CARDIFF UNIVERSITY
PRIFYSGOL CAERDYD

CMT107 Visual Computing

Edge Detection

Xianfang Sun, Jing Wu

School of Computer Science and Informatics

Cardiff University

Overview

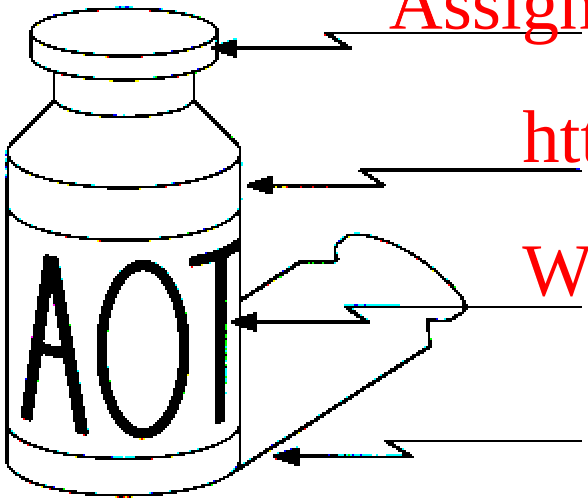
- Origin of Edges
- Characterising Edges
- Derivatives with Convolution
  - Finite Difference Filters
  - Image Gradient
- Canny Edge Detector

Edge Detection

- Goal: identify sudden changes (discontinuities) in an image
  - Intuitively, most semantic and shape information from the image can be encoded in the edges
  - More compact than pixels
- Ideal: artist's line drawing (but artists are also using object-level knowledge)

Origin of Edges

- Edges are caused by a variety of factors



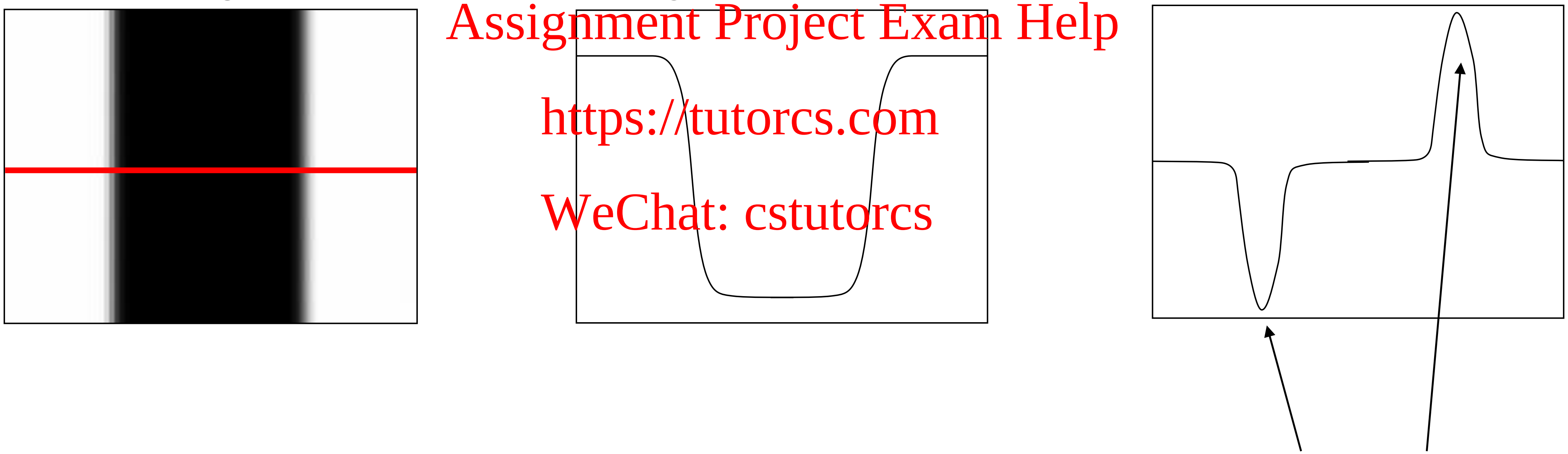
Characterising Edges

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

image

intensity function
(along horizontal scanline)

first derivative



edges correspond to
extrema of derivative

Derivatives with Convolution

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

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

- For discrete data, we can approximate using finite differences:

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

- To implement the above as convolution, what would be the associated filter?

-1	**1**

Partial Derivatives of an image

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

-1	1

(or

1	-1

)

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

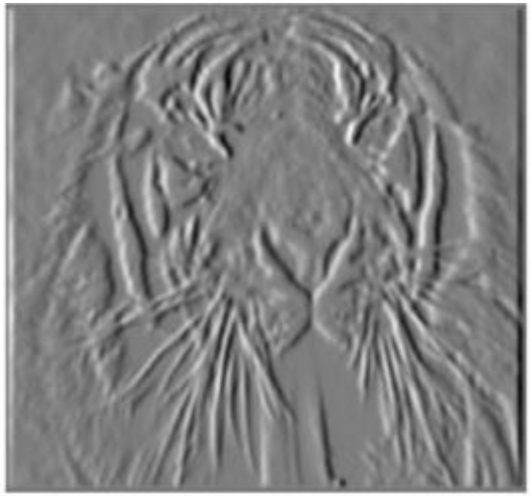
-1
1

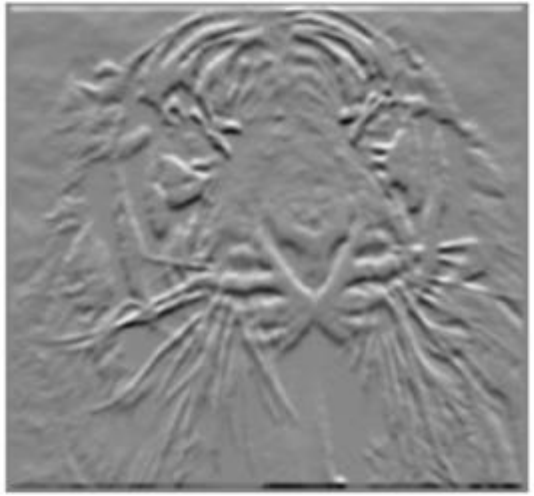
(or

1
-1

)

Intensity normalized to [0,255]





Can you tell which shows changes with respect to x?

Finite Difference Filters

- Other approximations of derivative filters:

Prewitt: $M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$, $M_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$

Sobel: $M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$, $M_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

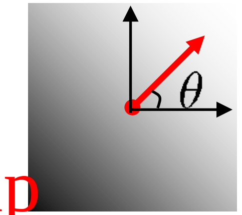
Roberts: $M_x = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, $M_y = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

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 in intensity

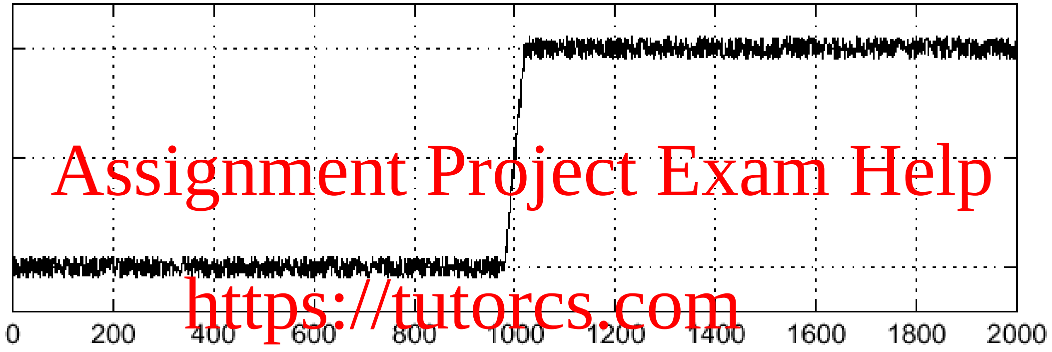
- The gradient direction is given by $\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$

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

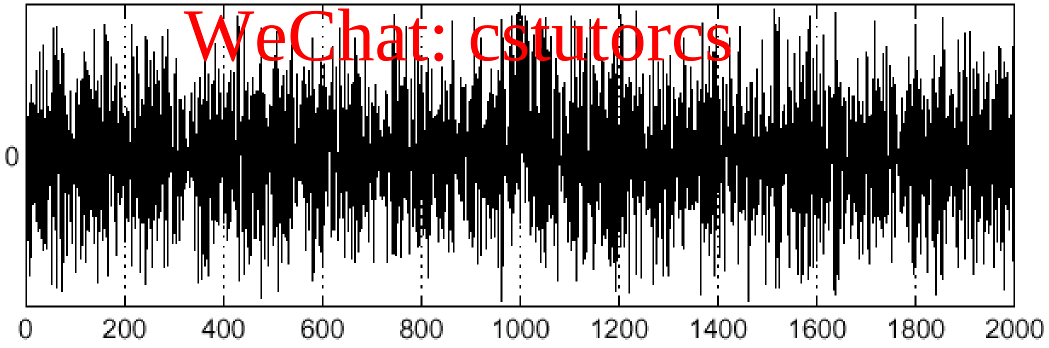
Effects of Noise

- Consider a single row or column of the image, and plot the intensity as a function of position

$f(x)$



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



Where is the edge?

Effects of Noise

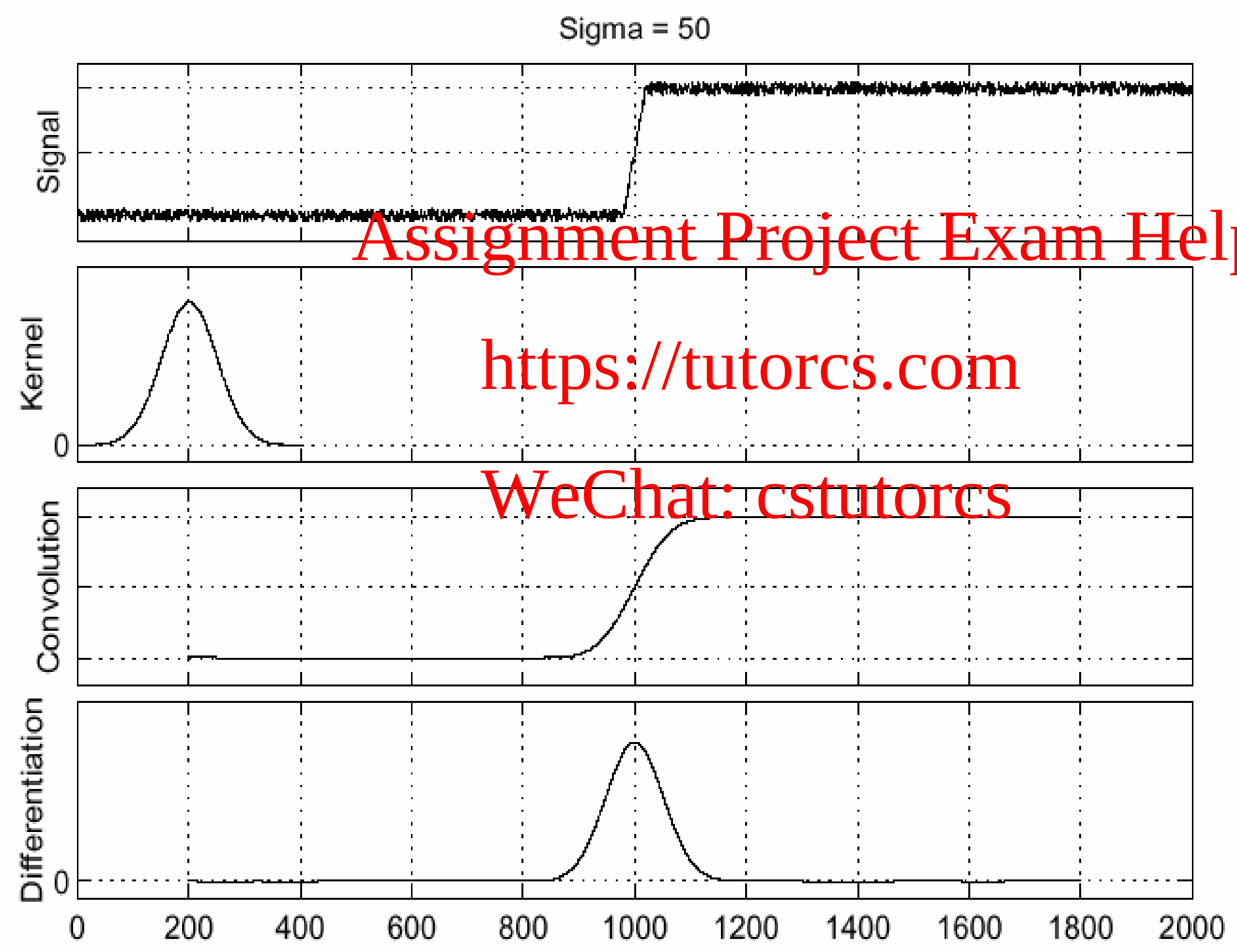
- Solution: smooth first

f

g

$f * g$

$\frac{\partial}{\partial x} f * g$



To find edges, look for peaks in $\frac{\partial}{\partial x} f * g$

Derivative Theorem of Convolution

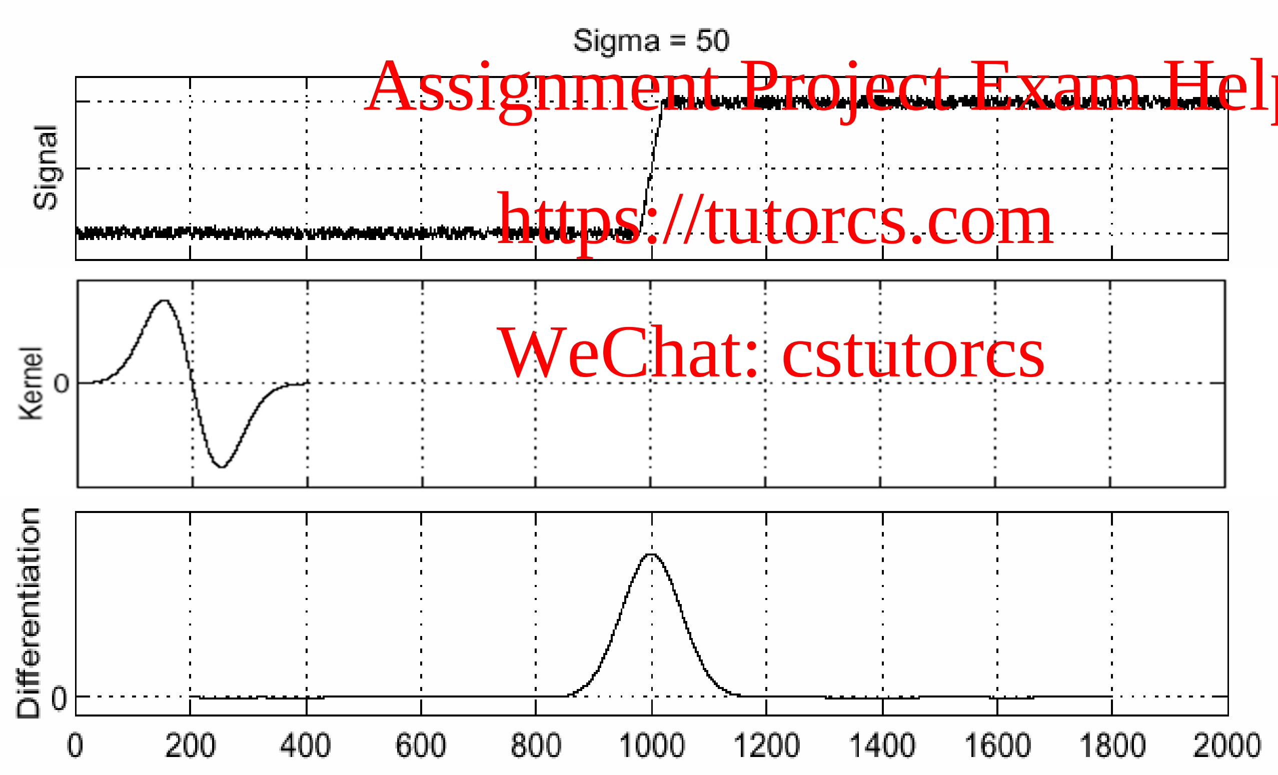
- Differentiation is convolution, and convolution is associative:

$$\frac{\partial}{\partial x}(f * g) = f * \frac{\partial}{\partial x} g$$

f

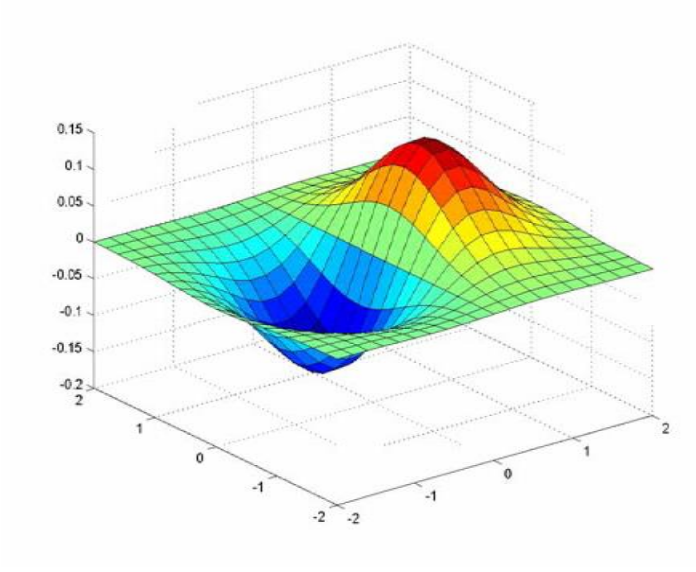
$\frac{\partial}{\partial x} g$

$f * \frac{\partial}{\partial x} g$

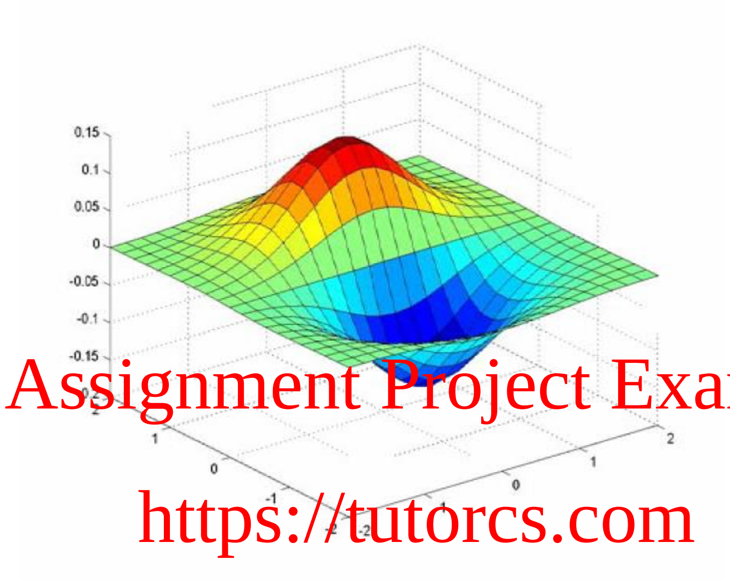


This saves us one operation

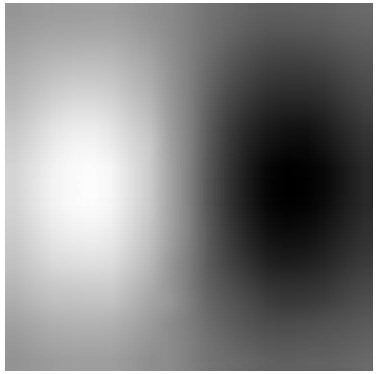
Derivative of Gaussian filter

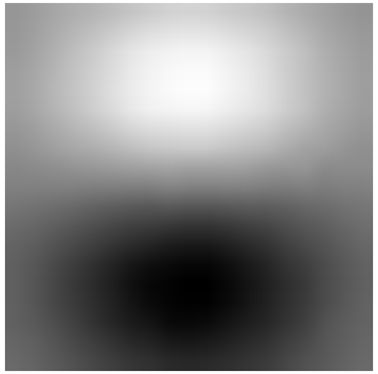


x direction



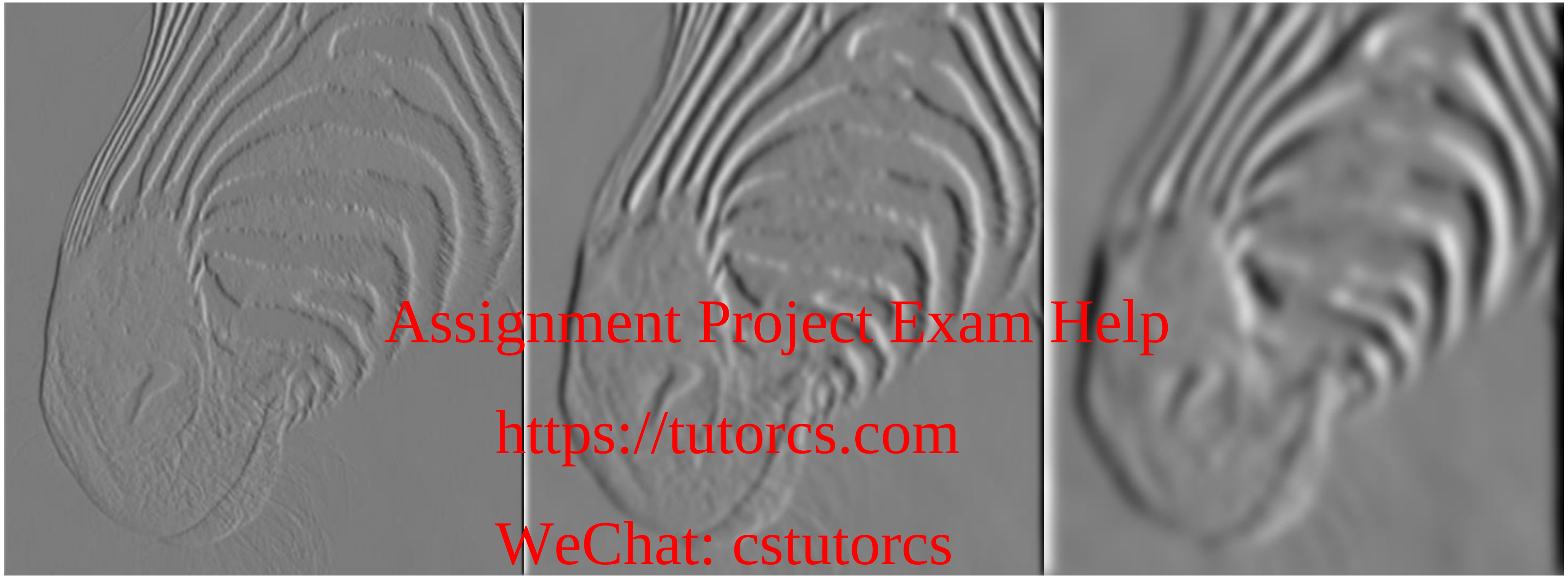
y direction





- Which finds horizontal / vertical edges?
- Are these filters separable?

Scale of Gaussian Derivative Filter

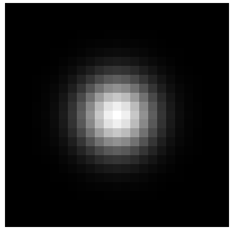


1 pixel | 3 pixels | 7 pixels

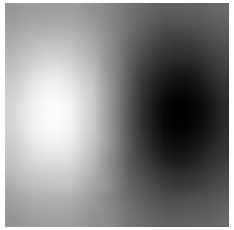
- Smoothed derivative removes noise, but blurs edge. Also find edges at different “scales”.

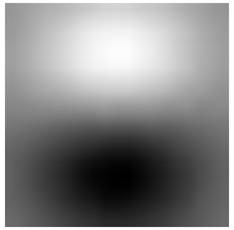
Review: Smoothing vs. Derivative Filters

- Smoothing filters
  - Gaussian: removes "high-frequency" components; "low-pass" filter
  - Can the values of a smoothing filter be negative?
  - What should the values sum to?
    - One: constant regions are not affected by the filter



- Derivative filters
  - Derivatives of Gaussian
  - Can the values of a derivative filter be negative?
  - What should the values sum to?
    - Zero: no response in constant regions
  - High absolute value at points of high contrast





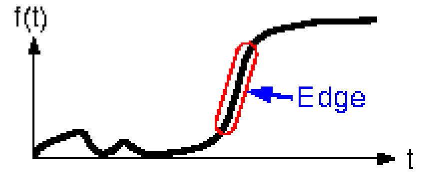
Canny Edge Detector

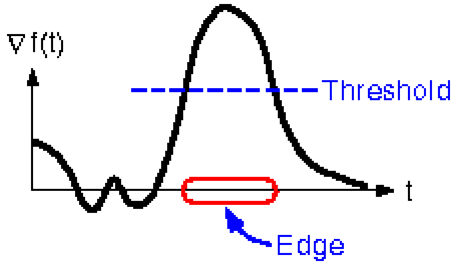
Original image

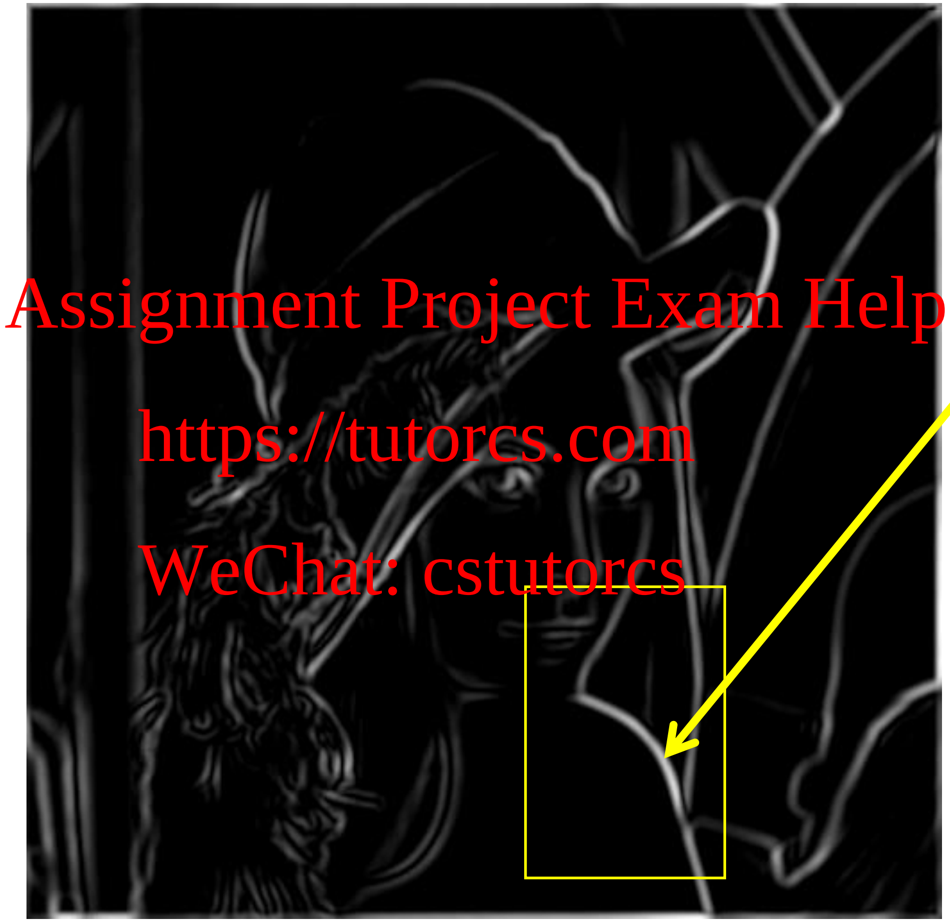
Canny Edge Detector

Norm of the gradient

Canny Edge Detector





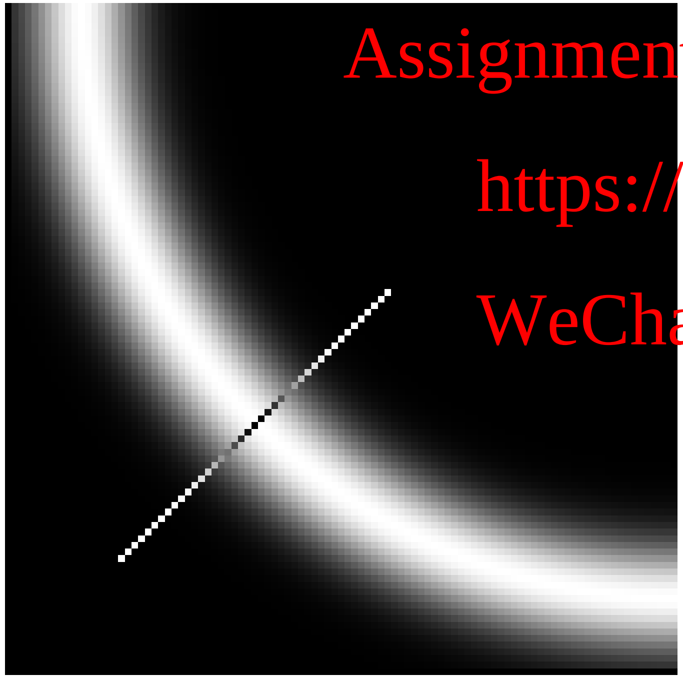


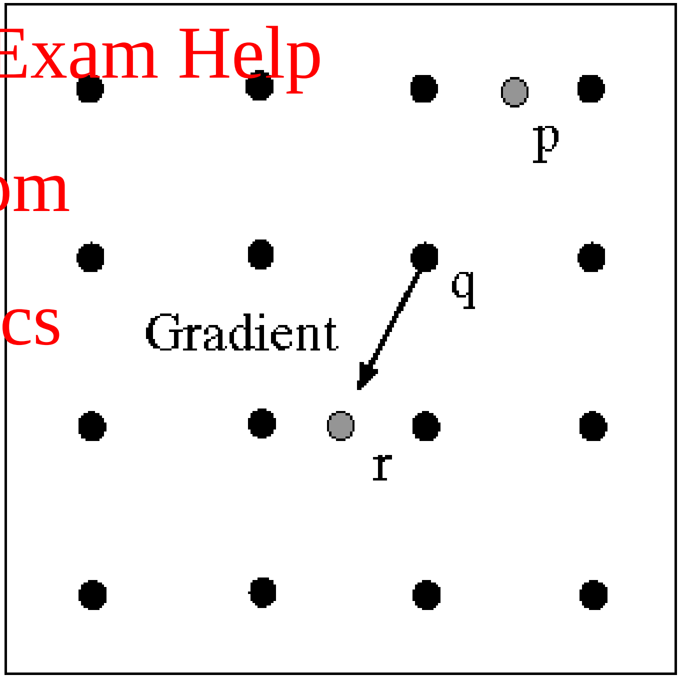
Thresholding

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

Non-maximum 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





Canny Edge Detector

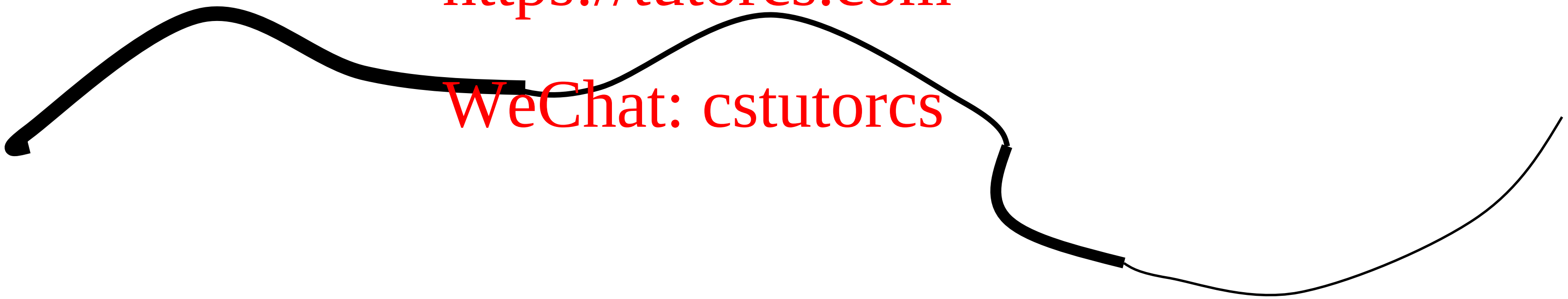
Problem: pixels along this edge didn't survive the thresholding

Thinning
(non-maximum suppression)

Hysteresis Thresholding

- Use a high threshold to start edge curves,

  and a low threshold to continue them



Hysteresis Thresholding

Original image

high threshold
(strong edges)

low threshold
(weak edges)

hysteresis threshold

Summary of Canny Edge Detector

1. Filter image with derivative of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression:
   - Thin wide “ridges” down to one pixel width
4. Linking and thresholding (hysteresis):
   - Define two thresholds: low and high
   - Use the high threshold to start edge curves and low threshold to continue them

J. Canny, *A Computational Approach To Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.

Summary

- What is edge detection?
- Describe different origin of edges.
- How to characterise edges?
- How to calculate image gradient using Prewitt, Sobel, or Roberts filters?
- Describe the steps of Canny edge detector.