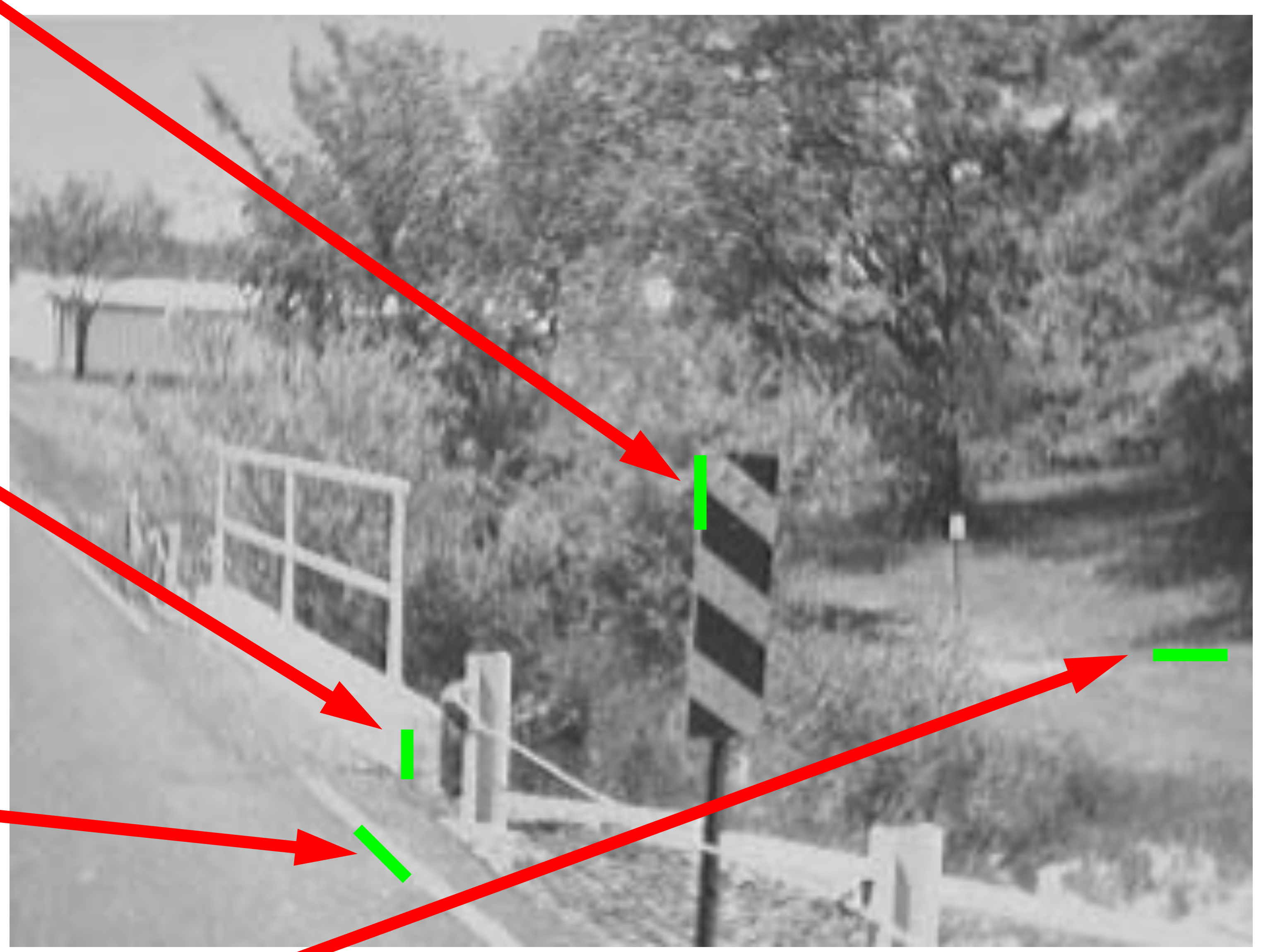
Goal: Identify sudden changes in image intensity

This is where most shape information is encoded

Example: artist's line drawing (but artist also is using object-level knowledge)

What Causes **Edges**?

- Depth discontinuity
- Surface orientation discontinuity
- Reflectance discontinuity (i.e., change in surface material properties)
- Illumination discontinuity (e.g., shadow)



Slide Credit: Christopher Rasmussen

Smoothing and Differentiation

Edge: a location with high gradient (derivative)

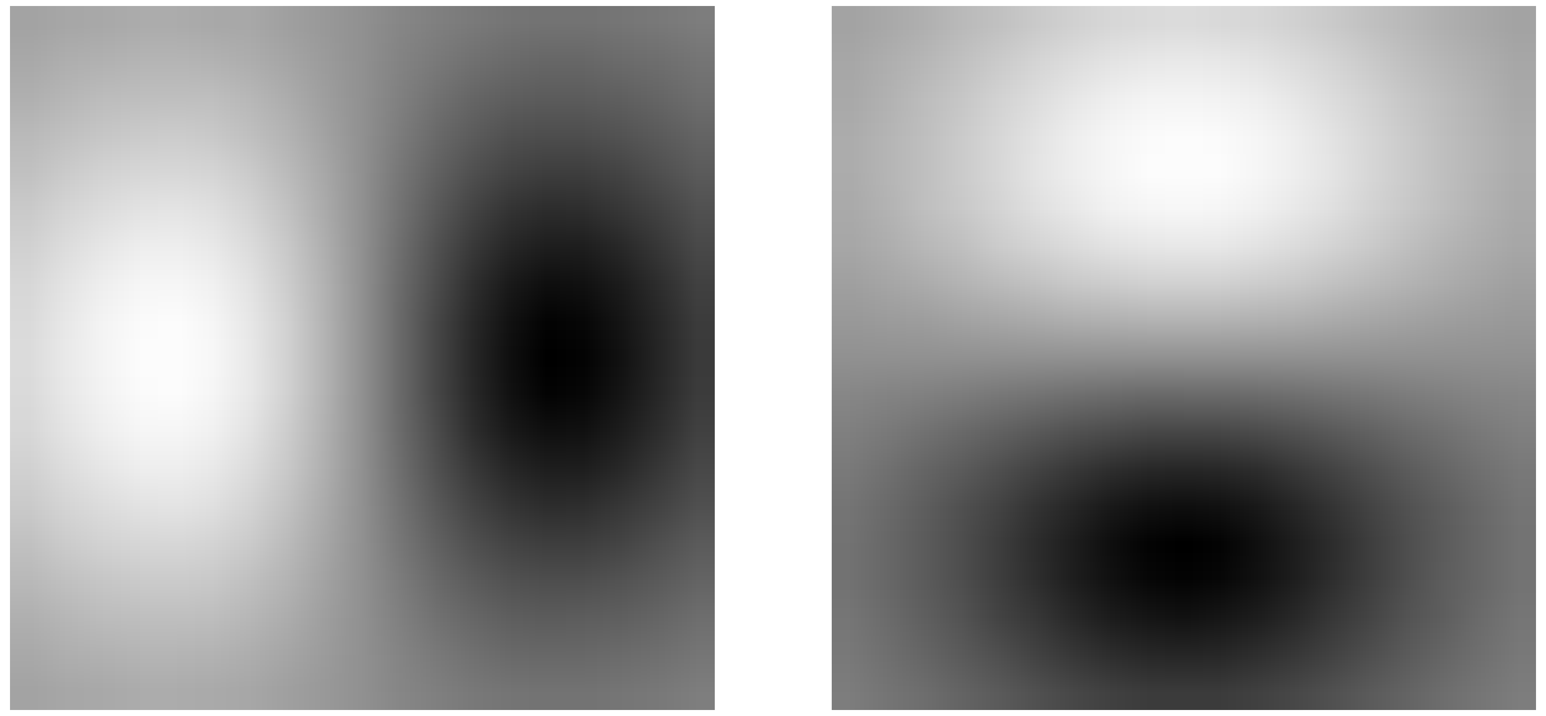
Need smoothing to reduce noise prior to taking derivative

Need two derivatives, in x and y direction

We can use **derivative of Gaussian** filters
— because differentiation is convolution, and
— convolution is associative

Let $\otimes$ denote convolution

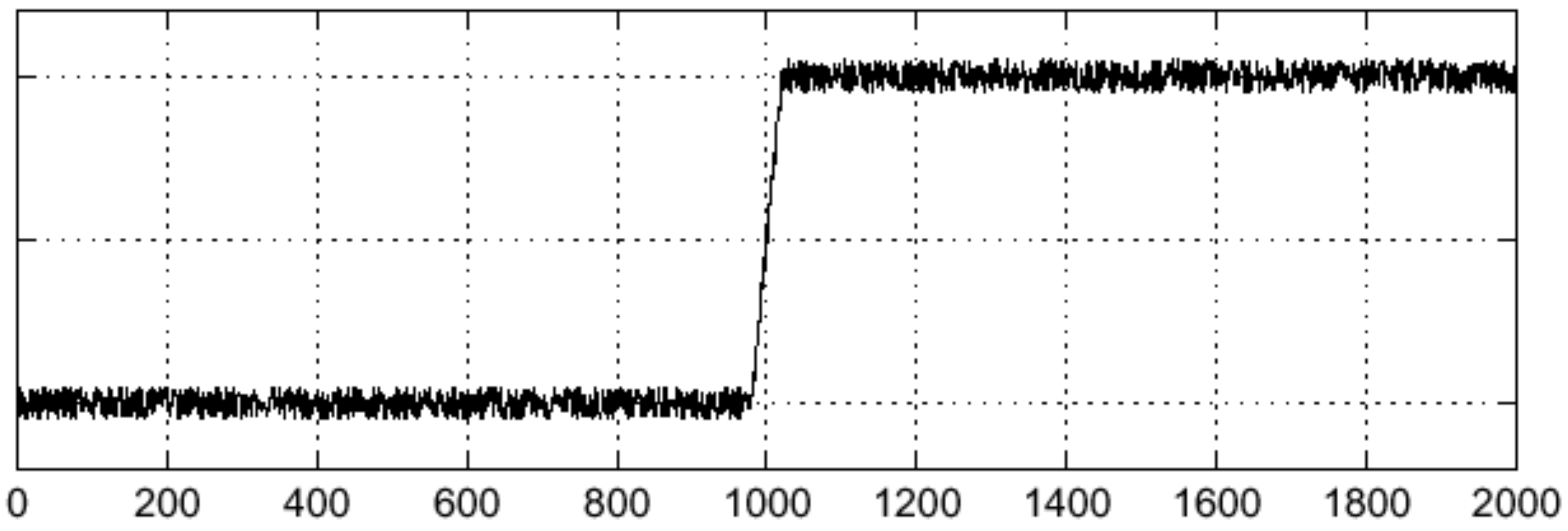
$$D \otimes (G \otimes I(X,Y)) = (D \otimes G) \otimes I(X,Y)$$



1D **Example**

Lets consider a row of pixels in an image:

$I(X, 245)$

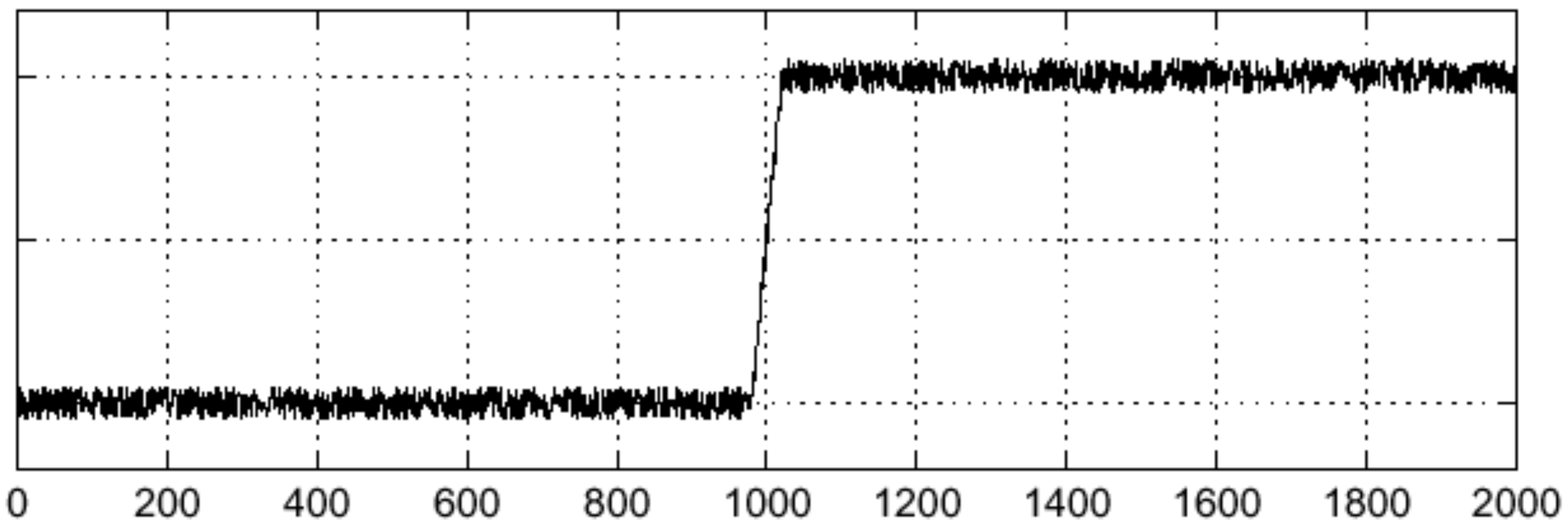


Where is the edge?

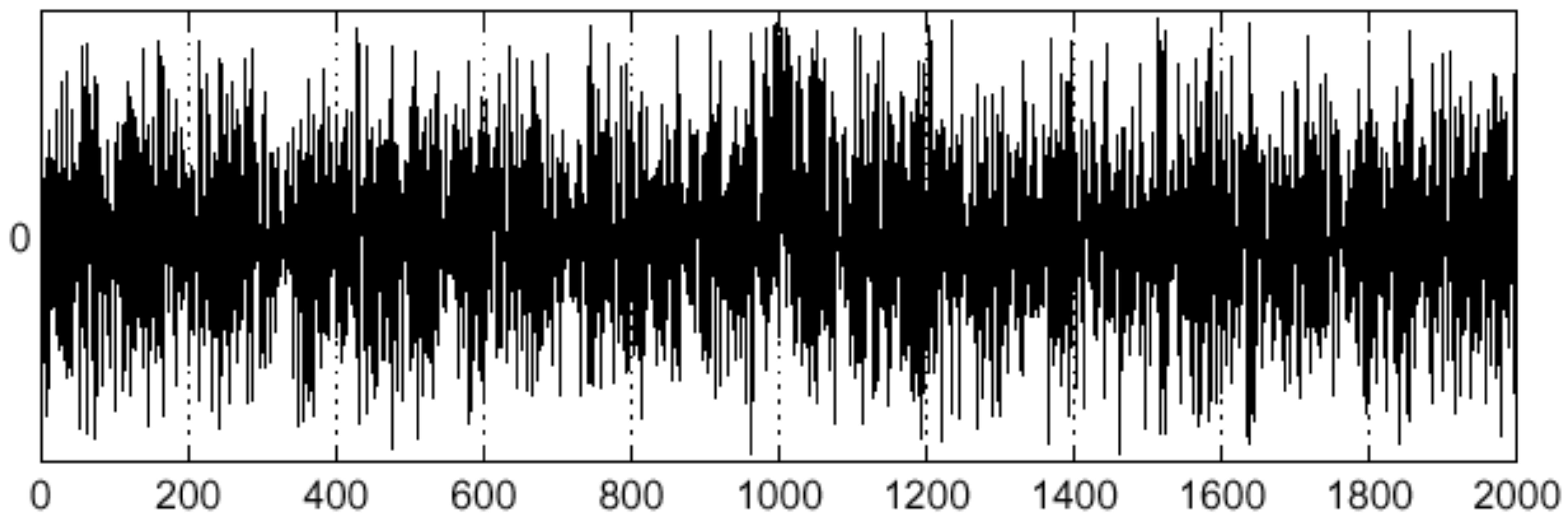
1D **Example**: Derivative

Lets consider a row of pixels in an image:

$$I(X, 245)$$



$$\frac{\partial I(X, 245)}{\partial x}$$



Where is the edge?

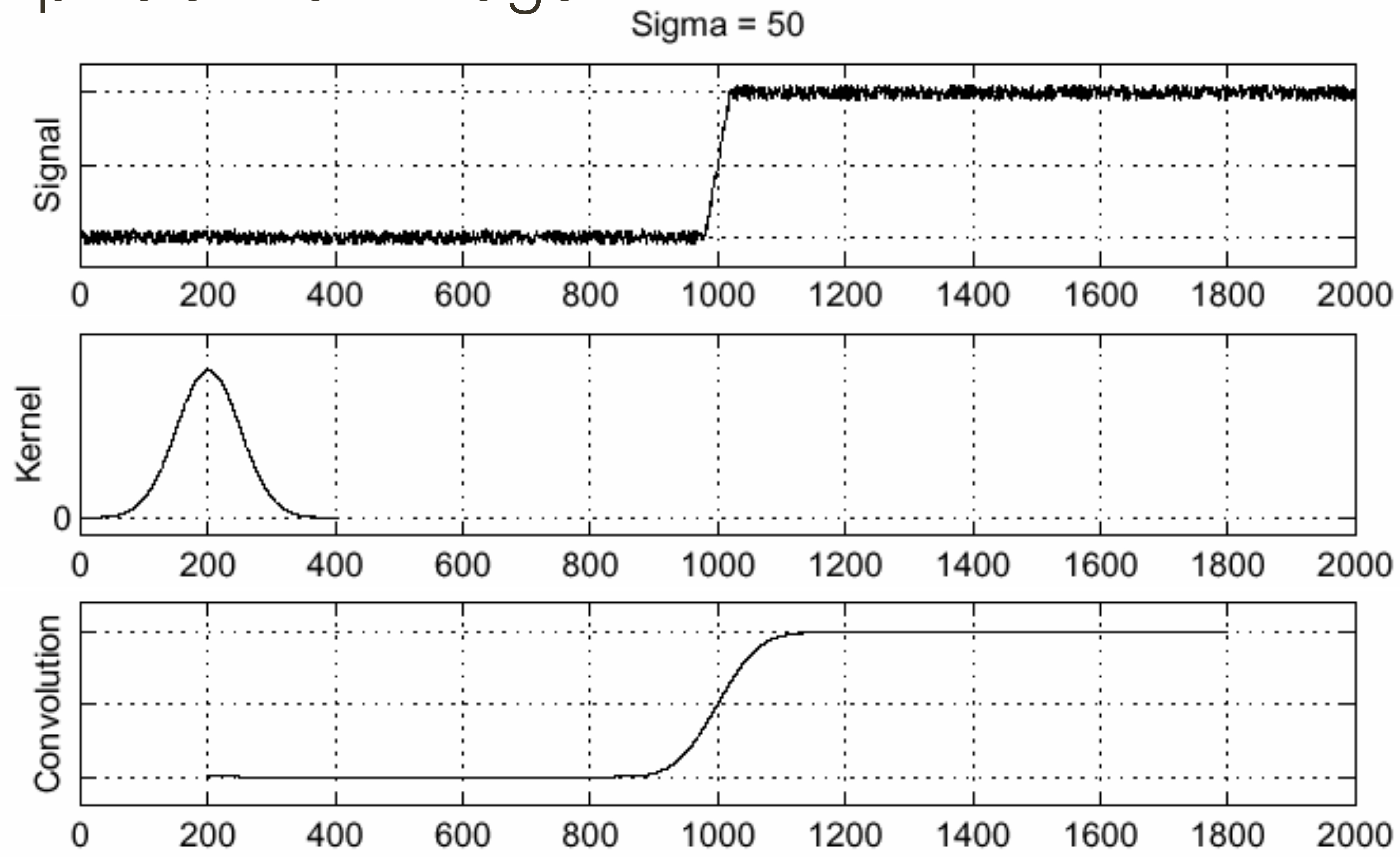
1D **Example**: Smoothing + Derivative

Lets consider a row of pixels in an image:

$I(X, 245)$

G

$G \otimes I(X, Y)$



1D **Example**: Smoothing + Derivative

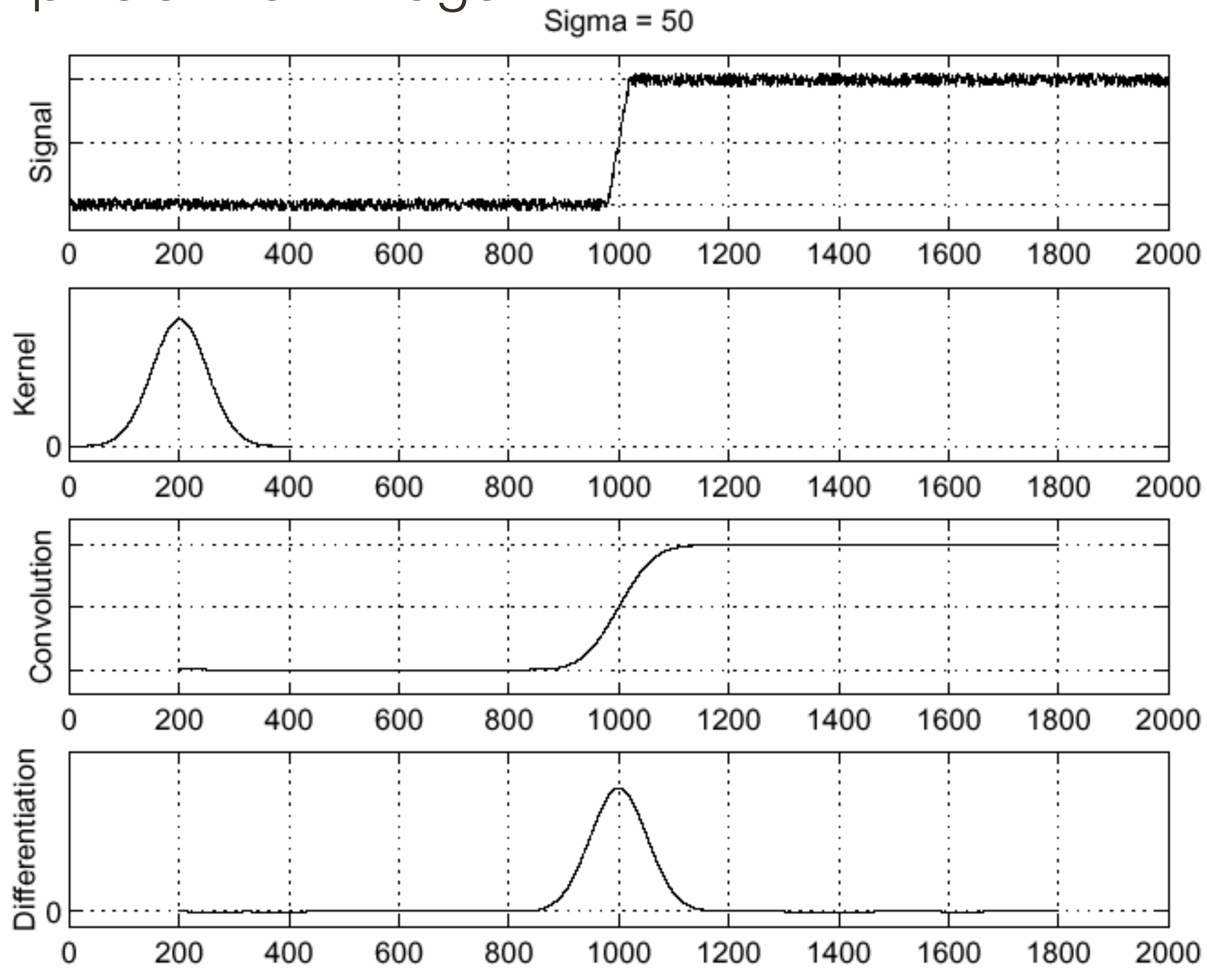
Lets consider a row of pixels in an image:

$$I(X, 245)$$

$$G$$

$$G \otimes I(X, Y)$$

$$\frac{\partial G \otimes I(X, Y)}{\partial x}$$



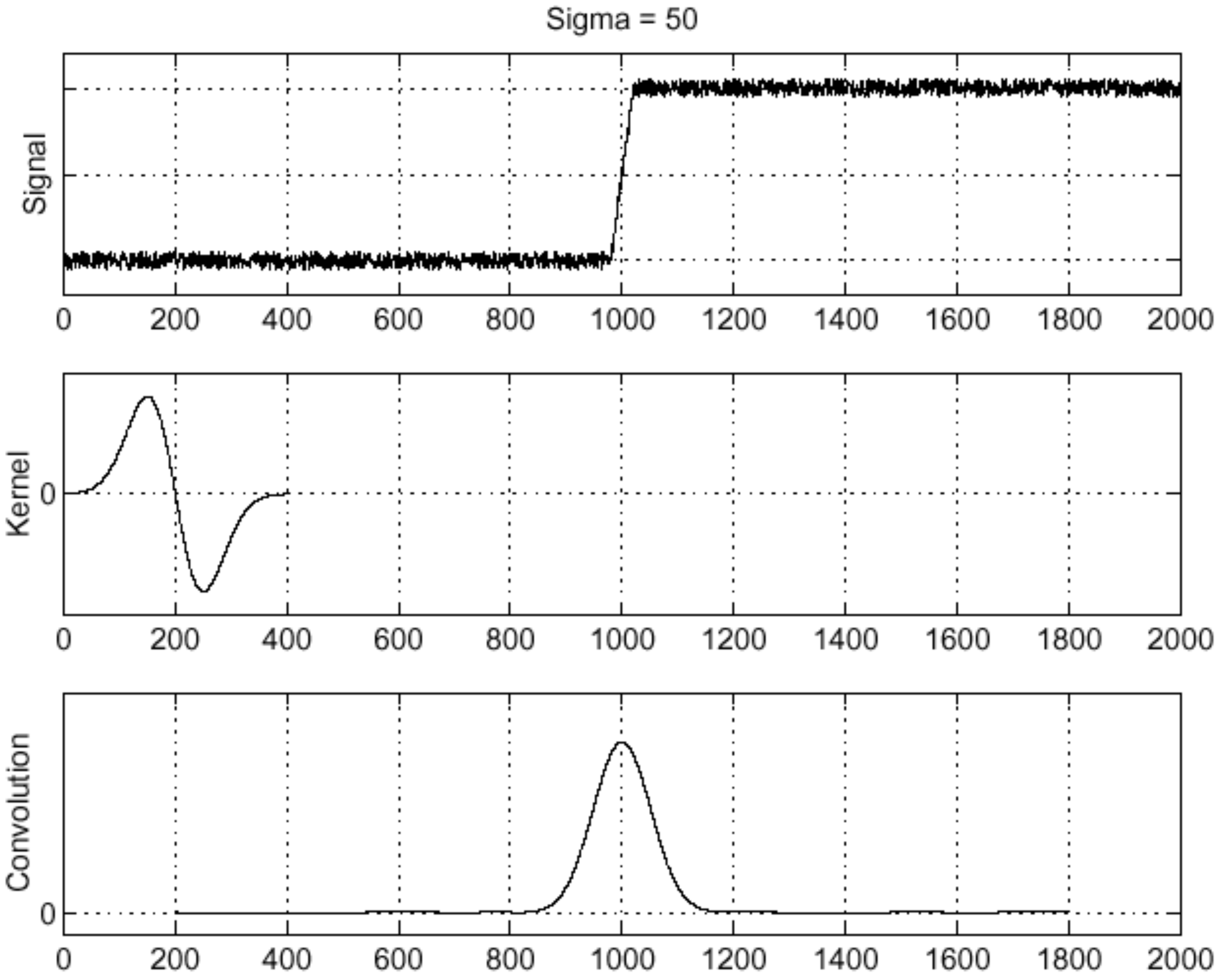
1D **Example**: Smoothing + Derivative (efficient)

Lets consider a row of pixels in an image:

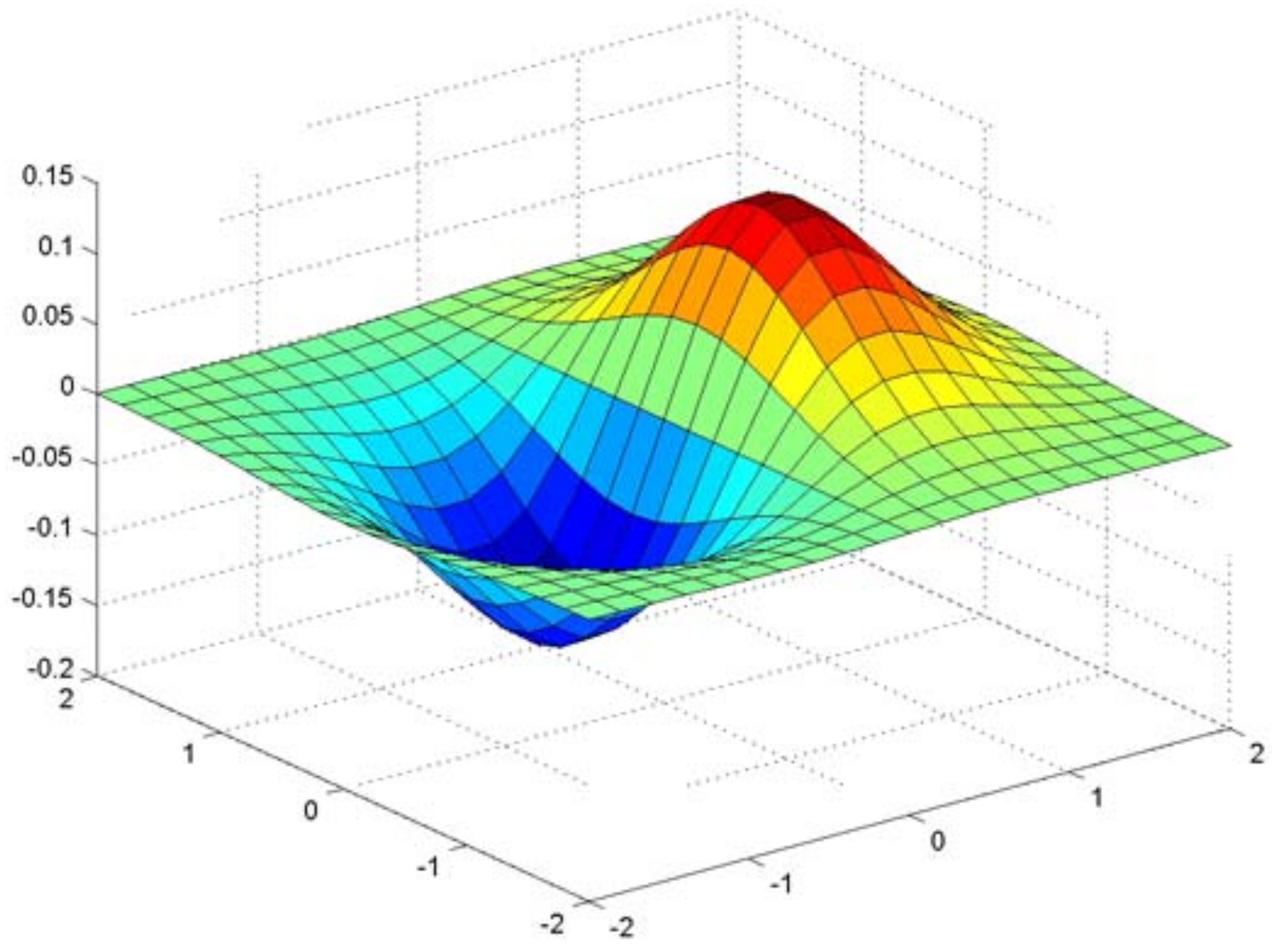
$$I(X, 245)$$

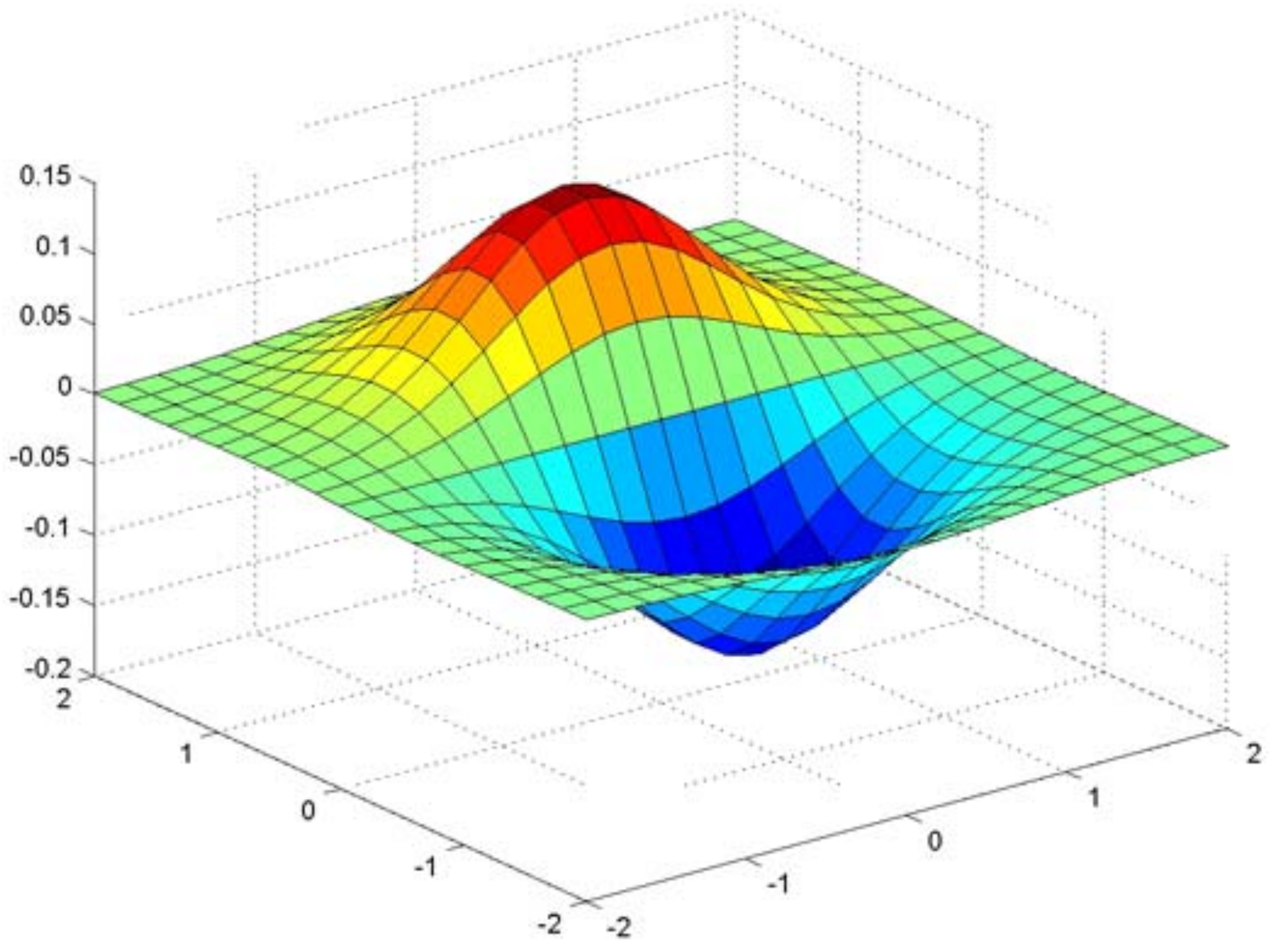
$$\frac{\partial G}{\partial x}$$

$$\frac{\partial G}{\partial x} \otimes I(X, Y)$$



Partial **Derivatives** of **Gaussian**





$$\frac{\partial}{\partial x} G_\sigma$$

$$\frac{\partial}{\partial y} G_\sigma$$

Slide Credit: Christopher Rasmussen

Gradient **Magnitude**

Let $I(X,Y)$ be a (digital) image

Let $I_x(X,Y)$ and $I_y(X,Y)$ be estimates of the partial derivatives in the x and y directions, respectively.

Call these estimates I_x and I_y (for short) The vector $[I_x, I_y]$ is the **gradient**

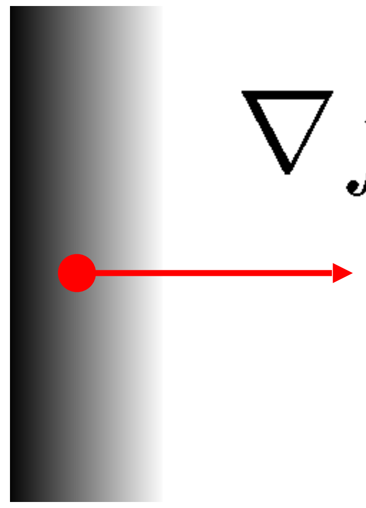
The scalar $\sqrt{I_x^2 + I_y^2}$ is the **gradient magnitude**

Image **Gradient**

The gradient of an image: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

Image **Gradient**

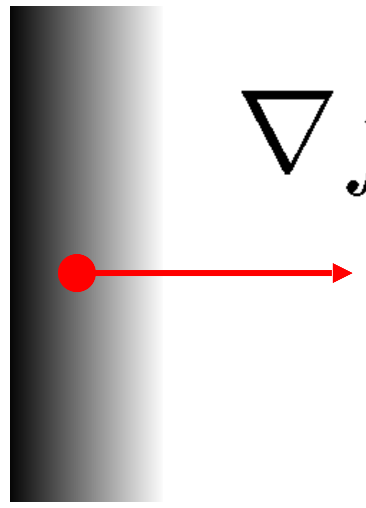
The gradient of an image: $$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$



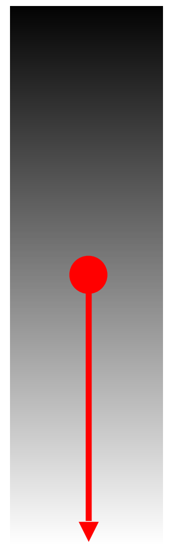
$$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$$

Image **Gradient**

The gradient of an image: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$



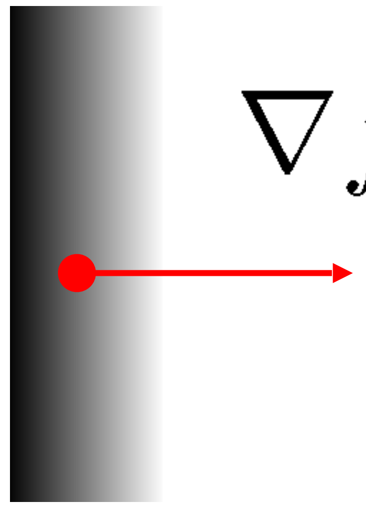
$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$



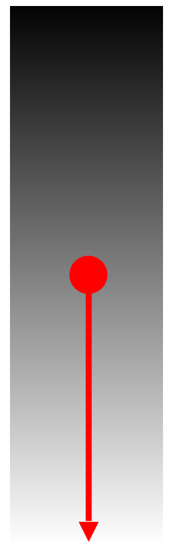
$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$

Image **Gradient**

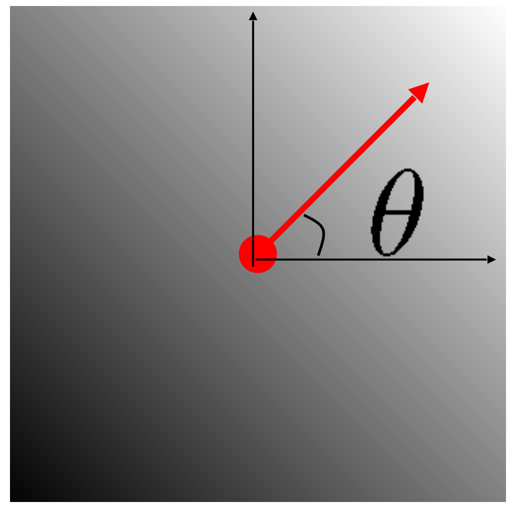
The gradient of an image: $$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$



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



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

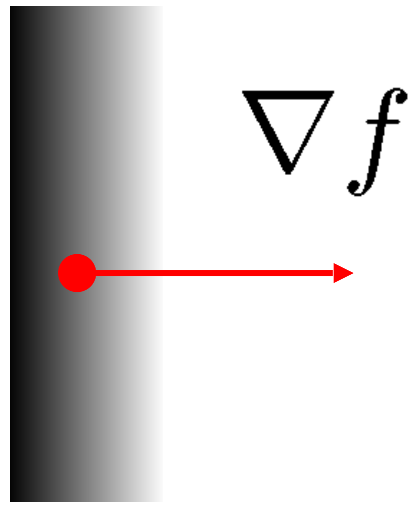


$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

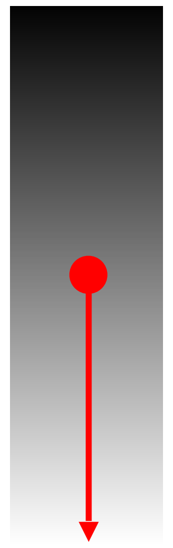
The gradient points in the direction of most rapid **increase of intensity**:

Image **Gradient**

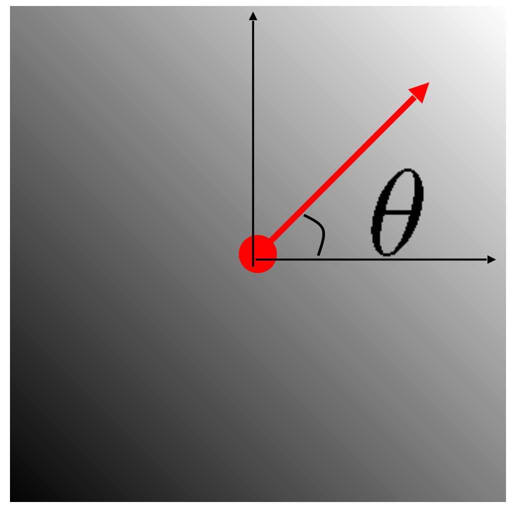
The gradient of an image: $$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$



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



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



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

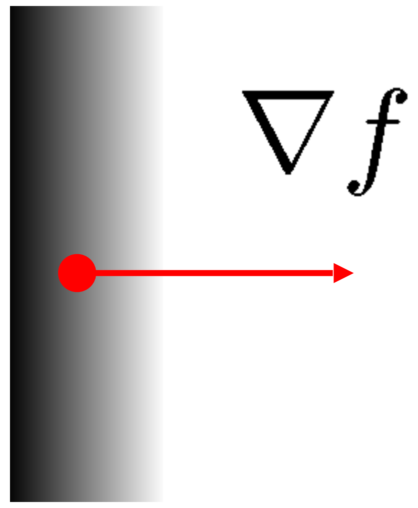
The gradient points in the direction of most rapid **increase of intensity**:

The **gradient direction** is given by:

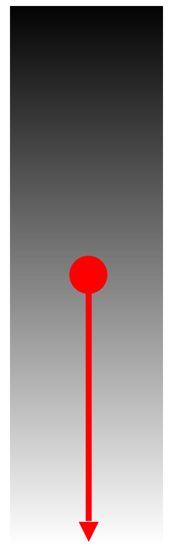
(how is this related to the direction of the edge?)

Image **Gradient**

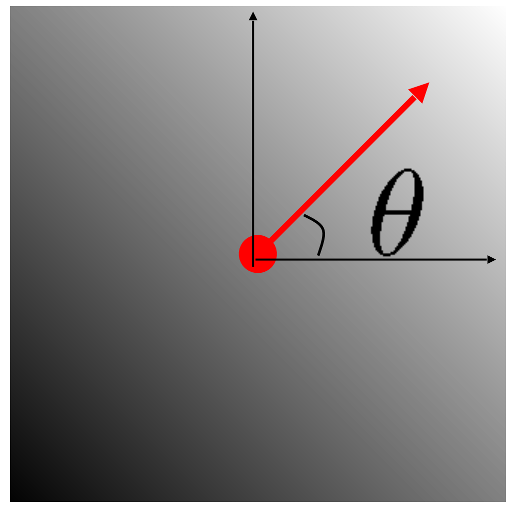
The gradient of an image: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$



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



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



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

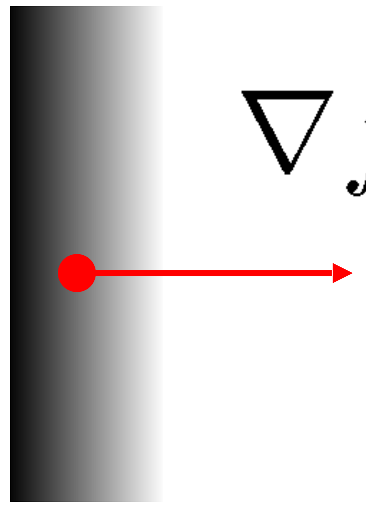
The gradient points in the direction of most rapid **increase of intensity**:

The **gradient direction** is given by: $\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$

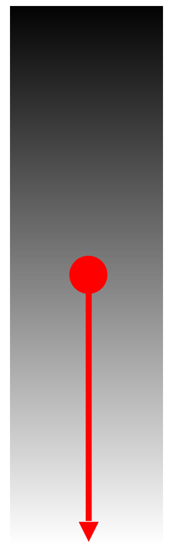
(how is this related to the direction of the edge?)

Image **Gradient**

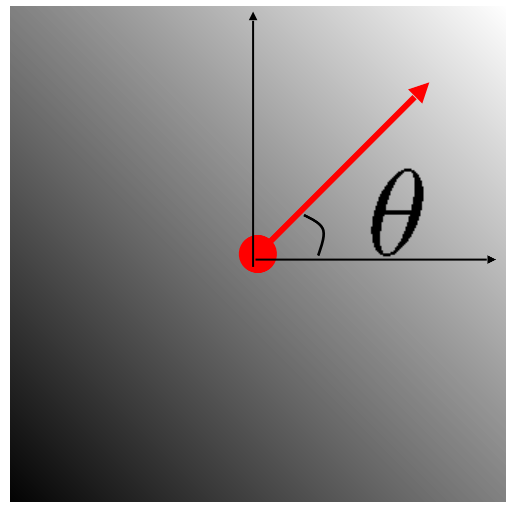
The gradient of an image: $$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$



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



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



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

The gradient points in the direction of most rapid **increase of intensity**:

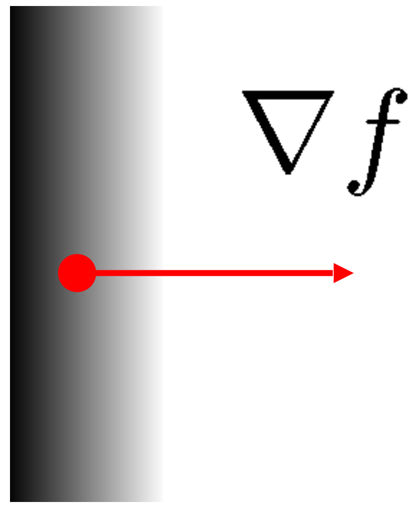
The **gradient direction** is given by:

(how is this related to the direction of the edge?)

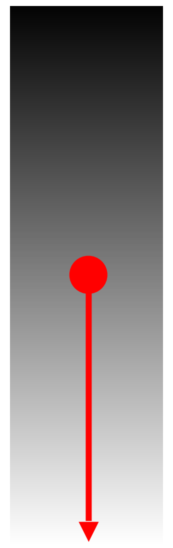
The edge strength is given by the **gradient magnitude**:

Image **Gradient**

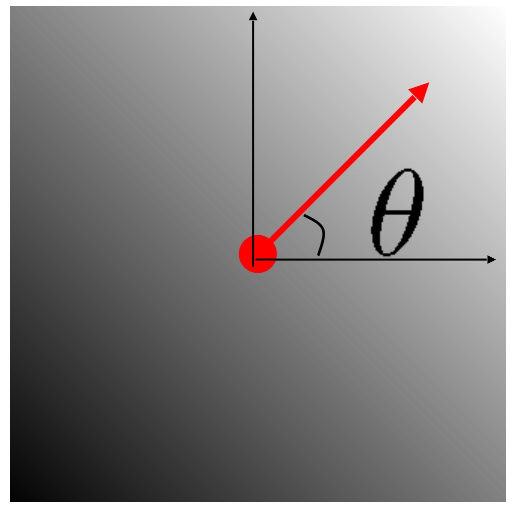
The gradient of an image: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$



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



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



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

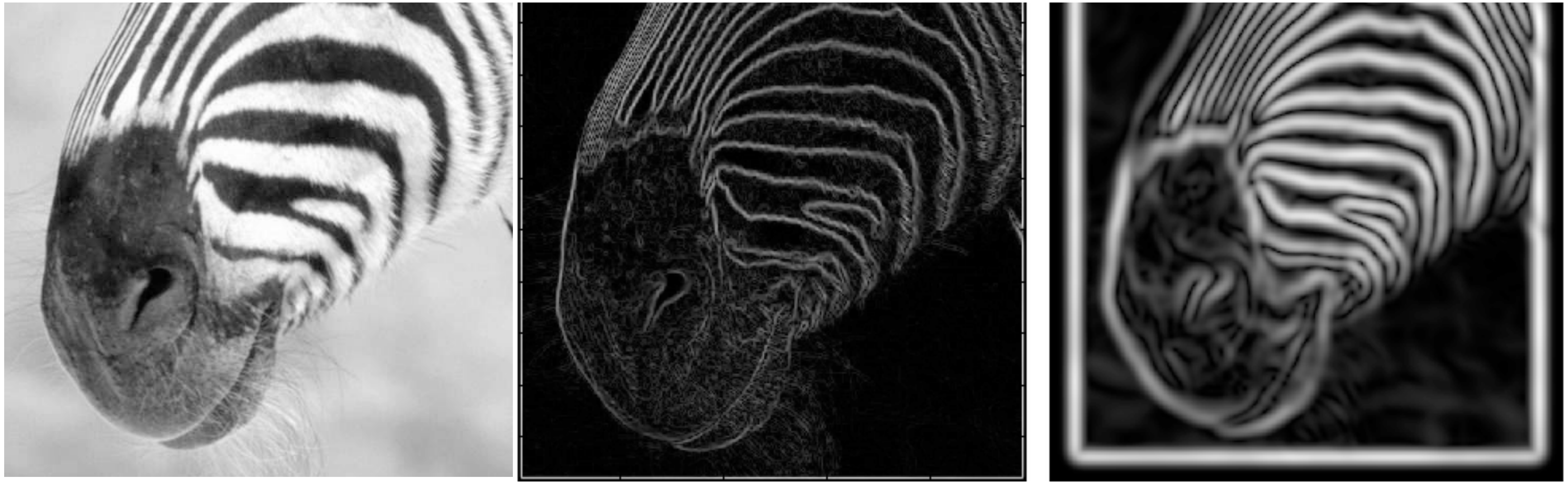
The gradient points in the direction of most rapid **increase of intensity**:

The **gradient direction** is given by: $\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$

(how is this related to the direction of the edge?)

The edge strength is given by the **gradient magnitude**: $\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$

Gradient **Magnitude**



$\sigma = 1$ $\sigma = 2$

Forsyth & Ponce (2nd ed.) Figure 5.4

Increased **smoothing**:

— eliminates noise edges

— makes edges smoother and thicker

— removes fine detail

Sobel Edge Detector

1. Use **central differencing** to compute gradient image (instead of first forward differencing). This is more accurate.

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

2. **Threshold** to obtain edges

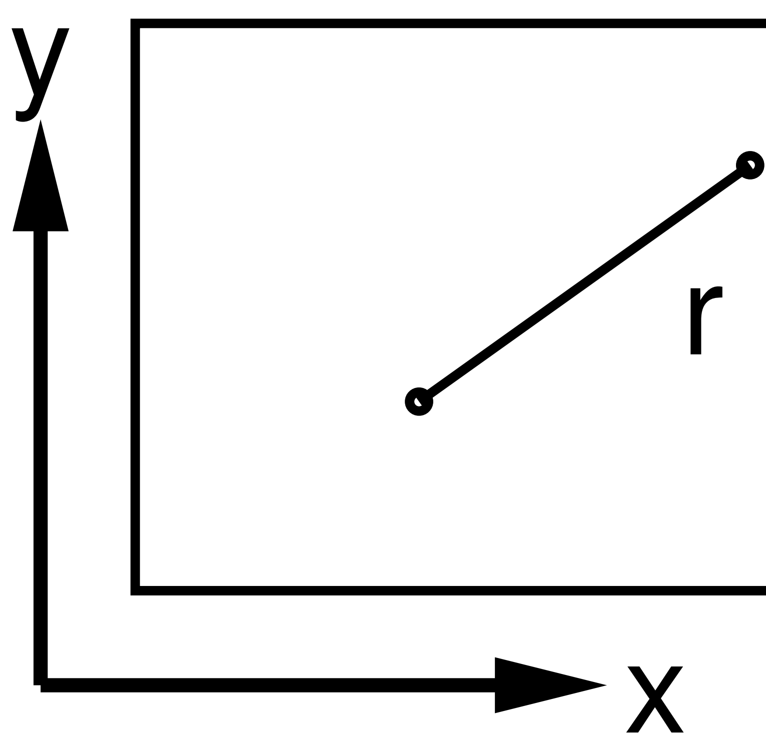
Original Image

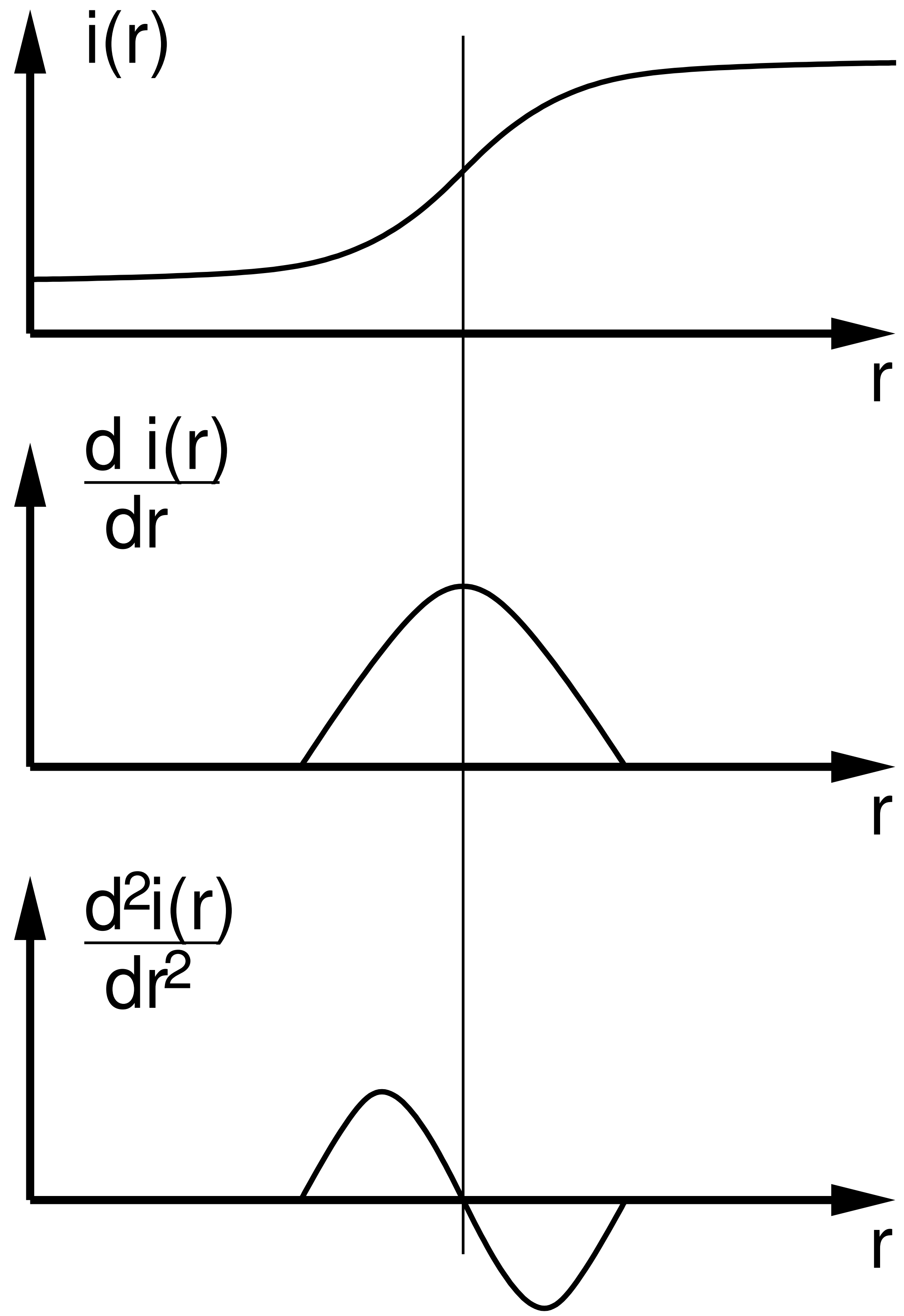
Sobel Gradient

Sobel Edges

Thresholds are brittle, we can do better!

Two Generic Approaches for **Edge** Detection





Two generic approaches to **edge point detection**:

— (significant) local extrema of a first derivative operator

— zero crossings of a second derivative operator

Marr / Hildreth **Laplacian of Gaussian**

A “**zero crossings** of a second derivative operator” approach

Design Criteria:

1. localization in space
2. localization in frequency
3. rotationally invariant

Marr / Hildreth **Laplacian of Gaussian**

A “**zero crossings** of a second derivative operator” approach

Steps:

1. Gaussian for smoothing

2. Laplacian (∇^2) for differentiation where

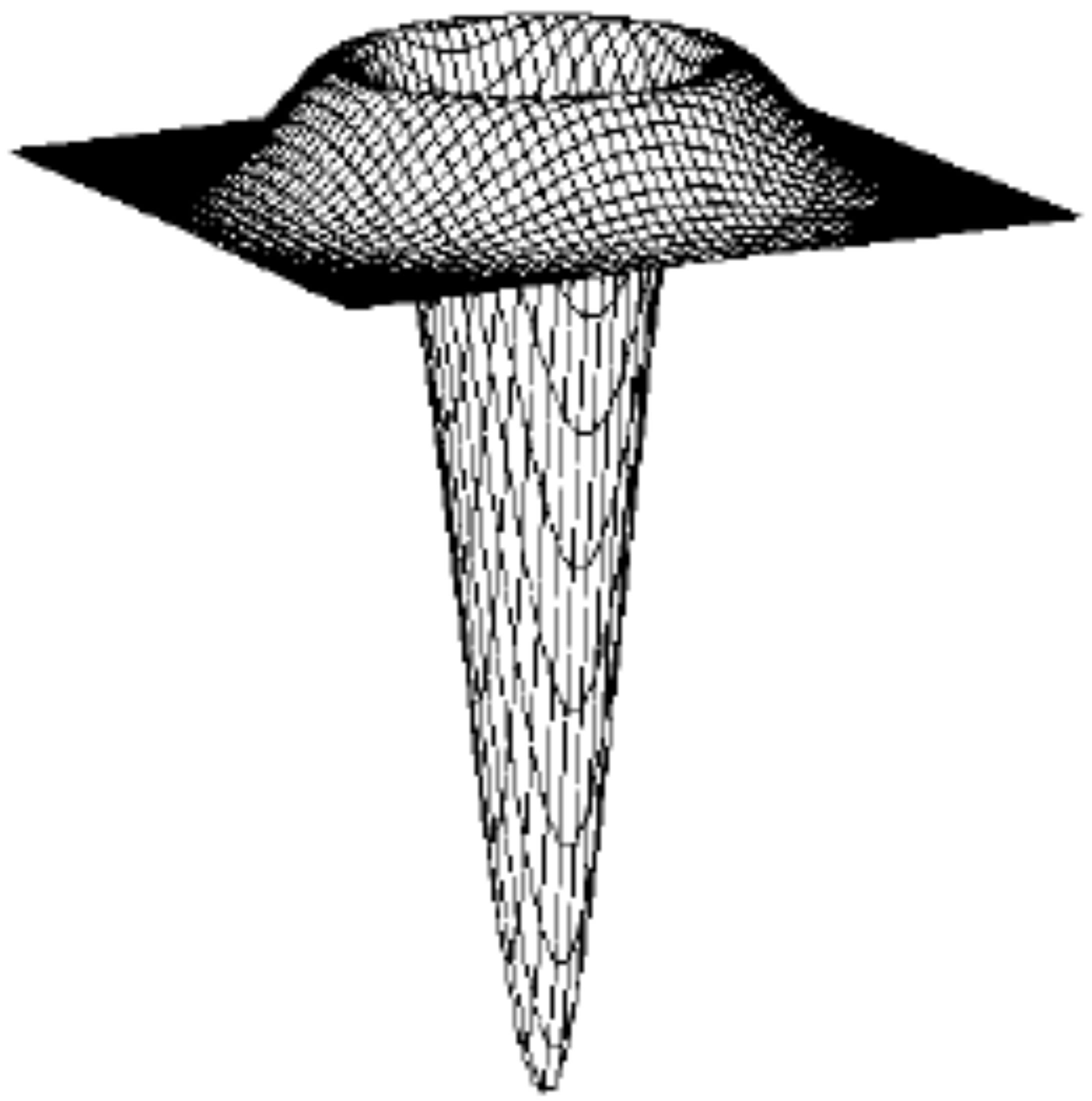
$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

3. Locate zero-crossings in the Laplacian of the Gaussian ($\nabla^2 G$) where

$$\nabla^2 G(x,y) = \frac{-1}{2\pi\sigma^4}\left[2 - \frac{x^2+y^2}{\sigma^2}\right]\exp^{-\frac{x^2+y^2}{2\sigma^2}}$$

Marr / Hildreth **Laplacian of Gaussian**

Here's a 3D plot of the Laplacian of the Gaussian ($\nabla^2 G$)



. . . with its characteristic "Mexican hat" shape

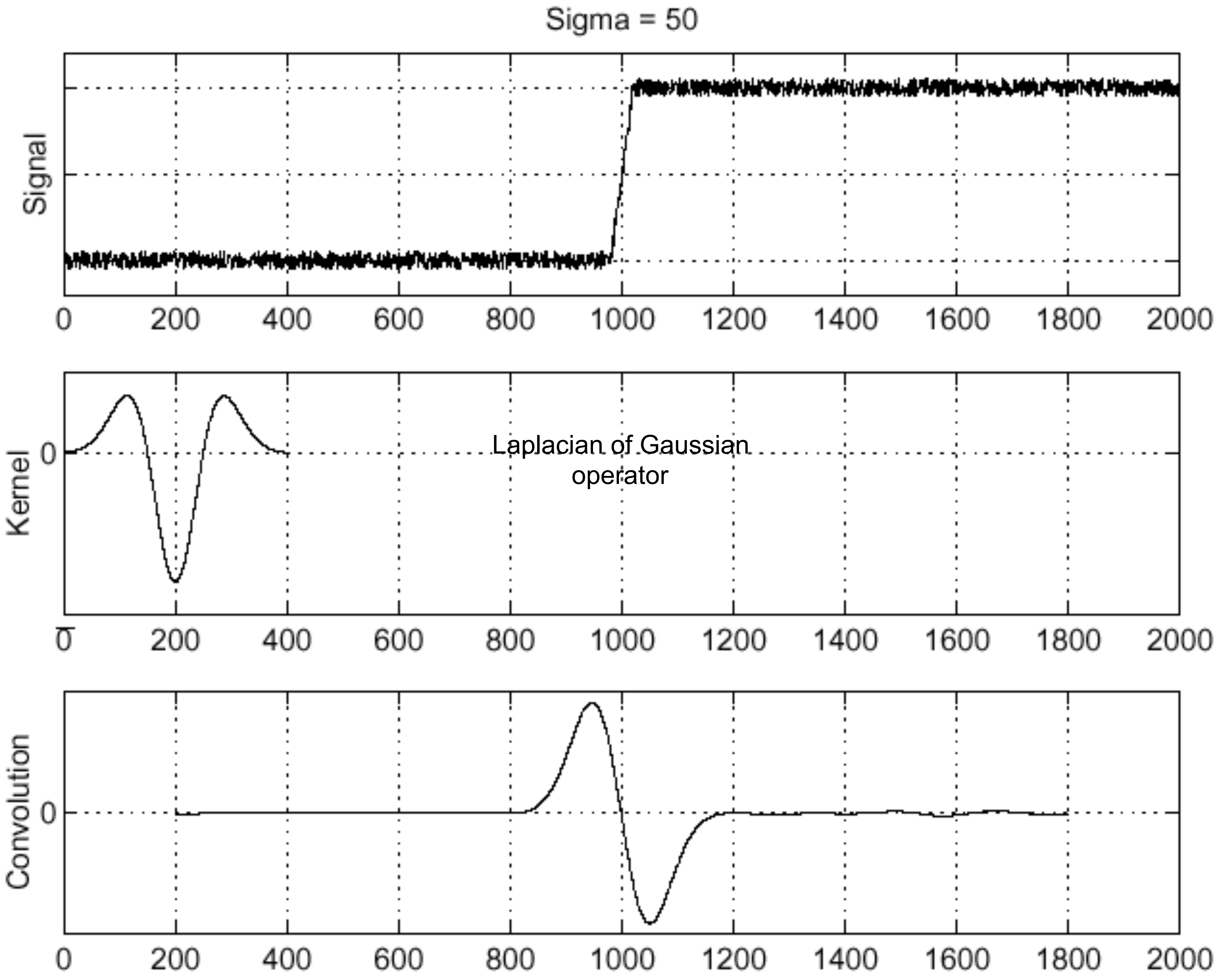
1D **Example**: Continued

Lets consider a row of pixels in an image:

$$I(X, 245)$$

$$\nabla^2 G$$

$$\nabla^2 G \otimes I(X, Y)$$



Where is the edge? Zero-crossings of bottom graph

Marr / Hildreth **Laplacian of Gaussian**

5 x 5 LoG filter

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

17 x 17 LoG filter

0	0	0	0	0	0	-1	-1	**-1**	-1	-1	0	0	0	0	0
0	0	0	0	-1	-1	-1	-1	**-1**	-1	-1	-1	-1	0	0	0
0	0	-1	-1	-1	-2	-3	-3	**-3**	-3	-3	-2	-1	-1	-1	0
0	0	-1	-1	-2	-3	-3	-3	**-3**	-3	-3	-3	-2	-1	-1	0
0	-1	-1	-2	-3	-3	-3	-2	**-3**	-2	-3	-3	-3	-2	-1	-1
0	-1	-2	-3	-3	-3	0	2	**4**	2	0	-3	-3	-3	-2	-1
-1	-1	-3	-3	-3	0	4	10	**12**	10	4	0	-3	-3	-3	-1
-1	-1	-3	-3	-2	2	10	18	**21**	18	10	2	-2	-3	-3	-1
-1	**-1**	**-3**	**-3**	**-3**	**4**	**12**	**21**	**24**	**21**	**12**	**4**	**-3**	**-3**	**-3**	**-1**
-1	-1	-3	-3	-2	2	10	18	**21**	18	10	2	-2	-3	-3	-1
-1	-1	-3	-3	-3	0	4	10	**12**	10	4	0	-3	-3	-3	-1
0	-1	-2	-3	-3	-3	0	2	**4**	2	0	-3	-3	-3	-2	-1
0	-1	-1	-2	-3	-3	-3	-2	**-3**	-2	-3	-3	-3	-2	-1	-1
0	-1	-1	-2	-3	-3	-3	-2	**-3**	-2	-3	-3	-3	-2	-1	-1
0	0	-1	-1	-1	-2	-3	-3	**-3**	-3	-3	-2	-1	-1	-1	0
0	0	0	0	-1	-1	-1	-1	**-1**	-1	-1	-1	-1	0	0	0

Scale (σ)

Image From: A. Campilho

Marr / Hildreth **Laplacian of Gaussian**

Image From: A. Campilho

Assignment 1: High Frequency Image

original

-

smoothed
(5x5 Gaussian)

=

original - smoothed
(scaled by 4, offset +128)

Assignment 1: High Frequency Image

original

-

smoothed
(5x5 Gaussian)

=

smoothed - original
(scaled by 4, offset +128)

Assignment 1: High Frequency Image