Image Gradients

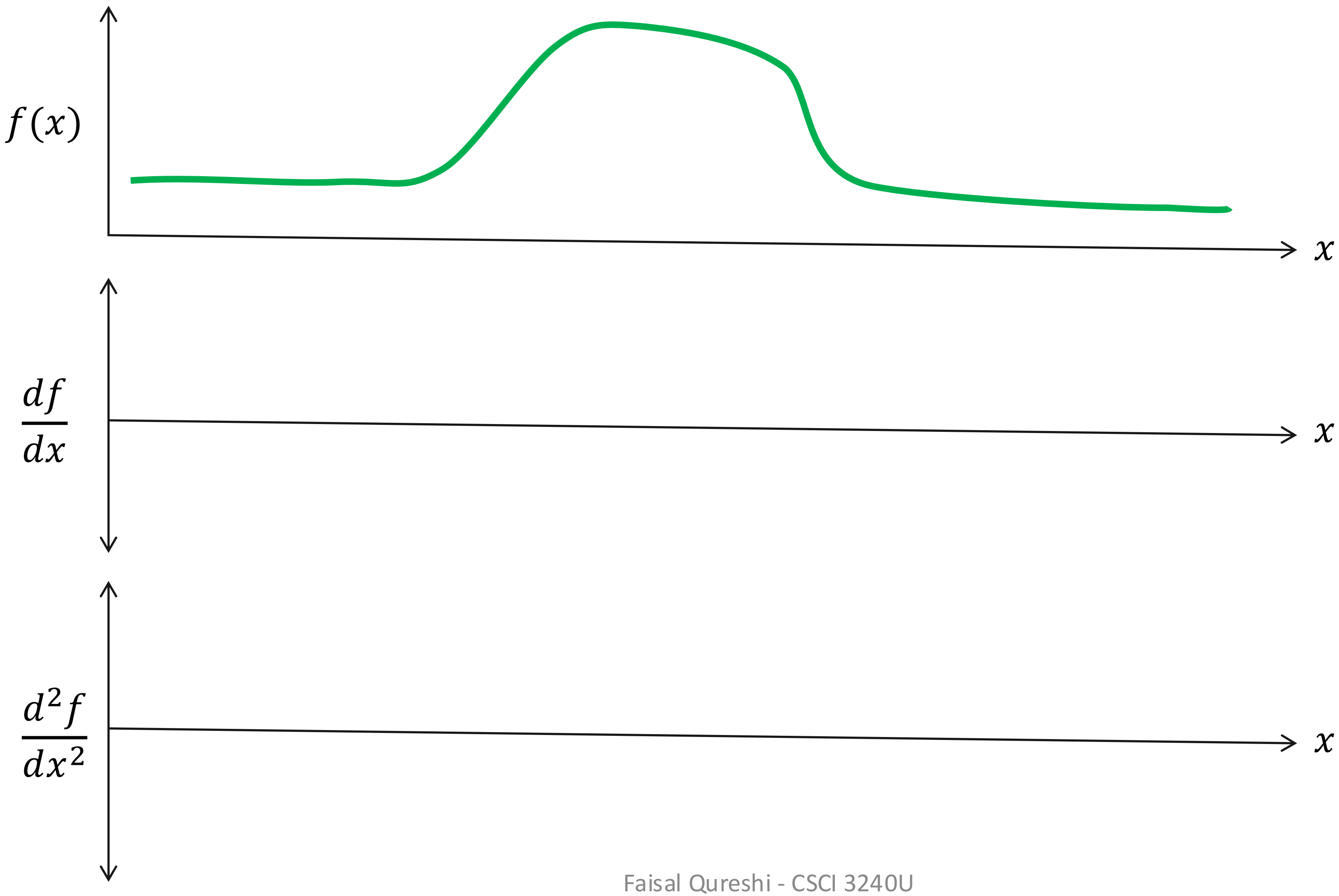
Computational Photography (CSCI 3240U)

Faisal Z. Qureshi
http://vclab.science.ontariotechu.ca
Ontario Tech University

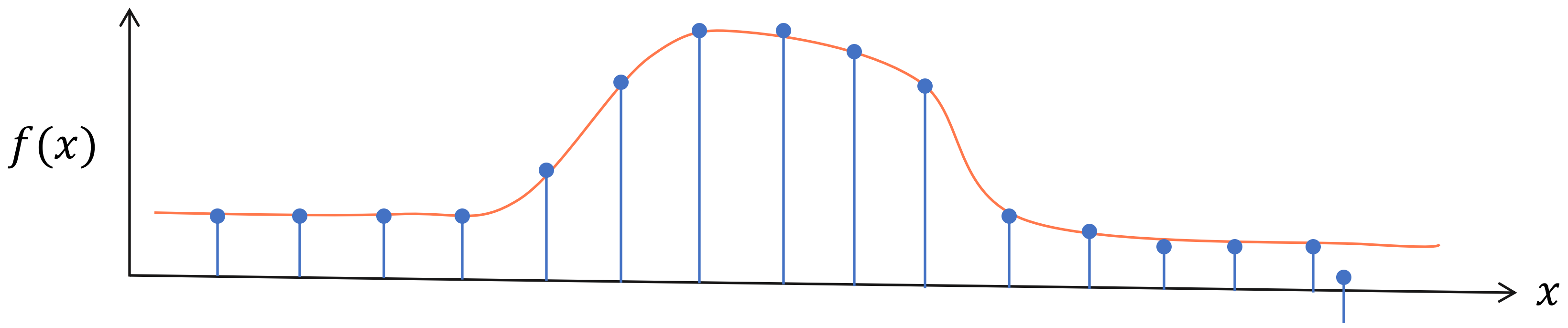
Today's lecture

- Why do we care about image gradients?
- Computing image gradients
- Sobel filters
- Gradient magnitude and directions
- Visualizing image gradients

Derivative: $$\frac{df}{dx} = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon}$$



Derivative: $$\frac{df}{dx} = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon}$$



Finite-difference approximation

$$\frac{df}{dx} \approx \frac{\Delta f}{\Delta x} = \frac{f(x+1) - f(x)}{(x+1) - x} = f(x+1) - f(x)$$

Use finite difference approximation to compute image derivatives

$$\frac{df}{dx} \approx \frac{\Delta f}{\Delta x} = \frac{f(x+1) - f(x)}{(x+1) - x} = f(x+1) - f(x)$$

$f =$

1	1	9	8	6	0	0
0	1	2	3	4	5	6

$f' =$

$f'' =$

Use finite difference approximation to compute image derivatives

$$\frac{df}{dx} \approx \frac{\Delta f}{\Delta x} = \frac{f(x+1) - f(x)}{(x+1) - x} = f(x+1) - f(x)$$

$f =$

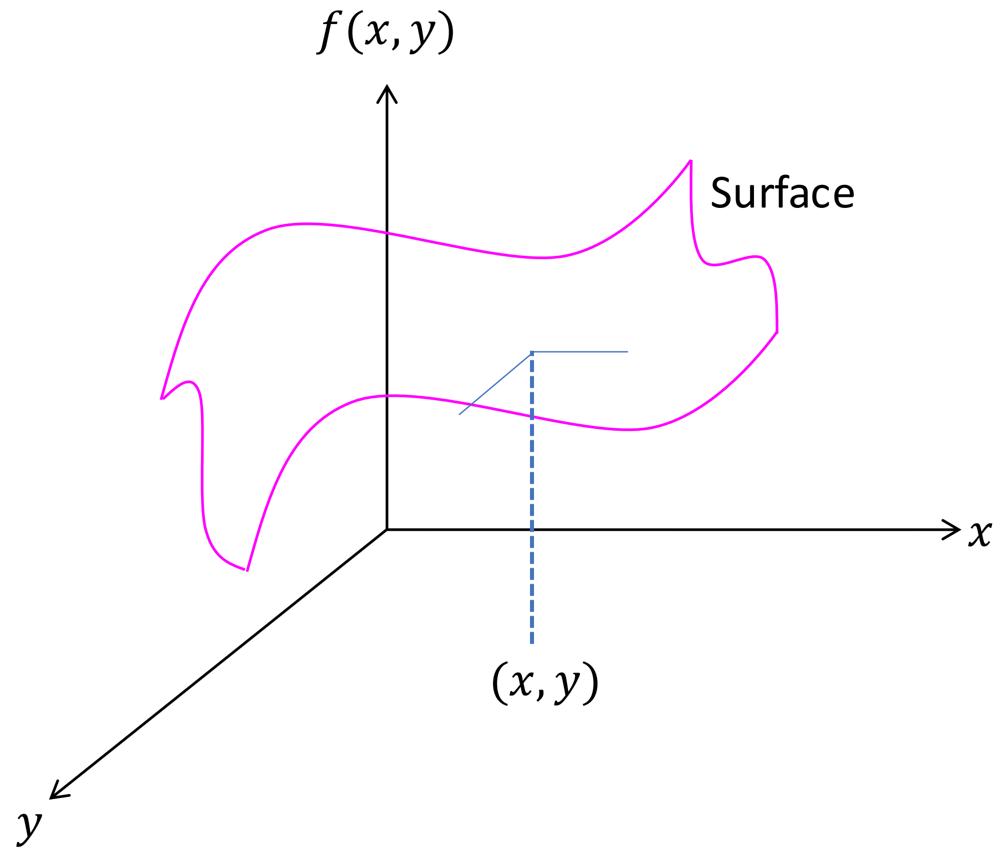
0	1	2	3	4	5	6
1	**1**	**9**	**8**	**6**	**0**	**0**

$f' =$

0	**8**	**-1**	**-2**	**-6**	**0**	**?**

$f * [1, -1] =$

Partial derivatives



How to compute image derivatives?

- Option 1: reconstruct a continuous function $f(x, y)$, then compute partial derivatives as f

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

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

How to compute image derivatives?

- Option 2: use finite differences to take a discrete derivative as

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

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

- We can achieve this using *convolution*

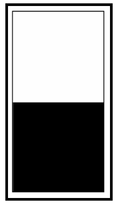
$H_x =$  $H_y =$ 

Image derivatives in x and y directions

Image

$f =$

1	1	9	8	1
8	8	8	8	8
1	3	5	8	1
5	3	2	8	6

Derivative along x

$f_x = f * [1, -1] =$

Derivative along y

$f_y = f * [1, -1]^T =$

Image gradient: $\nabla f = \left[\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y}\right]$

$f_x =$

0	8	-1	-7	
0	0	0	0	
2	2	3	-7	
-2	-1	6	-2	

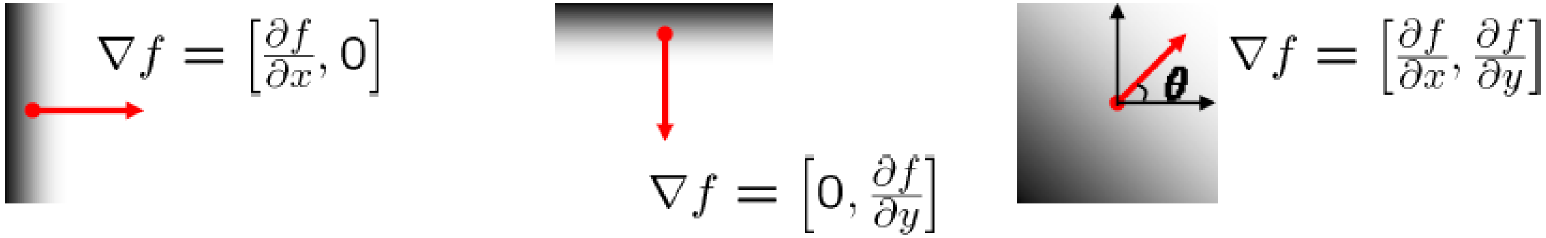
$f_y =$

7	7	-1	0	7
-7	-5	-3	0	7
4	0	-3	0	5

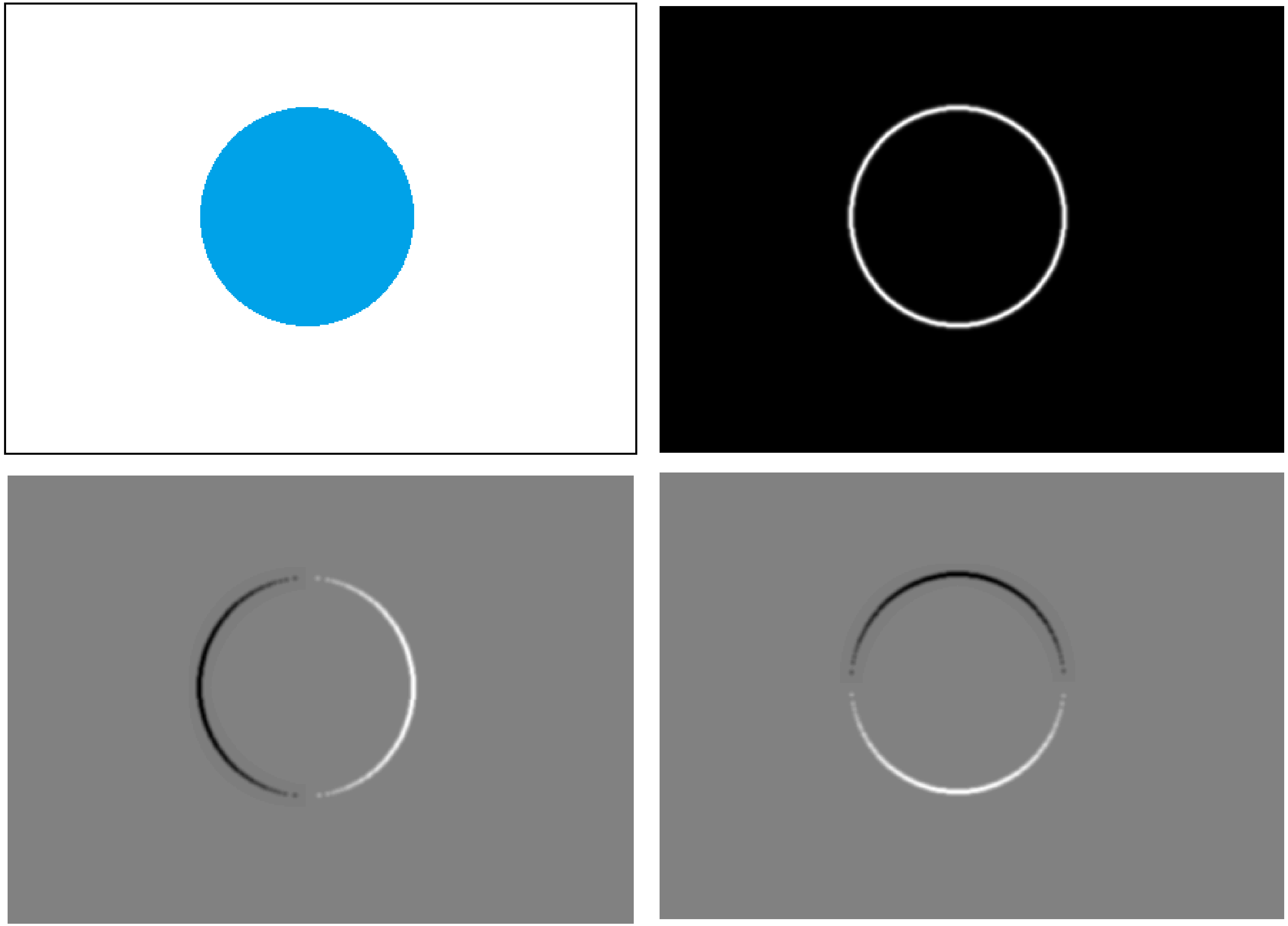
What is gradient at pixel location $(0,2)$? $(-1,-1)$

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

- The gradient of an image ∇f points to the direction of most rapid change in intensity



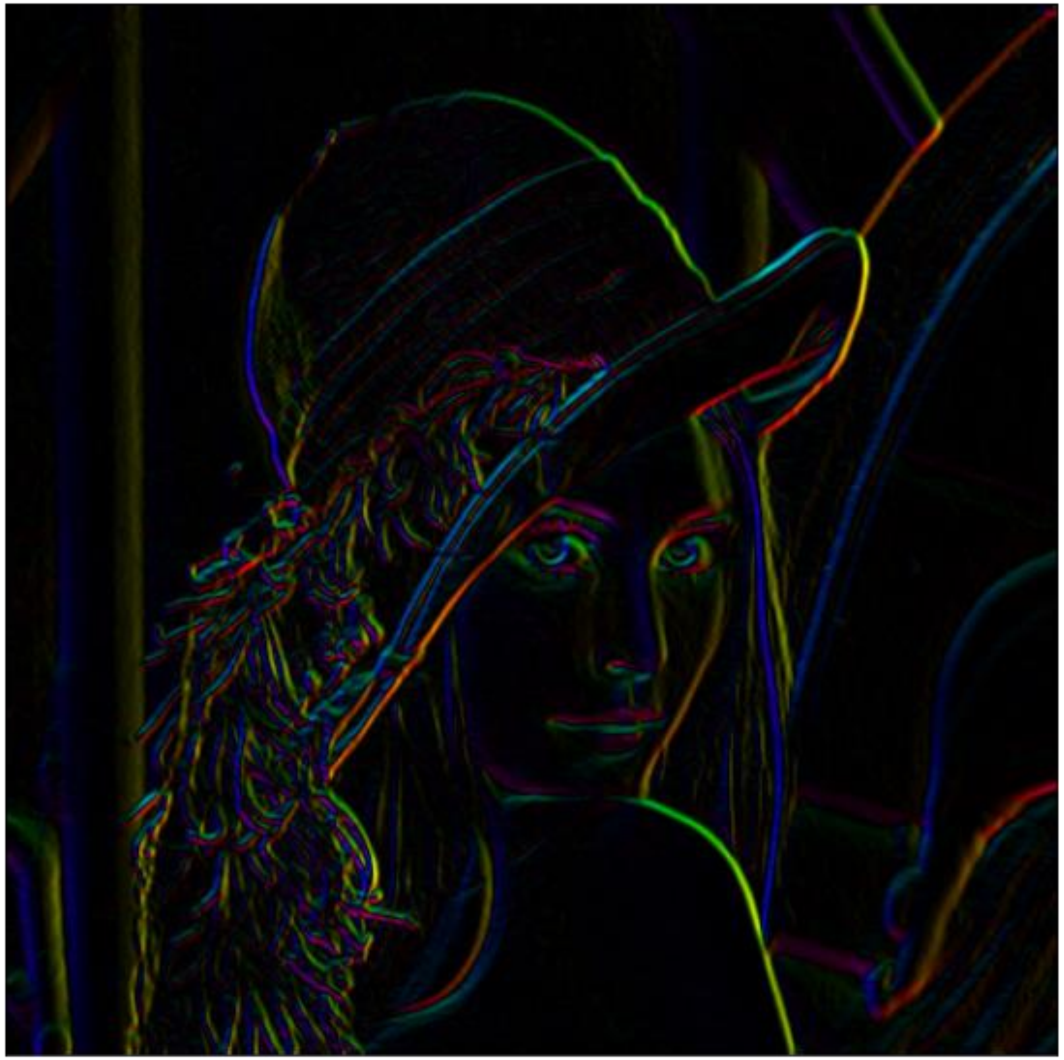
- Gradient magnitude: $\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$
- Gradient direction: $\theta = \tan^{-1}\left(\frac{\partial I}{\partial y} / \frac{\partial I}{\partial x}\right)$



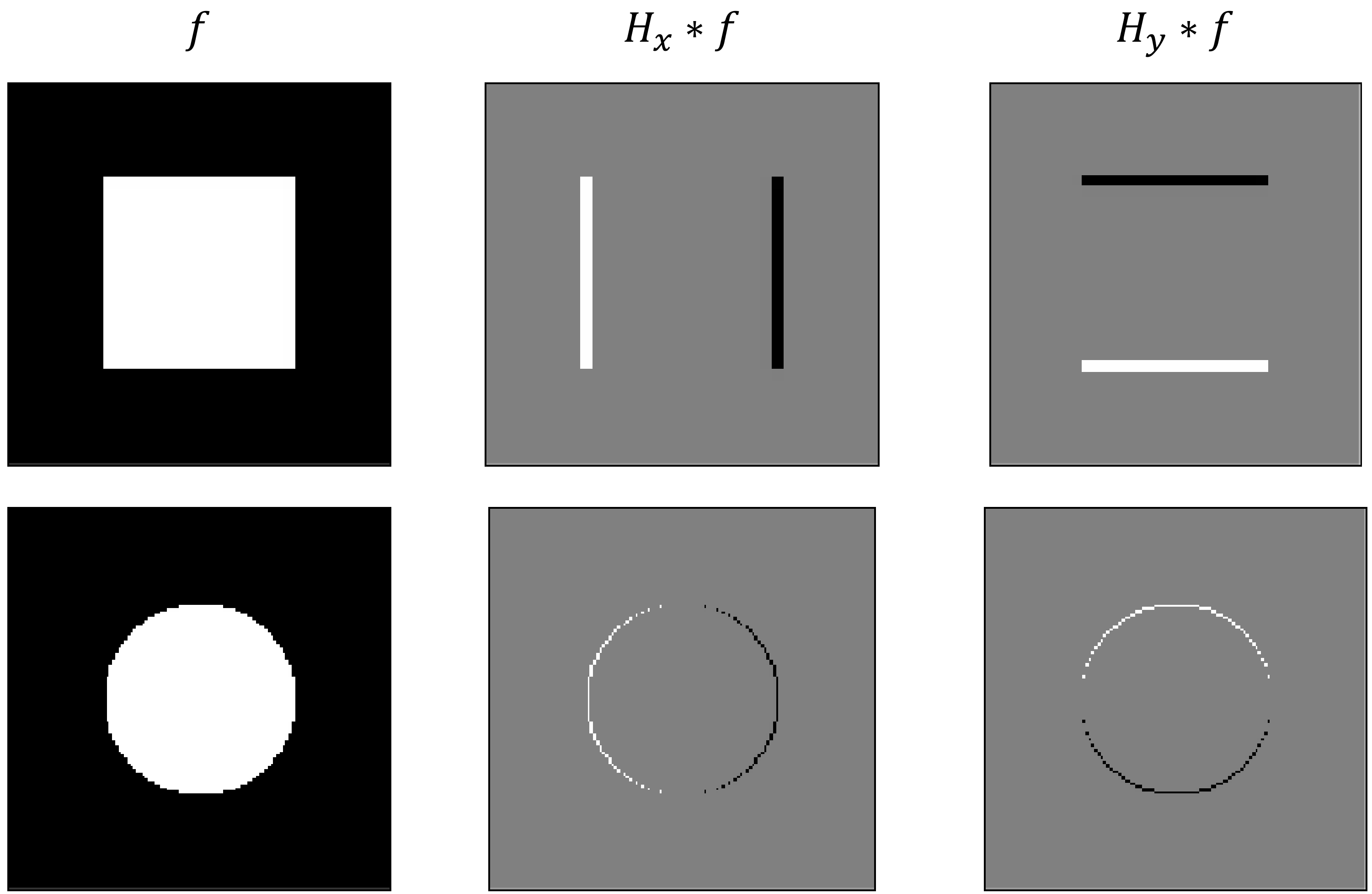
Gradient direction and magnitude

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

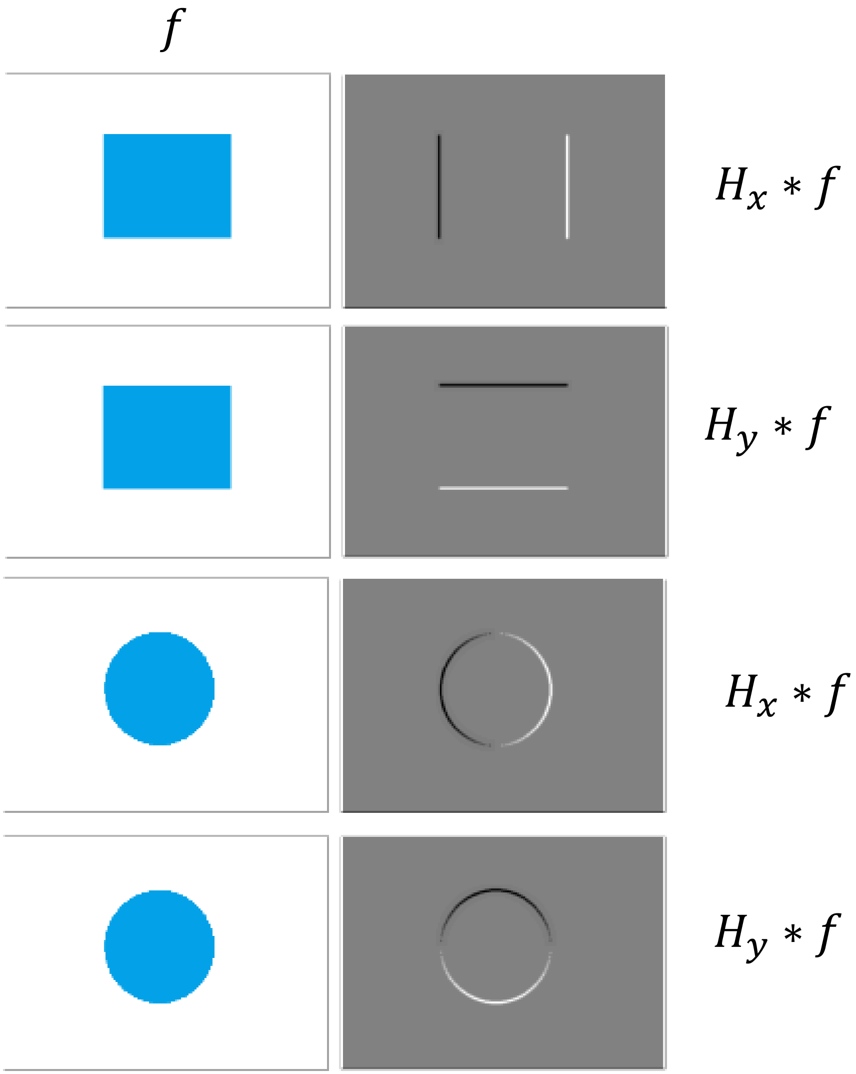
$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} \Big/ \frac{\partial f}{\partial x}\right)$$



Partial Derivative of an Image



Partial Derivative of an Image



$$H_x = \begin{bmatrix} 1 & -1 \end{bmatrix}$$

$$H_y = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Partial Derivatives of an Image

Filters for computing image derivatives

Finite Difference Filters

Sobel

$H_x =$

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

$H_y =$

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

Prewire

$H_x =$

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

$H_y =$

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

Roberts

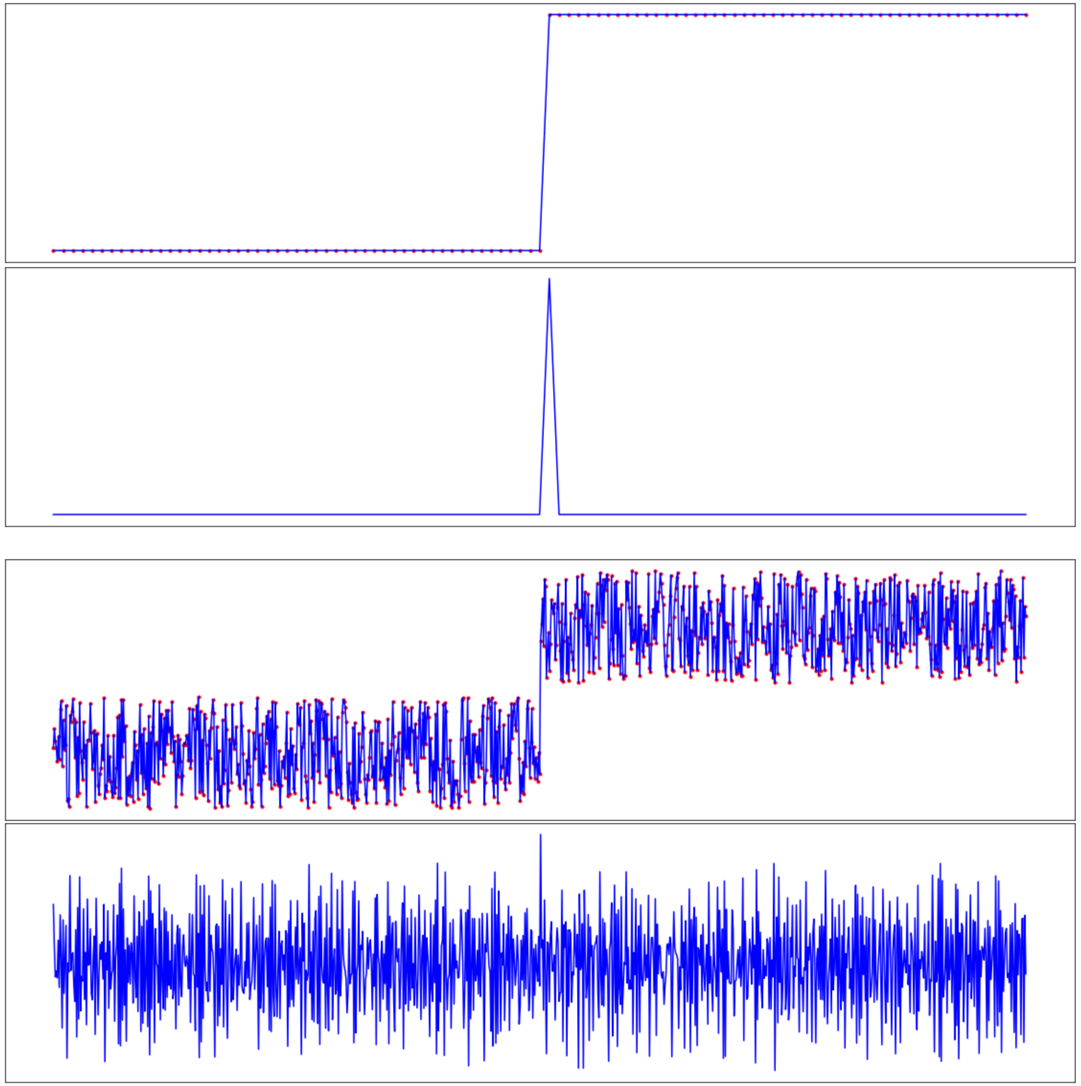
$H_x =$

0	-1
1	0

$H_y =$

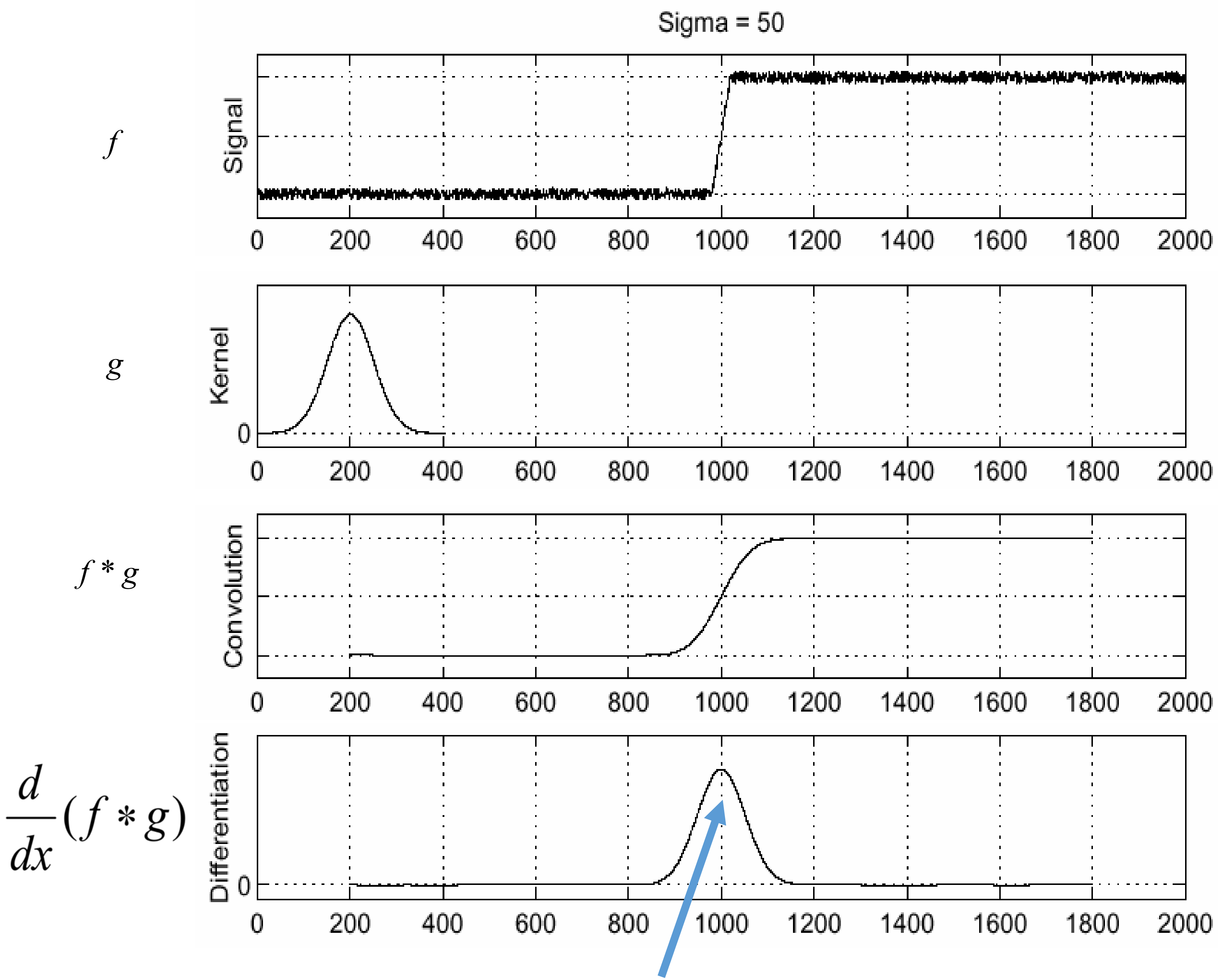
1	0
0	-1

Image noise and gradients



Effects of Noise

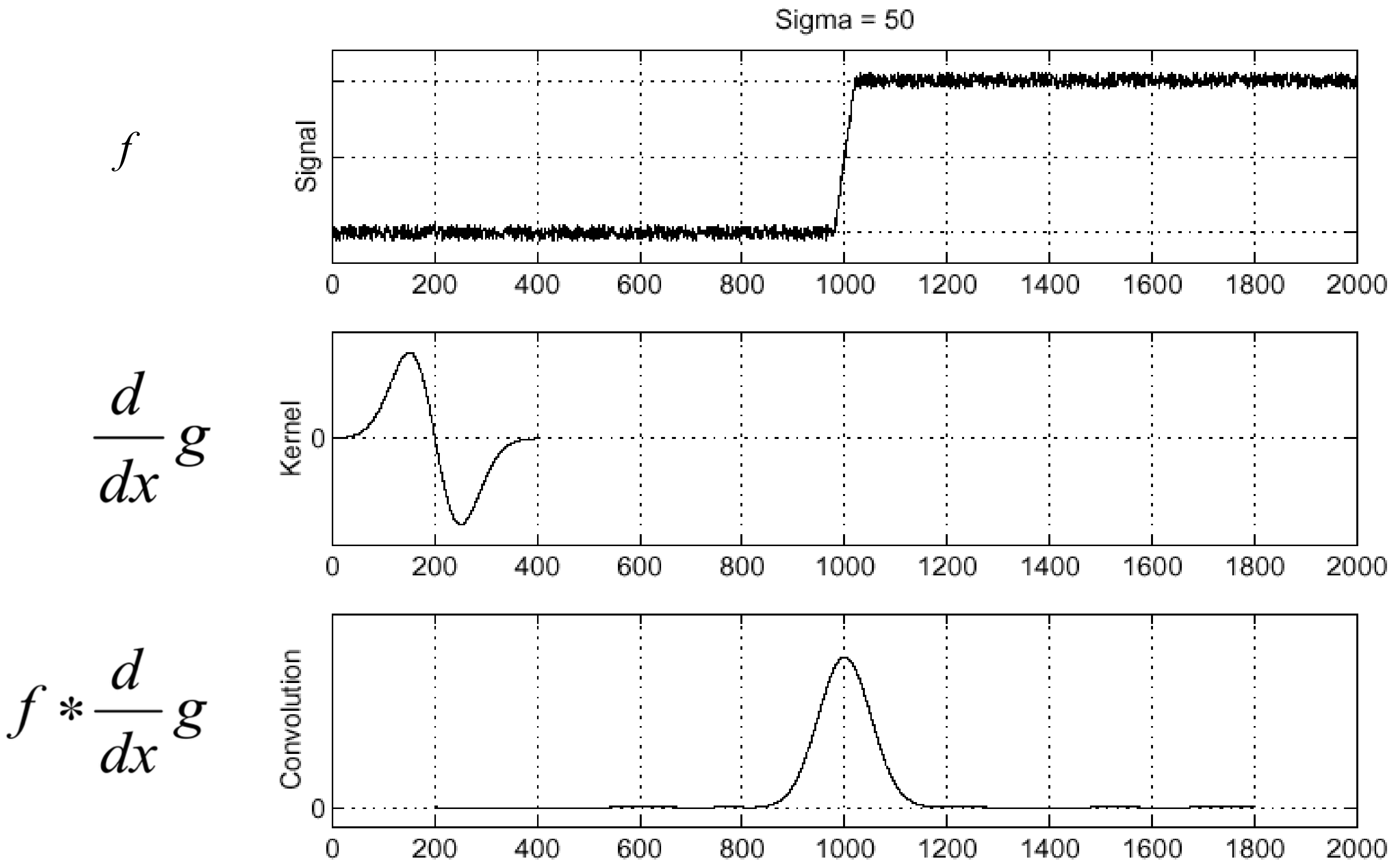
- Gradient is highly sensitive to noise
- Difference filters (that we can use for gradient computation) respond strongly to noise
  - The larger the noise, the stronger the response
- How to handle it?
  - Smooth first. Get rid of high-frequency component.



Peak corresponds to edge

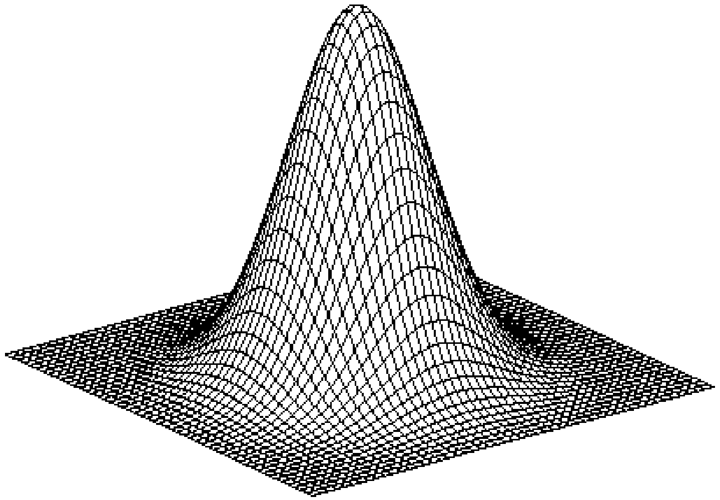
Differentiation via Convolution

- Convolution is associative: $\frac{d}{dx}(f * g) = f * \frac{d}{dx} g$



We saved one convolutional operation

Derivative of Gaussian filter



$* [1, -1] \ =$

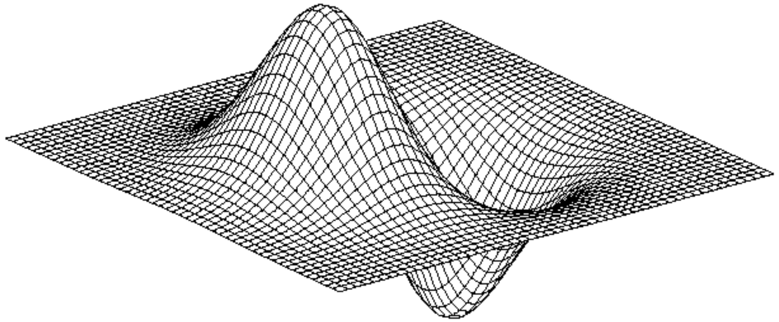
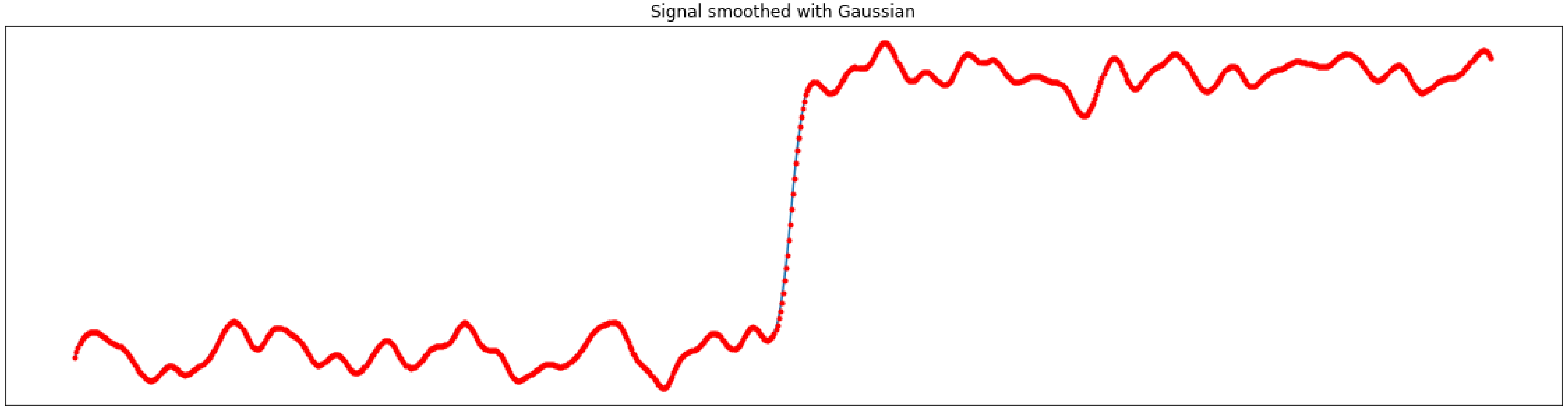
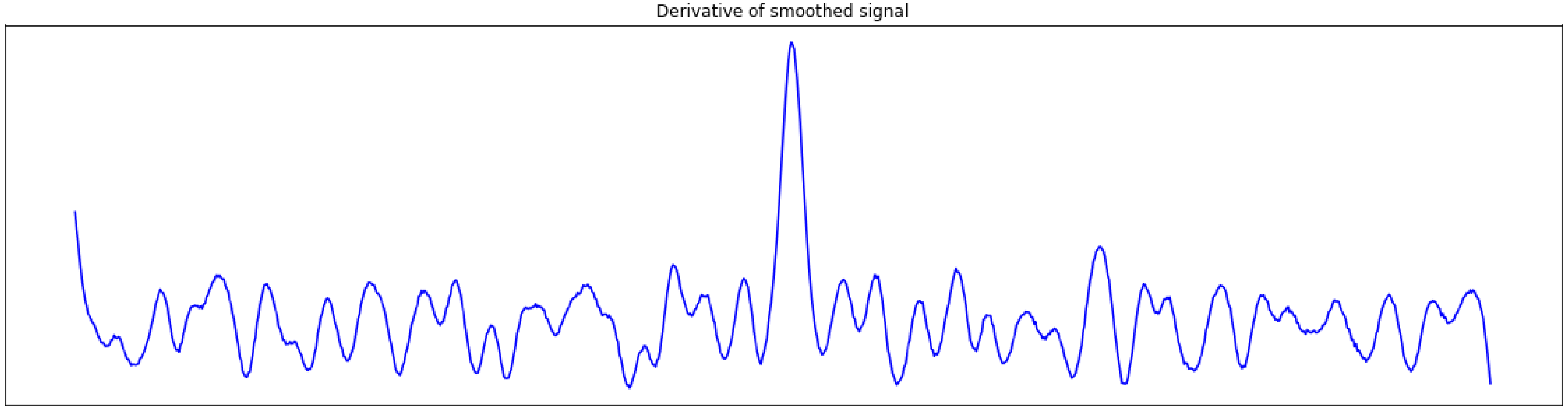
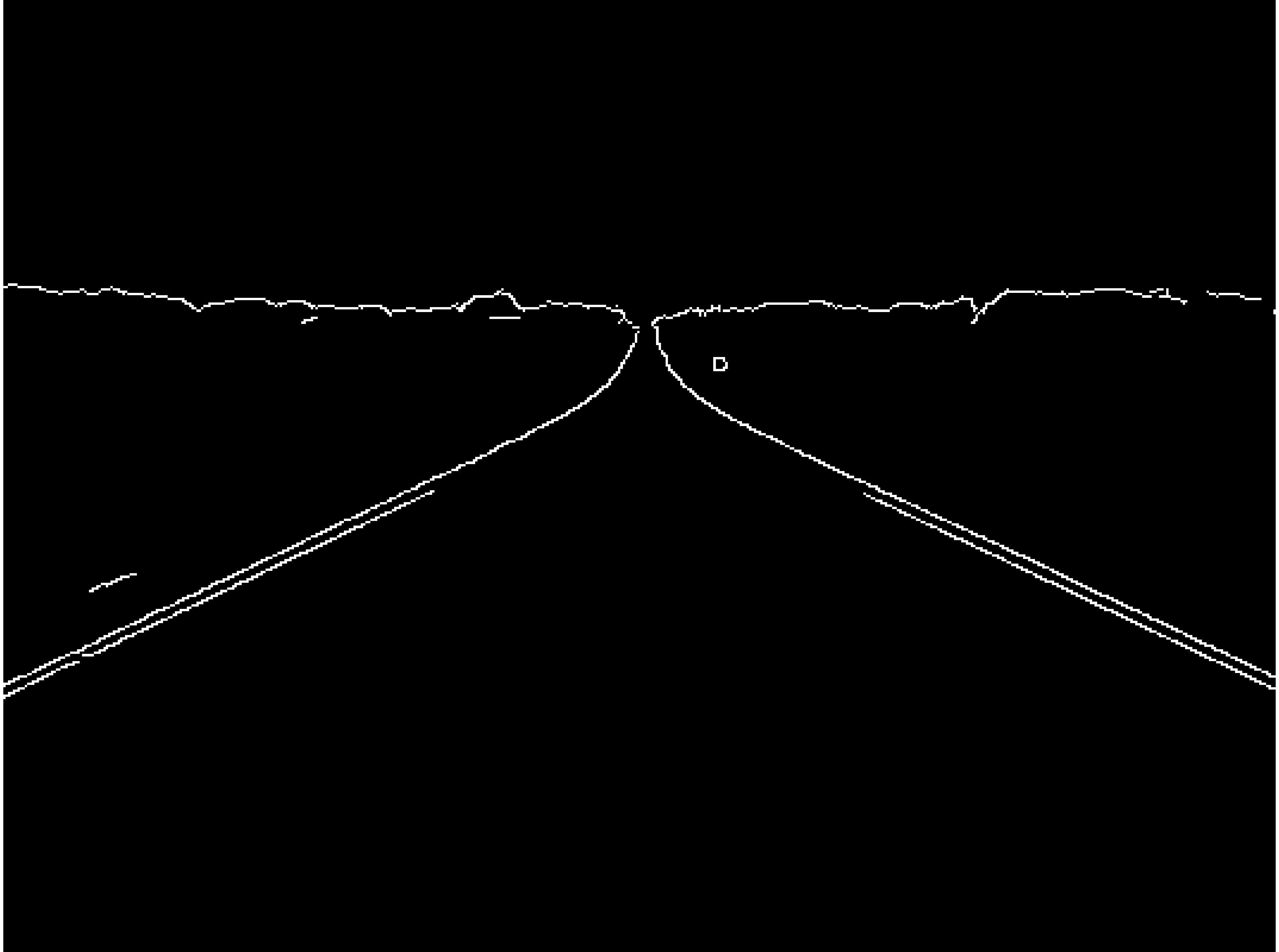


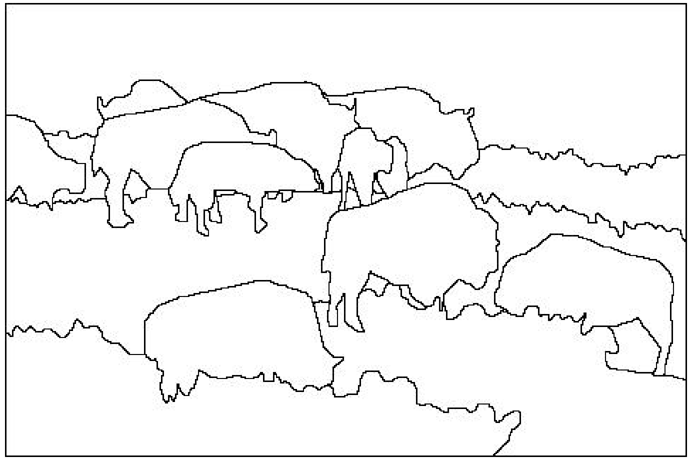
Image noise and gradients

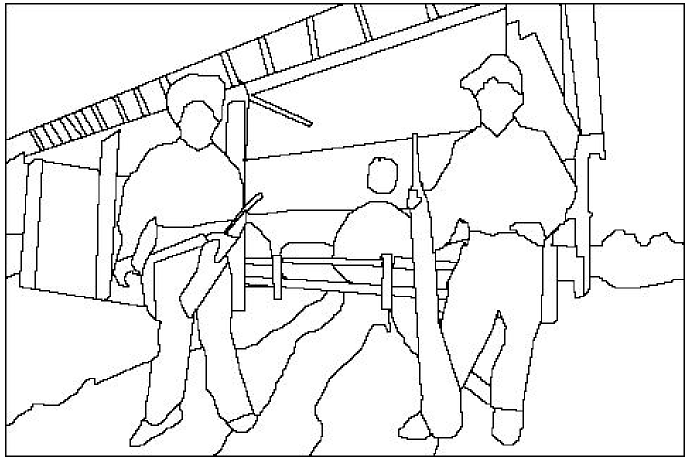






Edge Detection





Edge Detection

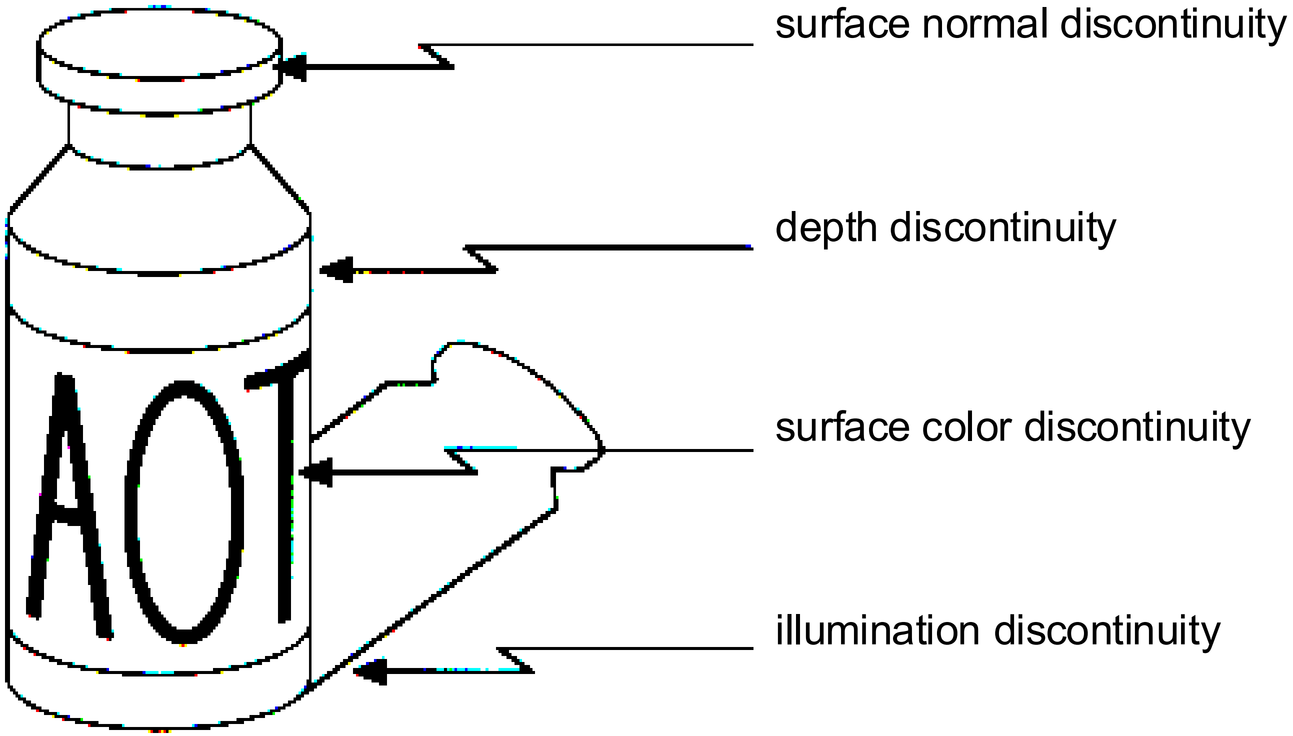
- Identify sudden changes (discontinuities) in an image
- Most semantic and shape information seen in an image can be encoded using the edges
- Edges are more compact than pixels

An artist's line drawing

[Source: D. Lowe]

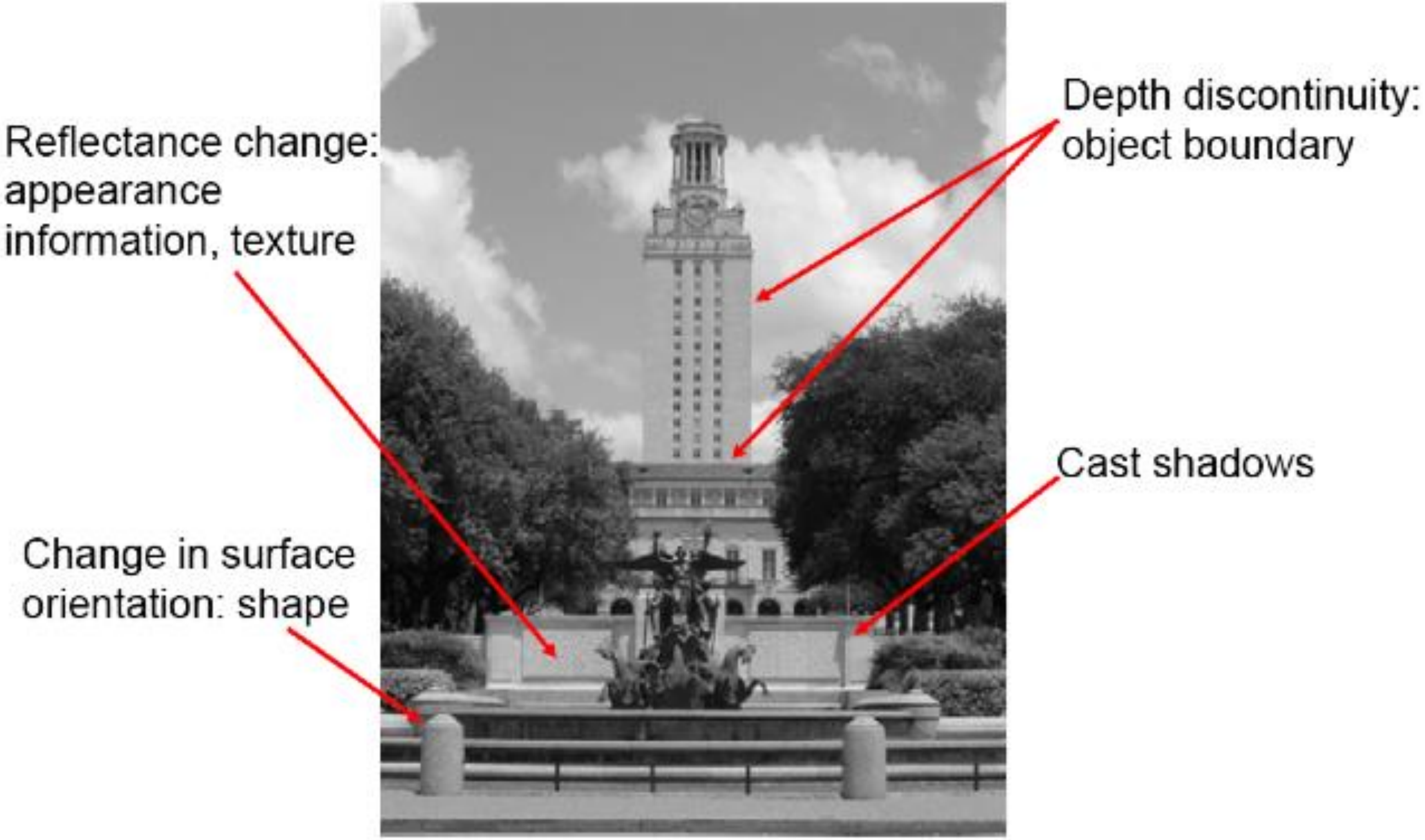
Origin of Edges

- Edges are caused by



[Source: Steve Seitz]

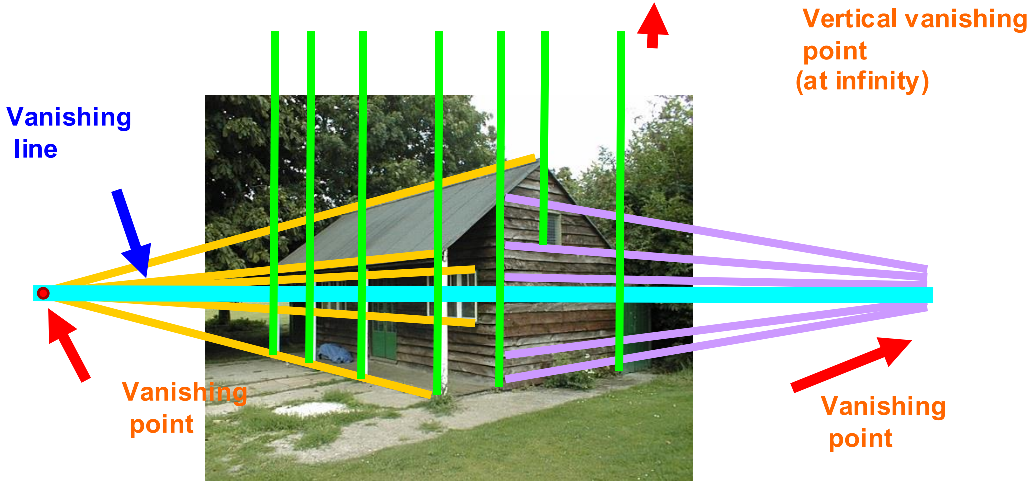
What causes edges?

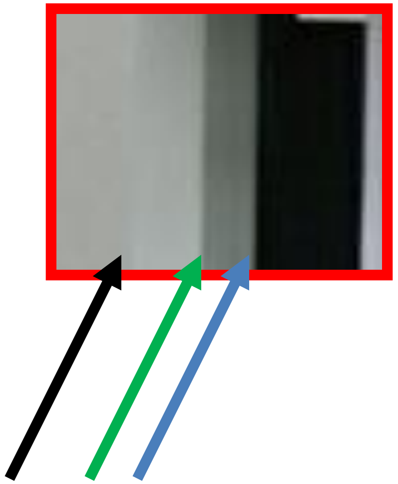


[Source: K. Grauman]

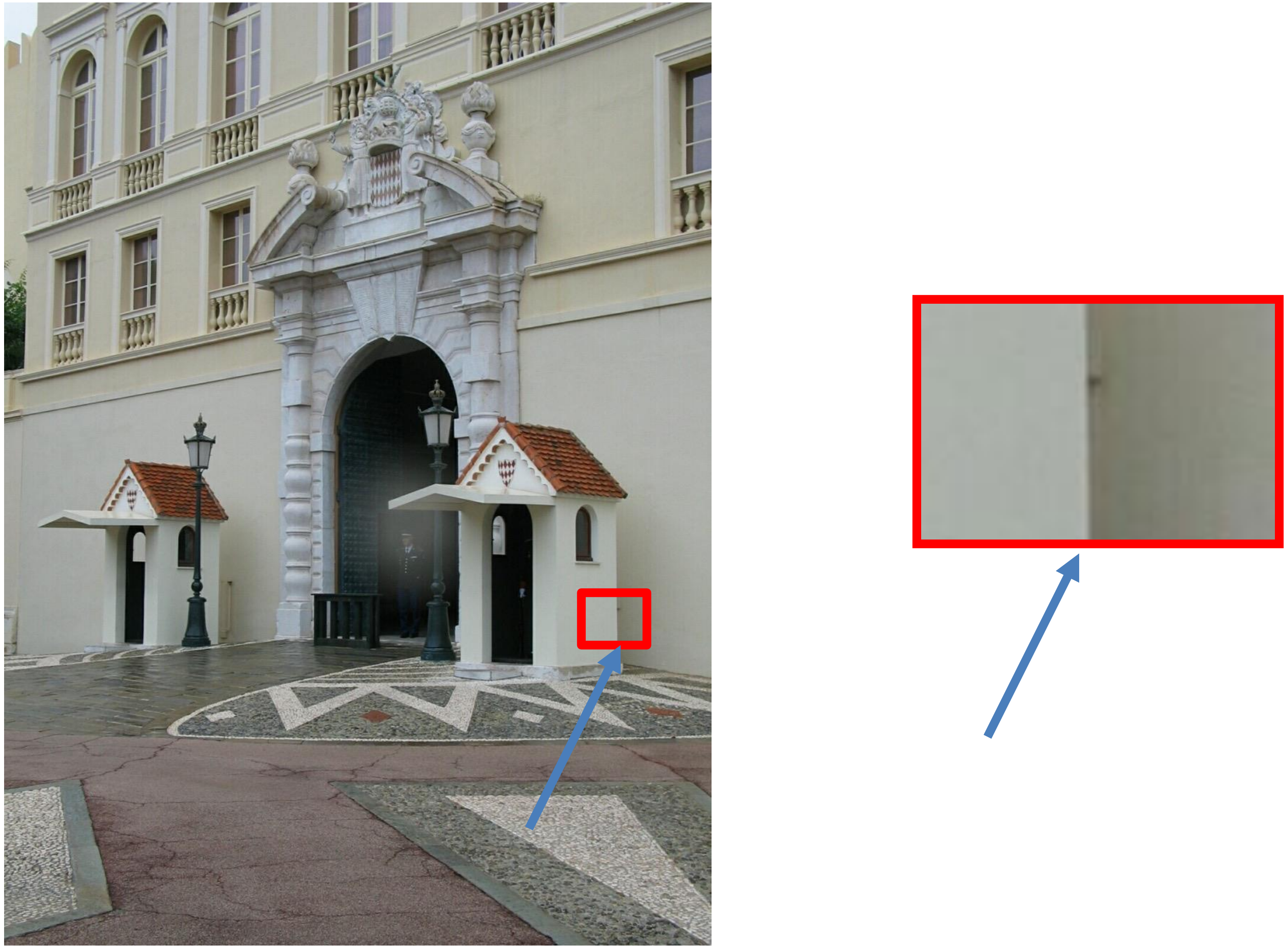
Why edge detection?

- Extract information
- Recognize objects
- Understand scene
- Reconstruct 3D from images
    - Recover viewpoint and geometry





[Source: Steve Seitz]

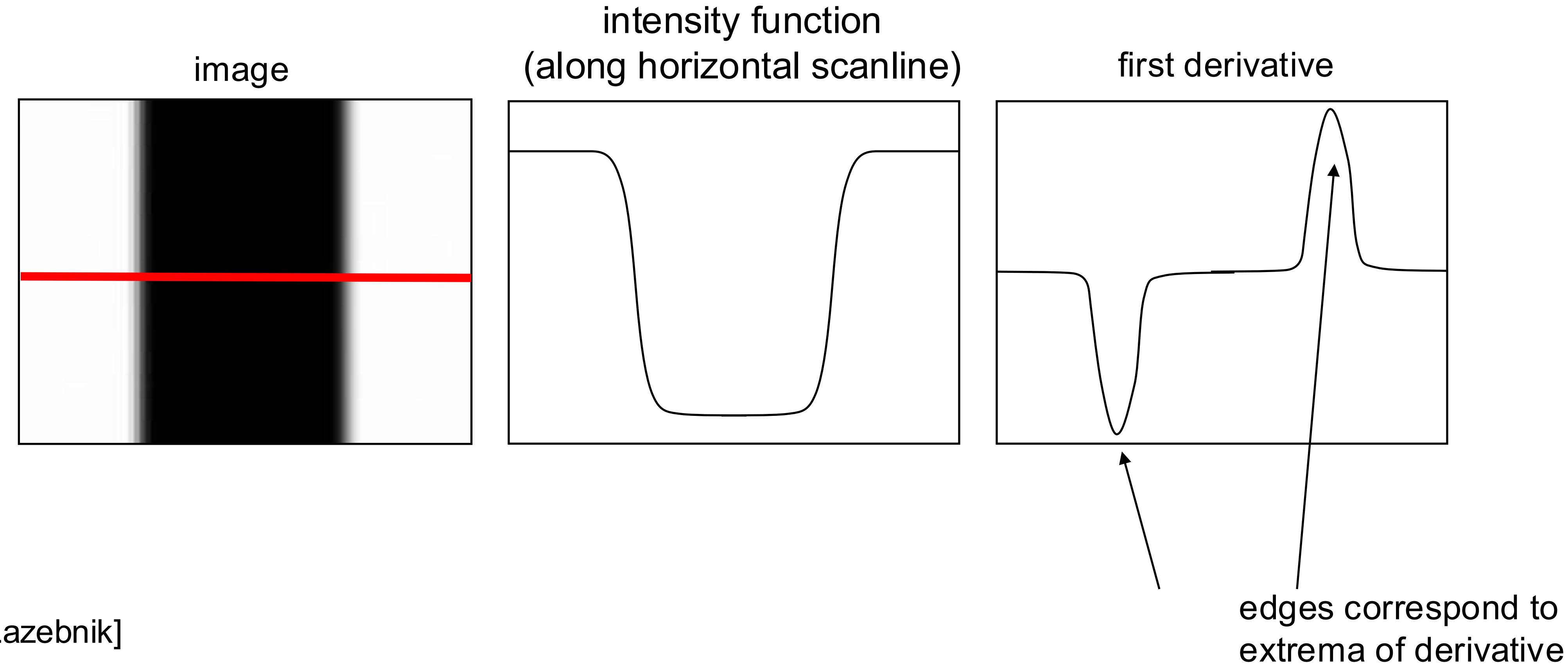


[Source: Steve Seitz]

[Source: Steve Seitz]

Characterizing Edges

- An edge is a place of rapid change in intensity



[Source: S. Lazebnik]

Summary

- Image gradients
- Finite-difference approximation filters
- Gradient magnitude and direction
- Image noise and gradients
- Edges and their importance