# CMPE 362
# Digital Image Processing

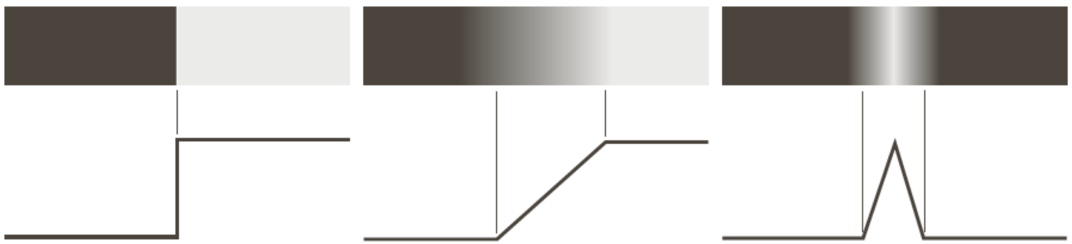
## Edge Detection

# Edge Detection

- Edge detection is the process of finding meaningful transitions in an image.
- The points where sharp changes in the brightness occur typically form the border between different objects or scene parts.
- Further processing of edges into lines, curves and circular arcs result in useful features for matching and recognition.
- Initial stages of mammalian vision systems also involve detection of edges and local features.

Slide credit: Selim Aksoy, Bilkent University

# Edge models



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.

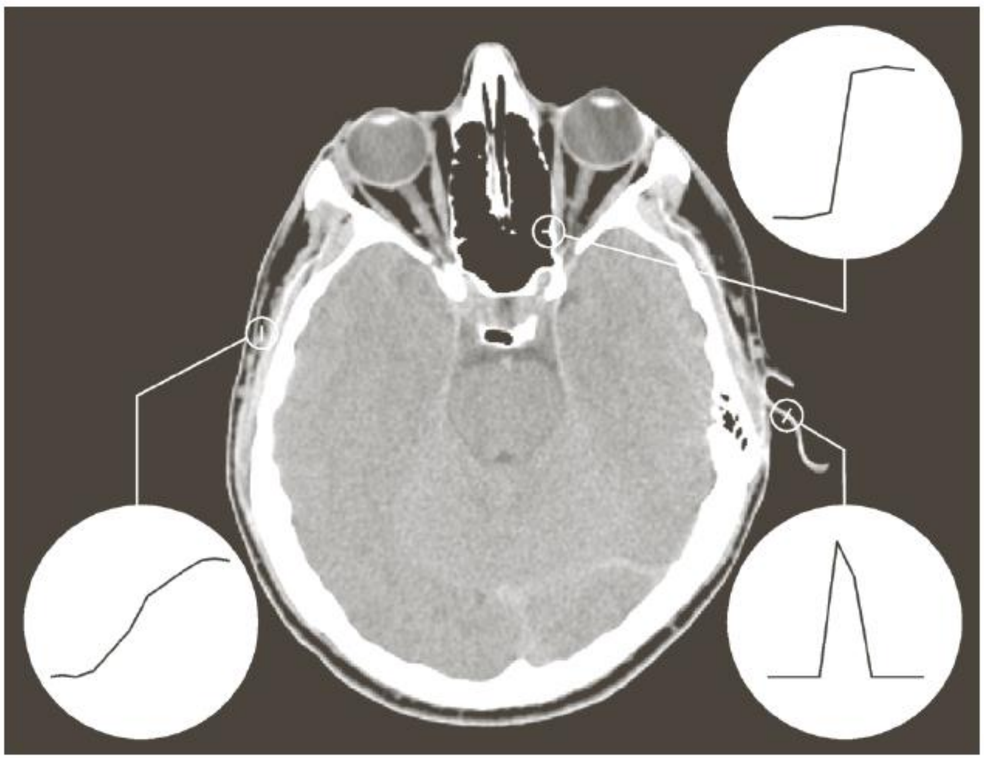


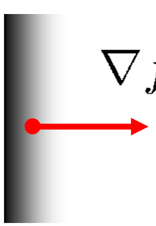
**FIGURE 10.9** A 1508 × 1970 image showing (zoomed) actual ramp (bottom, left), step (top, right), and roof edge profiles. The profiles are from dark to light, in the areas indicated by the short line segments shown in the small circles. The ramp and "step" profiles span 9 pixels and 2 pixels, respectively.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

Slide credit: Gonzales and Woods

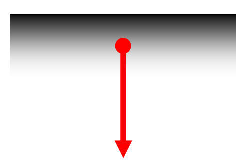
# Difference operators for 2D

- Contrast in the 2D picture function $f(x, y)$ can occur in any direction.
- From calculus, we know that the maximum change occurs along the direction of the *gradient*.
- The gradient of an image $f(x, y)$ at location $(x, y)$ is defined as the vector

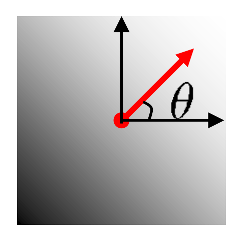
$$\nabla f = \left[ \frac{\partial f}{\partial x} \;\; \frac{\partial f}{\partial y} \right]^T .$$



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



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




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

Slide credit: Selim Aksoy, Bilkent University

# Difference operators for 2D

- The *magnitude of the gradient*

$$|\nabla f| = \left( \left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2 \right)^{1/2}$$

  gives the maximum rate of increase of $f(x, y)$ per unit distance in the direction of $\nabla f$.

- The *direction of the gradient*

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

  represents the direction of this change with respect to the $x$-axis.

# Difference operators for 2D

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

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

Roberts

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

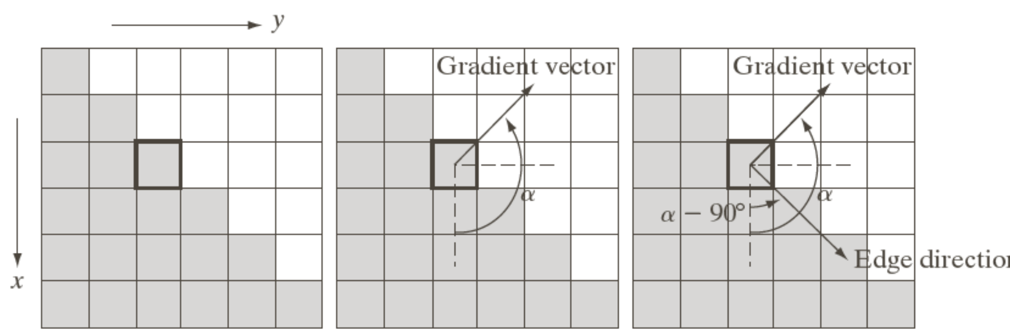
Prewitt

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

Sobel

a
b c
d e
f g

**FIGURE 10.14** A 3 × 3 region of an image (the $z$'s are intensity values) and various masks used to compute the gradient at the point labeled $z_5$.




a b c

**FIGURE 10.12** Using the gradient to determine edge strength and direction at a point. Note that the edge is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square in the figure represents one pixel.

Slide credit: Gonzales and Woods

# Difference operators for 2D

- The *Laplacian* of a 2D function $f(x, y)$ is a second-order derivative defined as

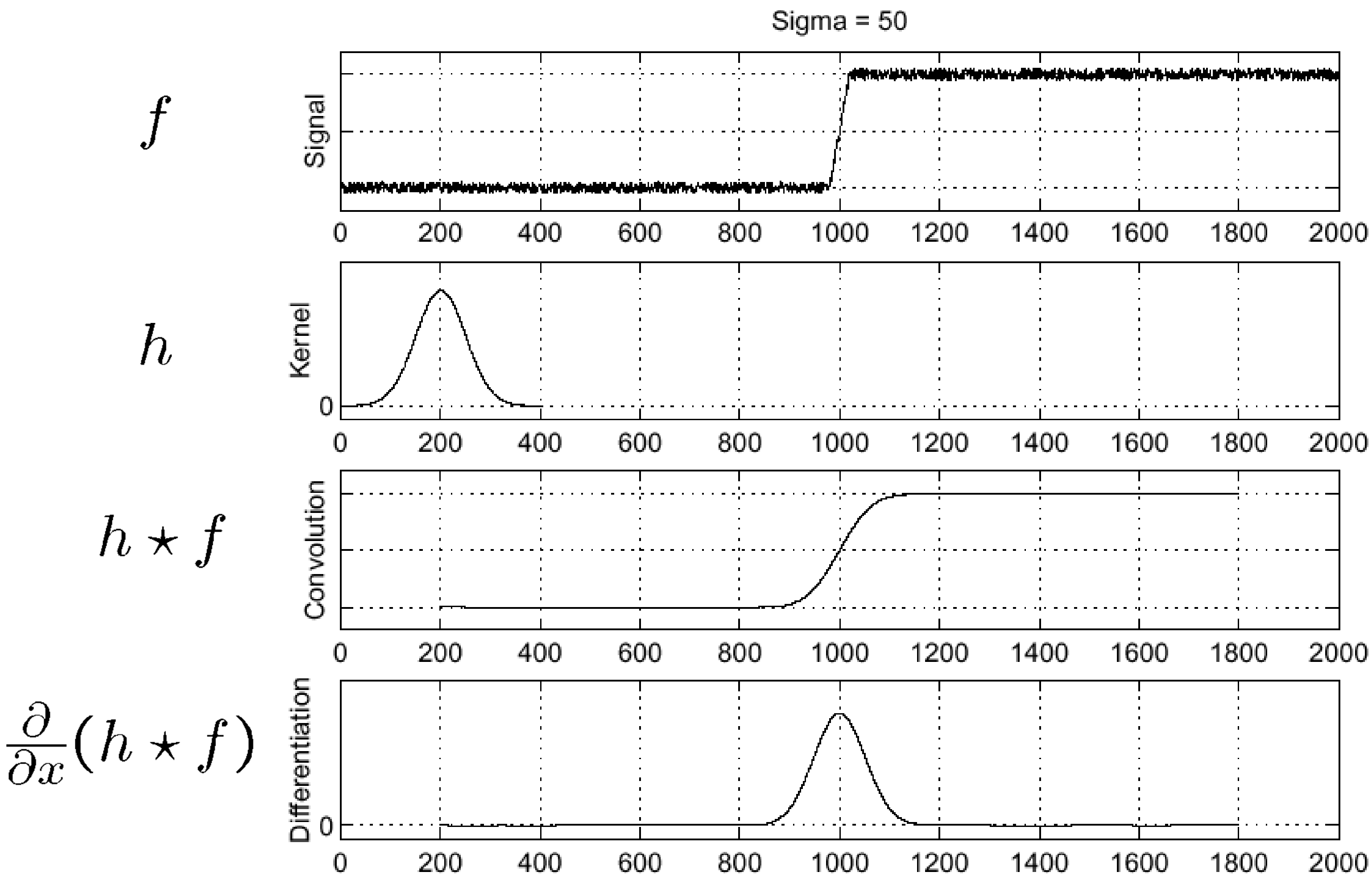
$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}.$$

- The Laplacian generally is not used in its original form for edge detection because:
  - It is sensitive to noise.
  - Its magnitude produces double edges.
  - It is unable to detect edge direction.
- However, its zero-crossing property can be used for edge localization.

Slide credit: Selim Aksoy, Bilkent University

# Difference operators under noise

Solution is to smooth first:




Slide credit: Steve Seitz

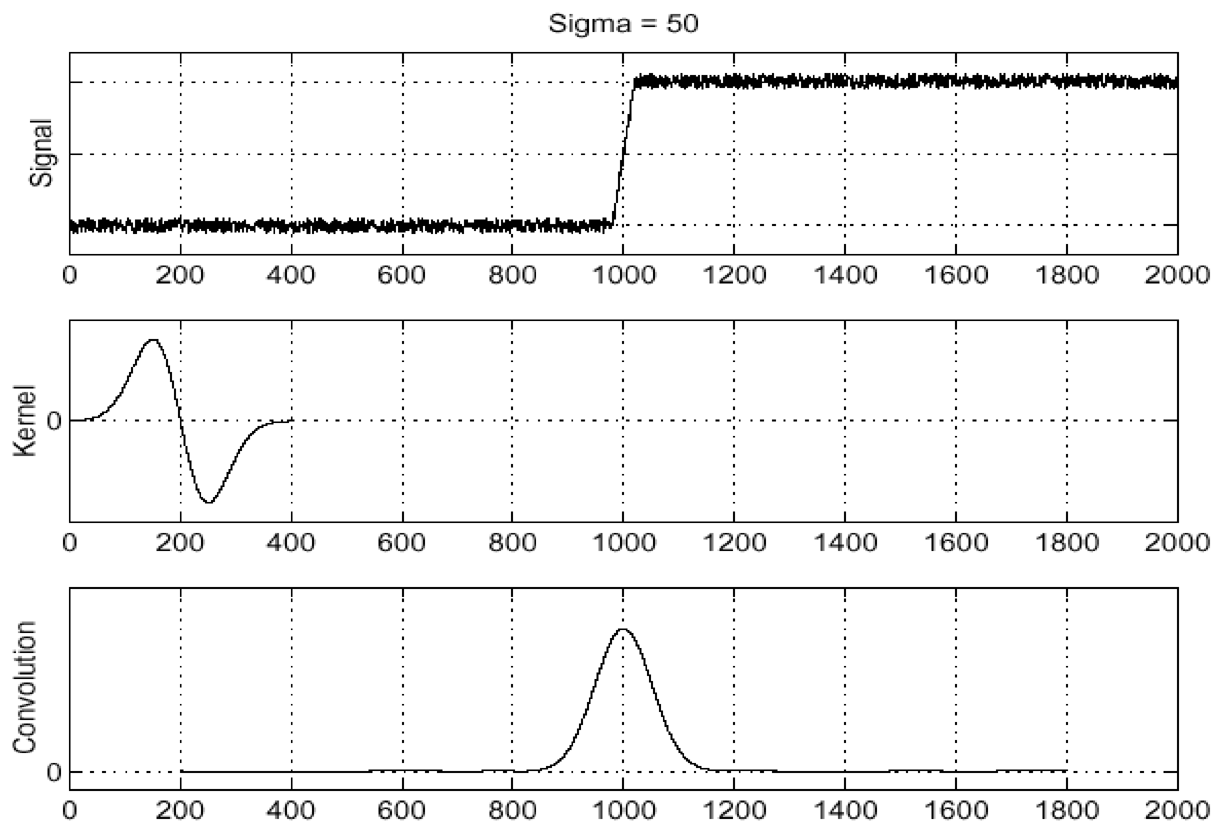
# Difference operators under noise

Differentiation property of convolution: $\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$

$f$

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

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




Slide credit: Steve Seitz

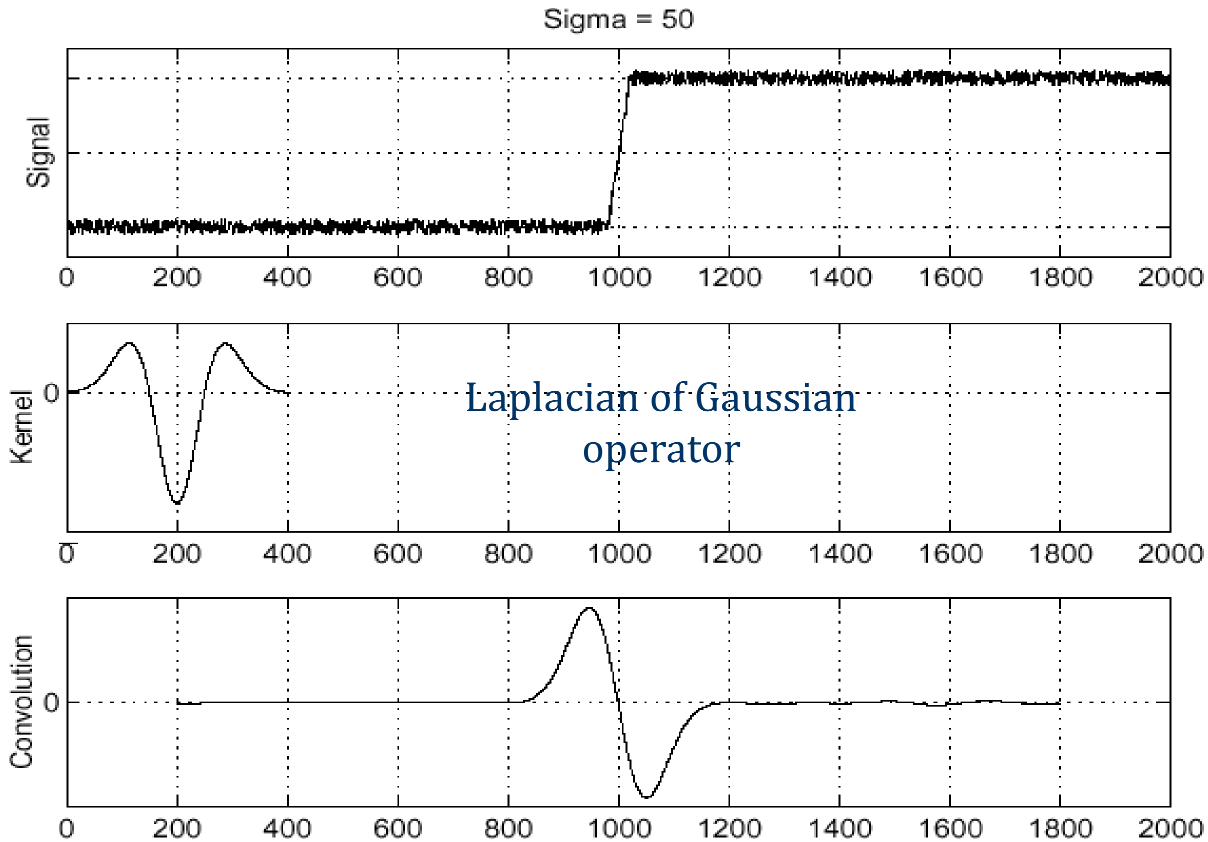
# Difference operators under noise

Consider: $\frac{\partial^2}{\partial x^2}(h \star f)$

$f$

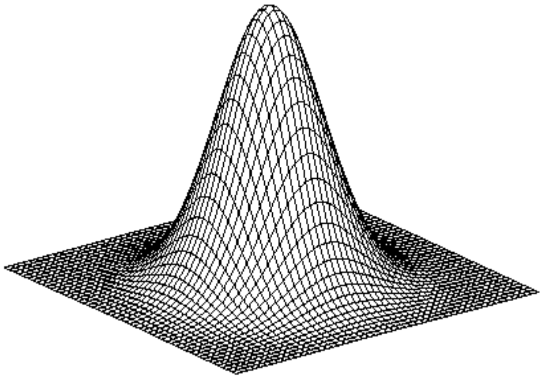
$\frac{\partial^2}{\partial x^2}h$

$(\frac{\partial^2}{\partial x^2}h) \star f$



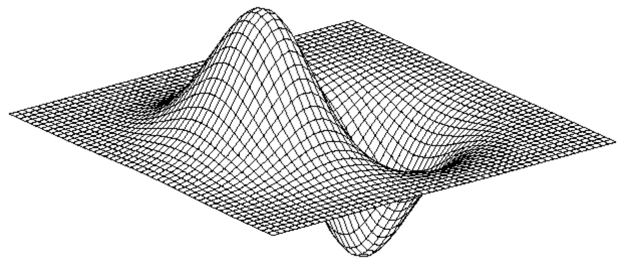

Slide credit: Steve Seitz

# Edge detection filters for 2D



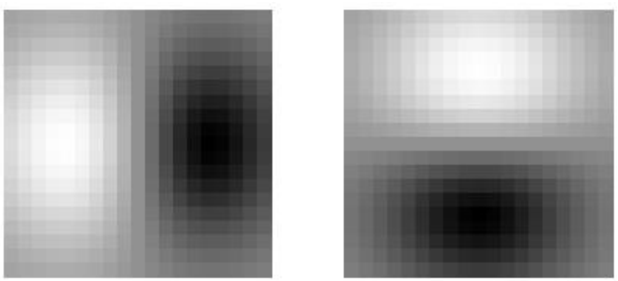
Gaussian

$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$

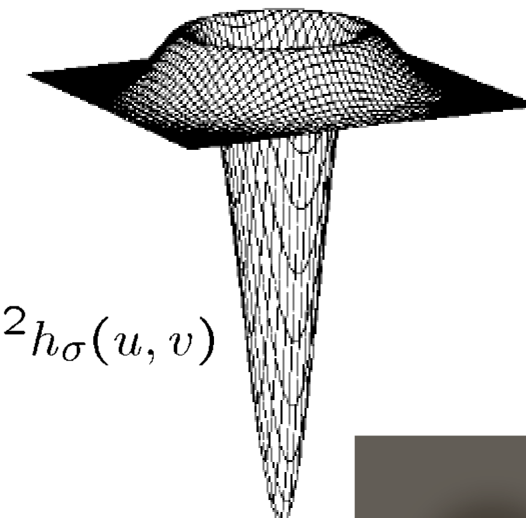


derivative of Gaussian

$$\frac{\partial}{\partial x} h_\sigma(u,v)$$



Laplacian of Gaussian



$$\nabla^2 h_\sigma(u,v)$$

Slide credit: Steve Seitz

# Difference operators for 2D

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

**FIGURE 10.16** (a) Original image of size $834 \times 1114$ 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 mask in Fig. 10.14(f) to filter the image. (c) $|g_y|$, obtained using the mask in Fig. 10.14(g). (d) The gradient image, $|g_x| + |g_y|$.

Slide credit: Gonzales and Woods

# Difference operators for 2D


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

Slide credit: Gonzales and Woods

# Edge detection

- Three fundamental steps in edge detection:
  1. Image smoothing: to reduce the effects of noise.
  2. Detection of edge points: to find all image points that are potential candidates to become edge points.
  3. Edge localization: to select from the candidate edge points only the points that are true members of an edge.

Slide credit: Selim Aksoy, Bilkent University

# Canny edge detector

1. Smooth the image with a Gaussian filter with spread σ.
2. Compute gradient magnitude and direction at each pixel of the smoothed image.
3. Zero out any pixel response less than or equal to the two neighboring pixels on either side of it, along the direction of the gradient (non-maxima suppression).
4. Track high-magnitude contours using thresholding (hysteresis thresholding).

Slide credit: Selim Aksoy, Bilkent University

# Canny edge detector

Original image (Lena)

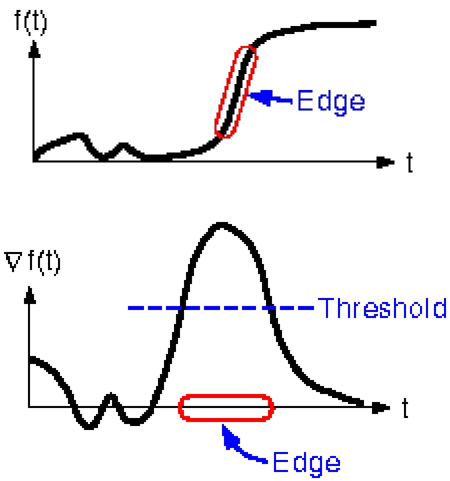
Slide credit: Steve Seitz

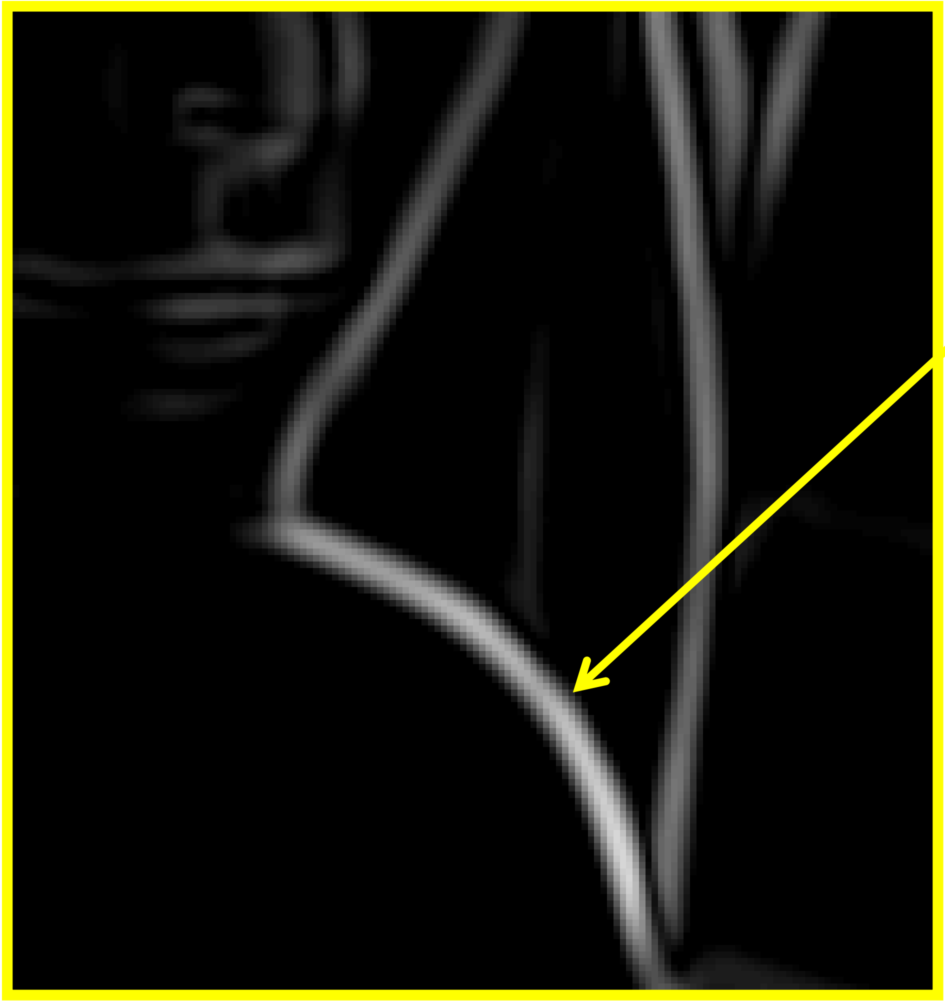
# Canny edge detector

Magnitude of the gradient

Slide credit: Steve Seitz

# Canny edge detector



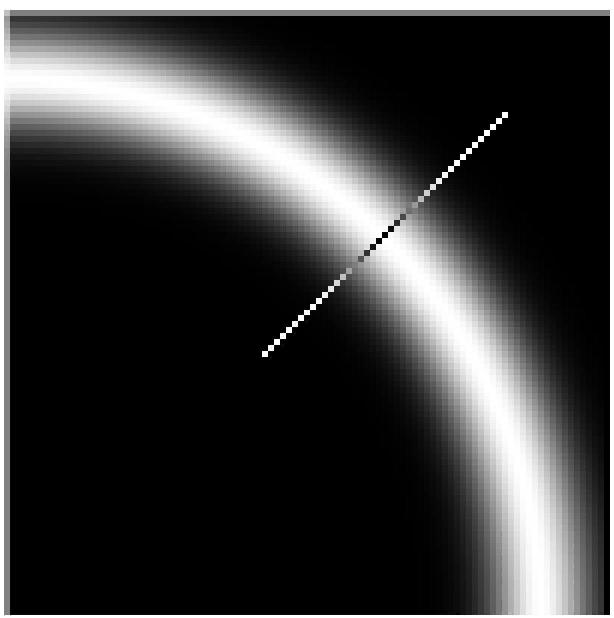



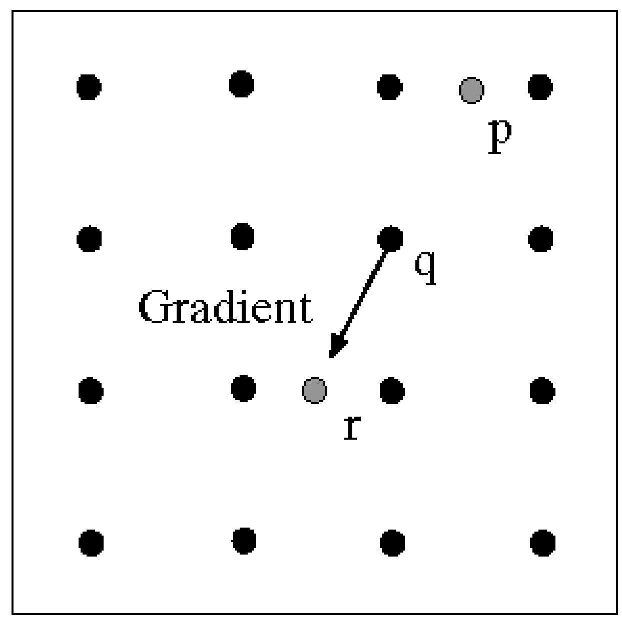
How to turn these thick regions of the gradient into curves?

Slide credit: Steve Seitz

# Canny edge detector

- Non-maxima suppression:
  - Check if pixel is local maximum along gradient direction.
  - Select single max across width of the edge.
  - Requires checking interpolated pixels p and r.
  - This operation can be used with any edge operator when thin boundaries are wanted.

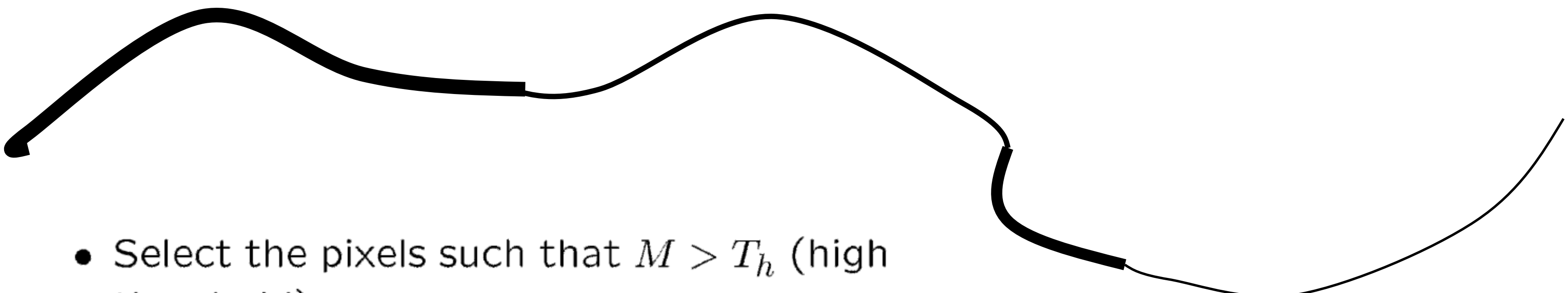





Slide credit: Selim Aksoy, Bilkent University

# Canny edge detector

- Hysteresis thresholding:
  - Use a high threshold to start edge curves, and a low threshold to continue them.



- Select the pixels such that $M > T_h$ (high threshold)
- Collect the pixels such that $M > T_l$ (low threshold) that are neighbors of already collected edge points

Slide credit: Selim Aksoy, Bilkent University

# Canny edge detector

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

**FIGURE 10.25** (a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) Thresholded gradient of smoothed image. (c) Image obtained using the Marr-Hildreth algorithm. (d) Image obtained using the Canny algorithm. Note the significant improvement of the Canny image compared to the other two.

Slide credit: Gonzales and Woods

# Canny edge detector

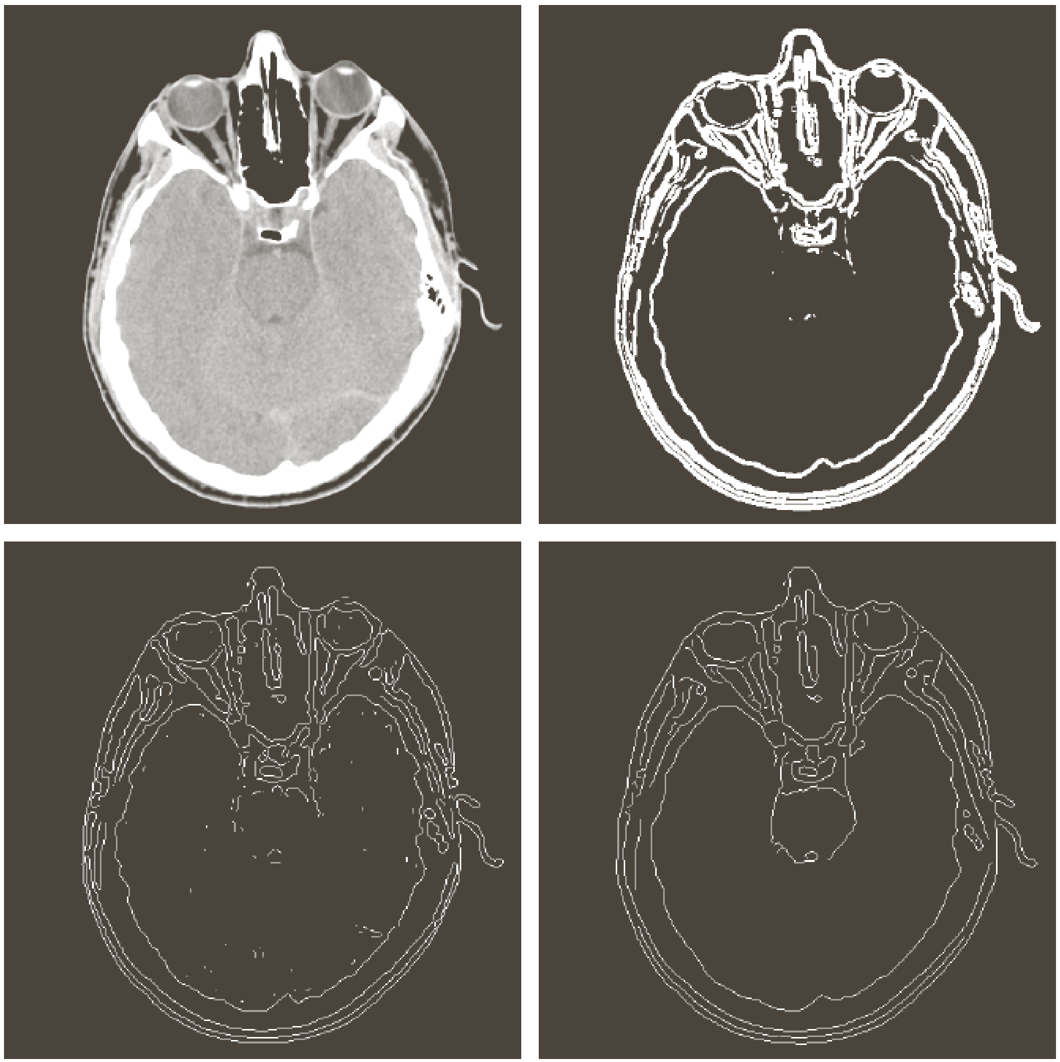


FIGURE 10.26 (a) Original head CT image of size $512 \times 512$ pixels, with intensity values scaled to the range $[0, 1]$. (b) Thresholded gradient of smoothed image. (c) Image obtained using the Marr-Hildreth algorithm. (d) Image obtained using the Canny algorithm. (Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)

Slide credit: Gonzales and Woods

# Canny edge detector

- The Canny operator gives single-pixel-wide images with good continuation between adjacent pixels.
- It is very sensitive to its parameters, which need to be adjusted for different application domains.

Slide credit: Selim Aksoy, Bilkent University

# Week 10 – Hands on Activity

- Prepare and submit a Jupyter Notebook file containing the code and the results for the following Task

# Task

- Read an image of your choice.
- Convert the image into gray scale image and display it.
- Detect edges in the image using Canny edge detector.
- Try two different sets of parameters where threshold1 varies only.