University of Nottingham
UK | CHINA | MALAYSIA

COMP-2032

Introduction to Image Processing

Lecture 6

Derivative and Edges

Learning Outcomes

IDENTIFY

1. Derivative Filters
2. Sharpening
3. What is Edge Detection?
4. Edge Detection using 1st Derivatives
5. Edge Detection using 2nd Derivatives
6. The Canny Operator

University of Nottingham
UK | CHINA | MALAYSIA

Derivative Filters

In 1 Dimension

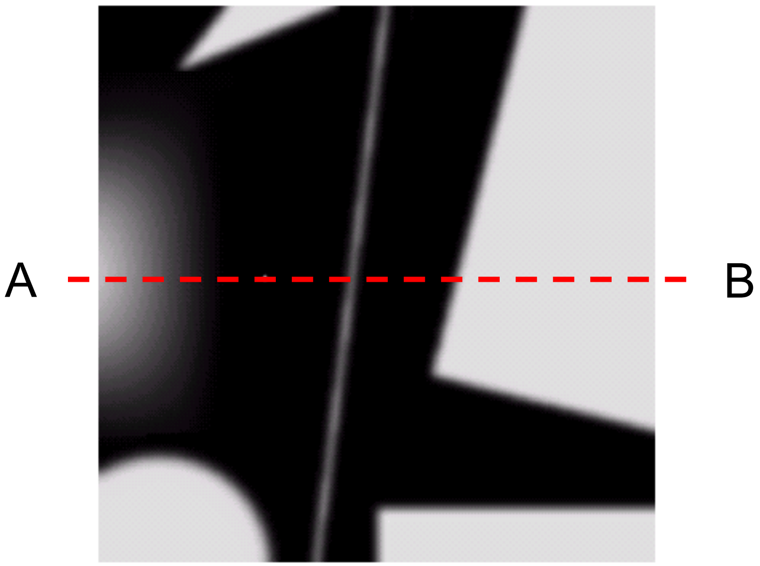
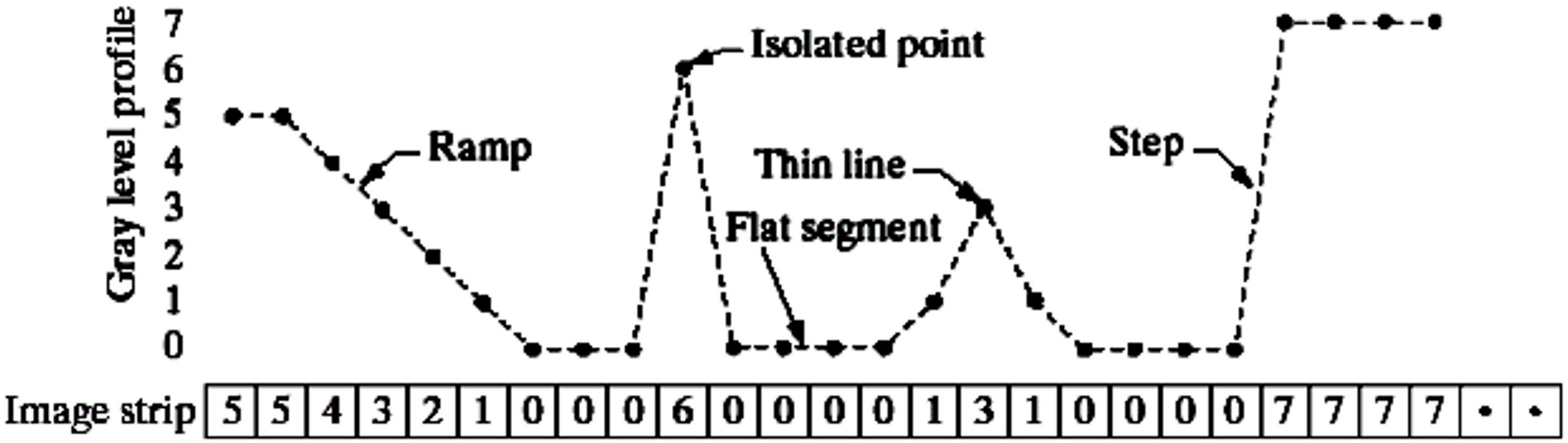


Image features are often characterised by changes in intensity



ACK: Prof. Tony Pridmore, UNUK

1st Derivative

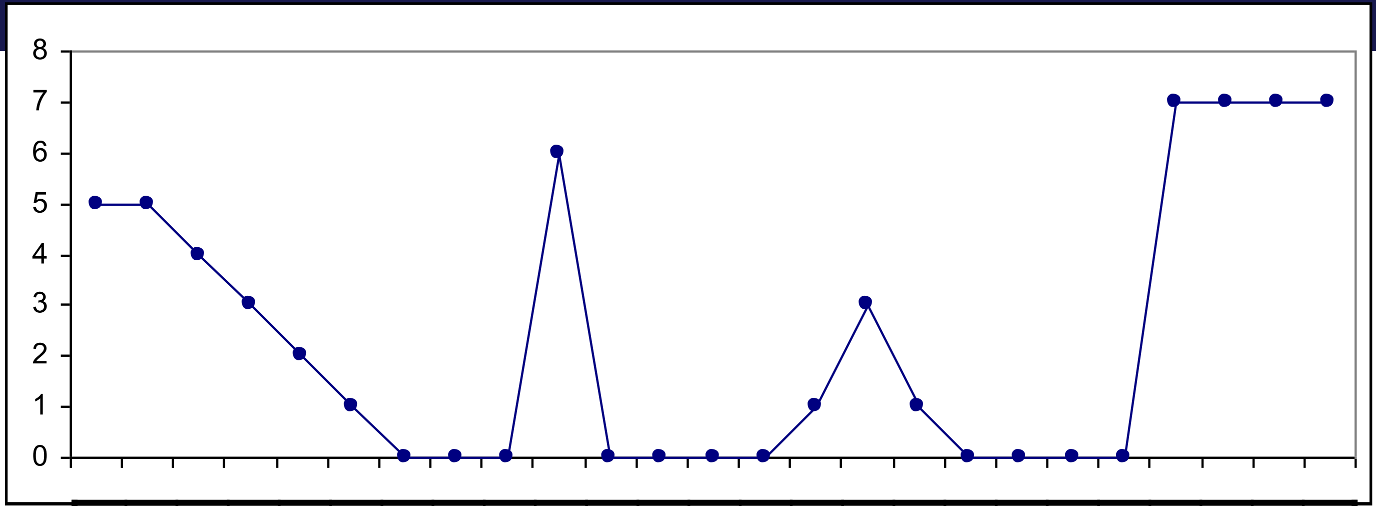
The 1st derivative of a function can be approximated by:

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

The difference between neighbouring values and measures the rate of change of the function

ACK: Prof. Tony Pridmore, UNUK

1st Derivative

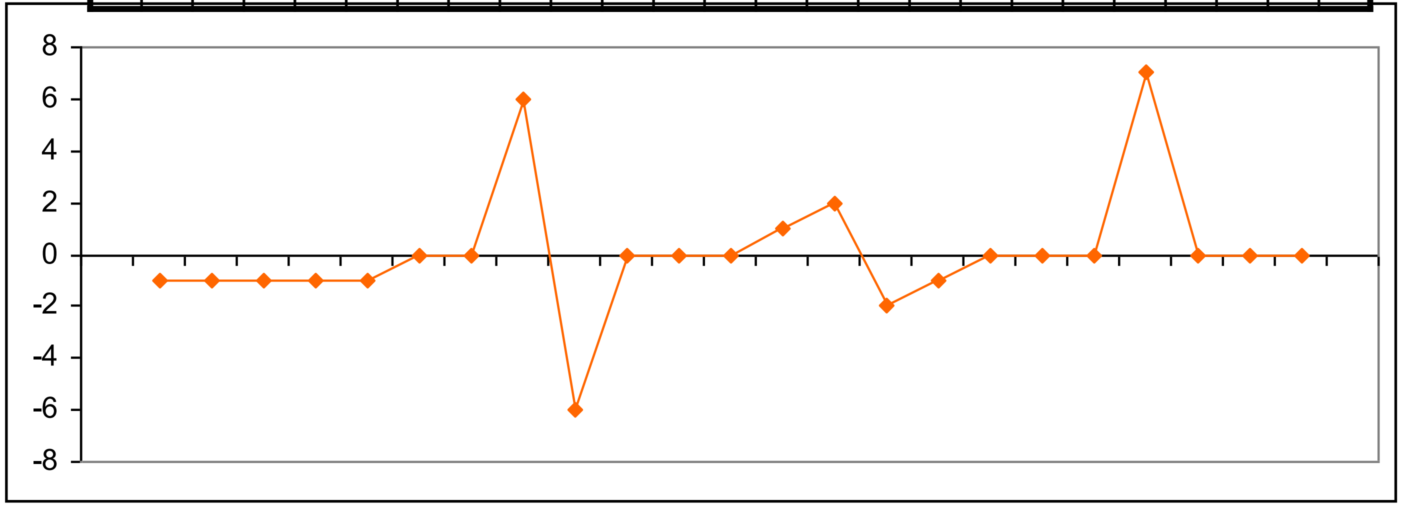


Raw data

5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7

1st Derivative

	0	-1	-1	-1	-1	-1	0	0	6	-6	0	0	0	1	2	-2	-1	0	0	0	7	0	0	0



ACK: Prof. Tony Pridmore, UNUK

2^{nd} Derivative

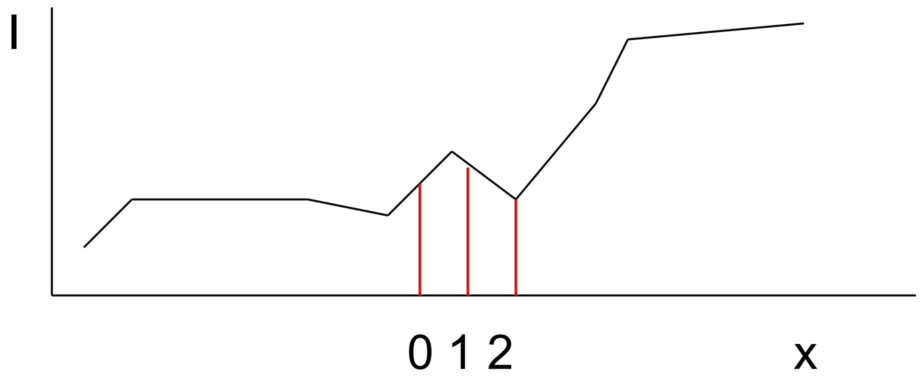
The formula for the 2^{nd} derivative of a function is:

$$\frac{\partial^2 f}{\partial^2 x} = f(x+1) + f(x-1) - 2f(x)$$

- Simply takes into account the values of both – before and after the current value
- Derived by estimating the 1^{st} derivative at x + 0.5 and x – 0.5 and computing the derivative of the resulting data

ACK: Prof. Tony Pridmore, UNUK

2^{nd} Derivative



$I''(1) = (I'(1.5) - I'(0.5))/1$

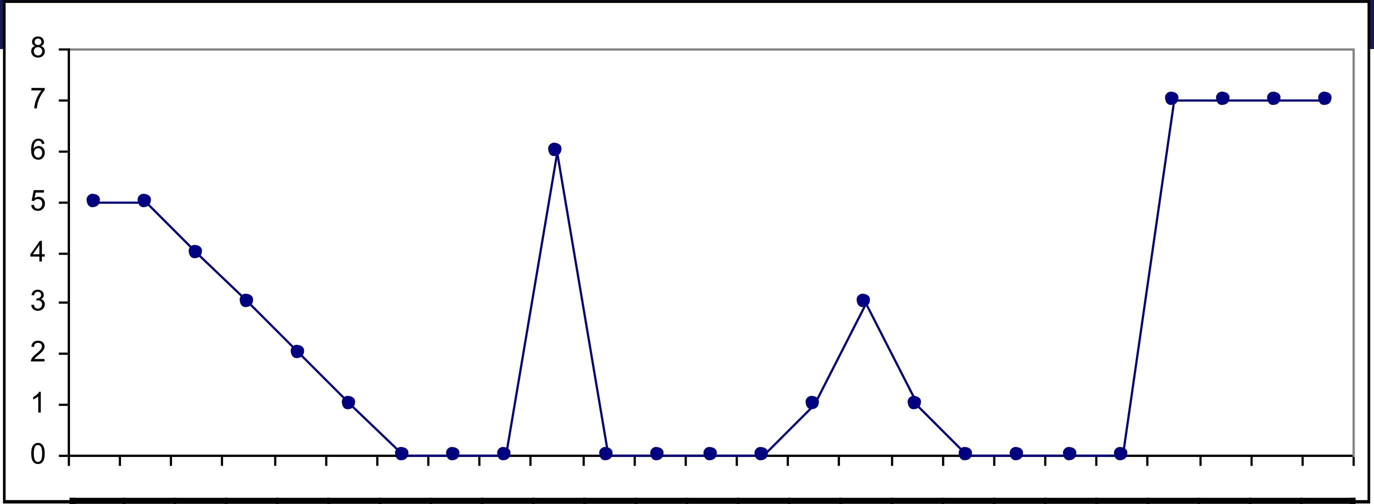
$I'(0.5) = (I(1) - I(0))/1$ and $I'(1.5) = (I(2) - I(1))/1$

$\therefore\ I''(I) = 1.I(0) - 2.I(1) + 1.I(2)$

1	-2	1

ACK: Prof. Tony Pridmore, UNUK

2nd Derivative

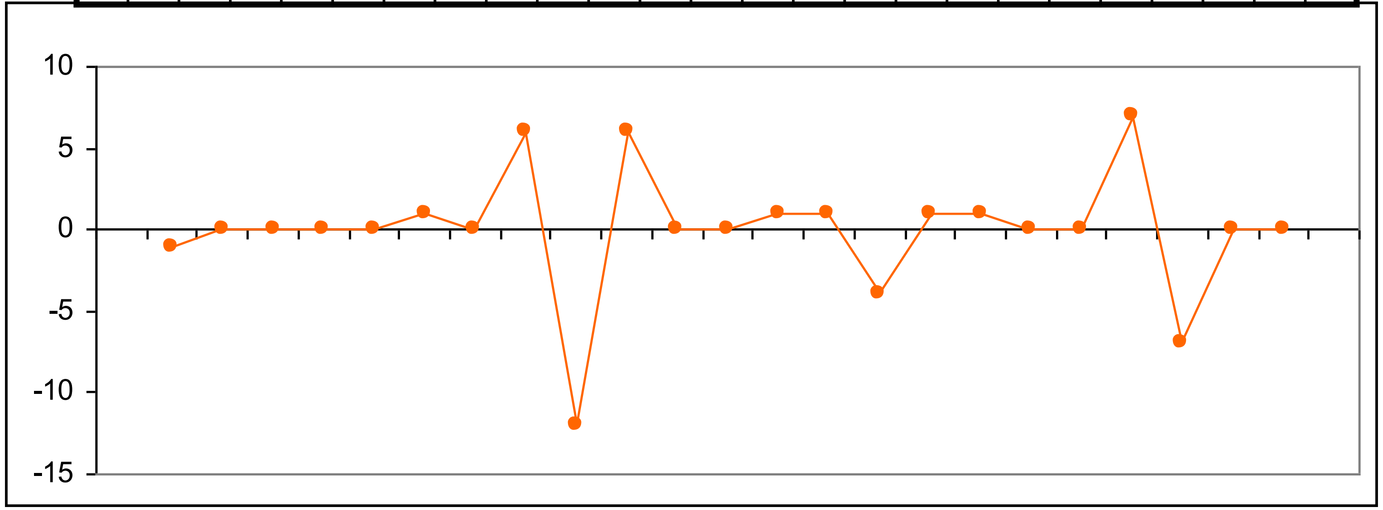


Raw data

5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7

2nd Derivative

	-1	0	0	0	0	1	0	6	-12	6	0	0	1	1	-4	1	1	0	0	7	-7	0	0	



ACK: Prof. Tony Pridmore, UNUK

Derivatives in 2 Dimension

- 2^{nd} derivatives generalise to 2D quite easily, implementing a 1^{st} derivative in 2D is a little more complex

For a function f(x,y) the gradient of f at coordinates (x,y) is given as a column vector:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

REMEMBER

- Computation of the 1^{st} derivative can't be done by convolution alone

ACK: Prof. Tony Pridmore, UNUK

1st Derivative Filtering

The magnitude of the 1st derivative vector is

$$\nabla f = mag(\nabla \mathrm{f})$$

$$= \left[G_x^2 + G_y^2\right]^{1/2}$$

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

which can be simplified to

$$\nabla f \approx |G_x| + |G_y|$$

ACK: Prof. Tony Pridmore, UNUK

1st Derivative Filters

Many 1st derivatives filters have been proposed

Roberts' Cross Operators

G_x:

1	0
0	-1

G_y:

0	1
-1	0

Sobel Operators

G_x:

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

G_y:

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

These operators are most commonly associated with edge detection

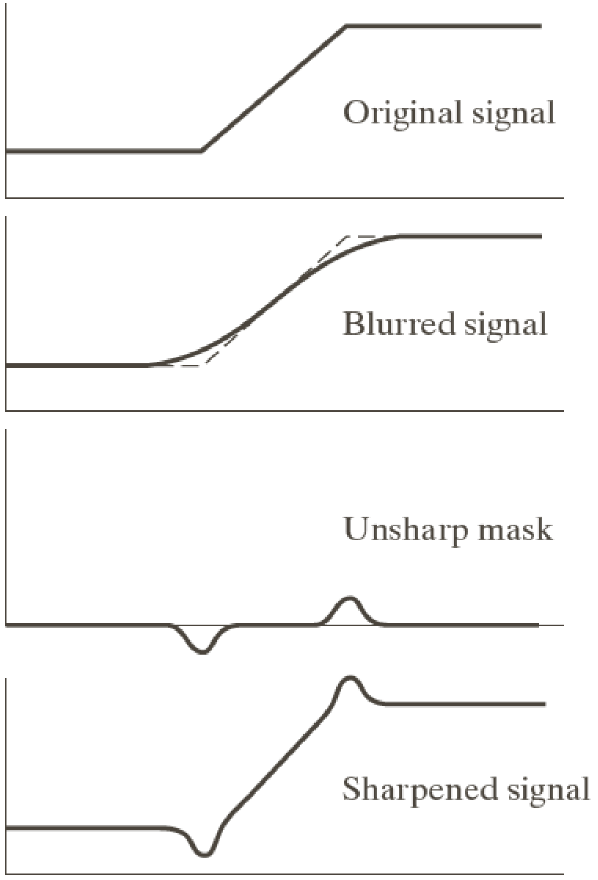
ACK: Prof. Tony Pridmore, UNUK

Image Sharpening

Edge Enhancement: Unsharp Masking

- Edges are important
- Sometimes we want to enhance them without (much) affecting the rest of the image

1. Take the original image
2. Gaussian smooth it
3. Subtract the smoothed version from the original to make an *unsharp mask*
4. Add the mask to the original to make the edge appear more obvious

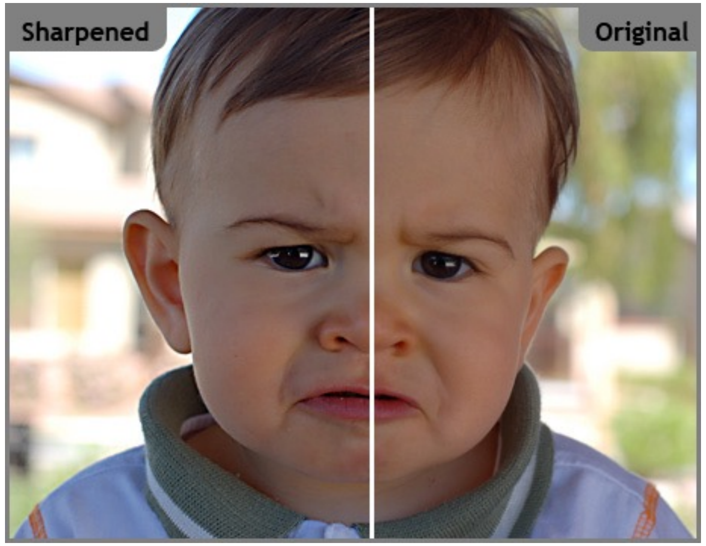


ACK: Prof. Tony Pridmore, UNUK

Unsharp Masking

Makes edges noticeable sharper

- Even if they are noise
- Sometimes too much



ACK: Prof. Tony Pridmore, UNUK

Derivative Filters

Unsharp filtering enhances edges by comparing the original with a smoothed image

- Relies on the smoothing effect of a Gaussian function introducing a difference between original and processed images
- Parameterised by σ
- Simple, but effect is hard to predict, so hard to parameterise

A more direct way to highlight edges and other features associated with high image gradients is to estimate derivatives…

ACK: Prof. Tony Pridmore, UNUK

Image Sharpening with Derivatives

The 2nd derivatives is more useful for image enhancement than the 1st derivative

- Stronger response to fine detail
- Simpler implementation

The most common sharpening filter is the *Laplacian*

- Isotropic
- One of the simplest sharpening filters
- Straightforward digital implementation via convolution

ACK: Prof. Tony Pridmore, UNUK

The Laplacian

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

$$\frac{\partial^2 f}{\partial^2 x} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{\partial^2 f}{\partial^2 y} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

ACK: Prof. Tony Pridmore, UNUK

The Laplacian

$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x, y)$$

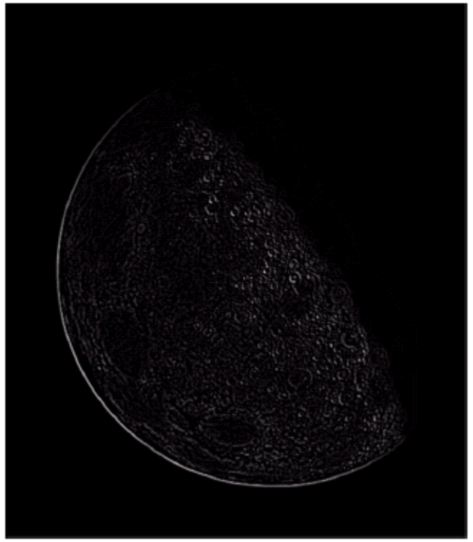
0	1	0
1	-4	1
0	1	0

ACK: Prof. Tony Pridmore, UNUK

The Laplacian

Highlights edges and other discontinuities

Original Image



Laplacian Filtered Image

Laplacian Filtered Image Scaled for Display

ACK: Prof. Tony Pridmore, UNUK

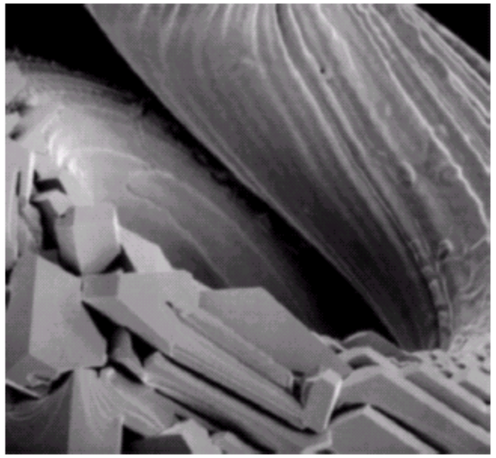
A Single Enhancement Operator

$$
\begin{aligned}
g(x,y) = f(x,y) - \nabla^2 f \\
&= f(x,y) - [f(x+1,y) + f(x-1,y) \\
&\qquad + f(x,y+1) + f(x,y-1) \\
&\qquad - 4f(x,y)] \\
&= 5f(x,y) - f(x+1,y) - f(x-1,y) \\
&\qquad - f(x,y+1) - f(x,y-1)
\end{aligned}
$$

ACK: Prof. Tony Pridmore, UNUK

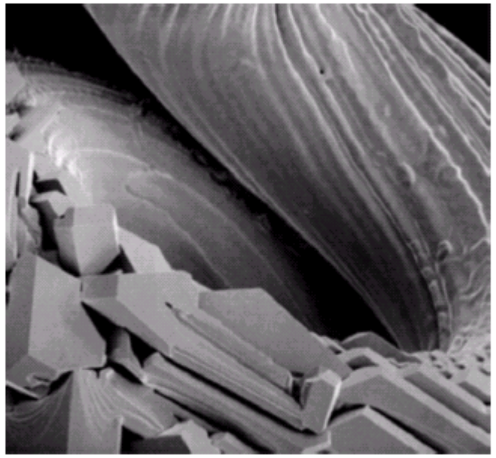
A Single Operator

Convolution with this operator performs image sharpening in a single step



Input

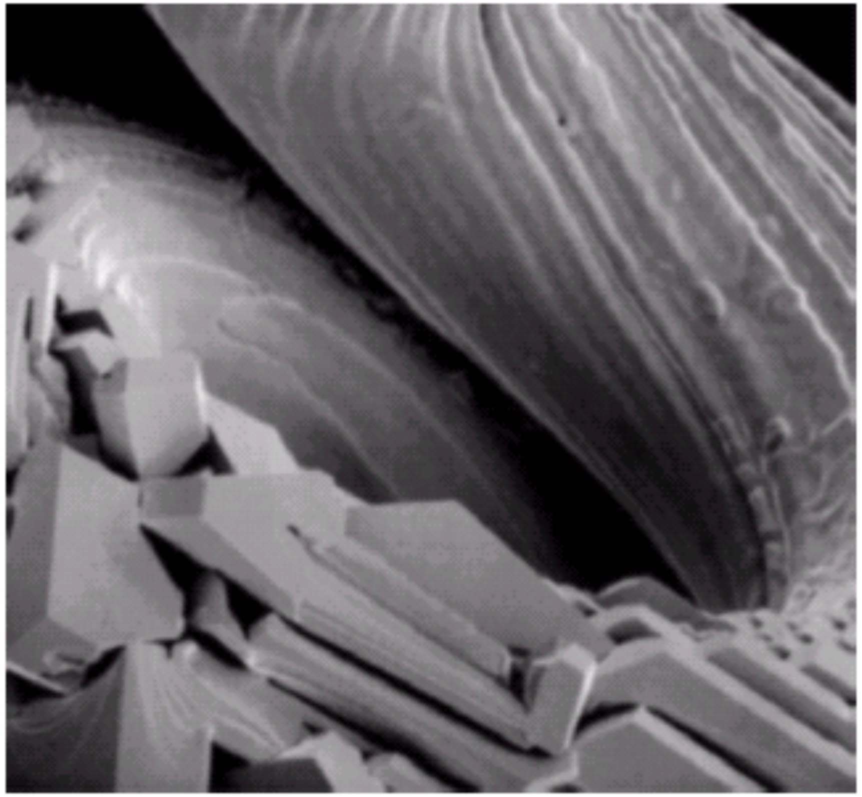
0	-1	0
-1	5	-1
0	-1	0

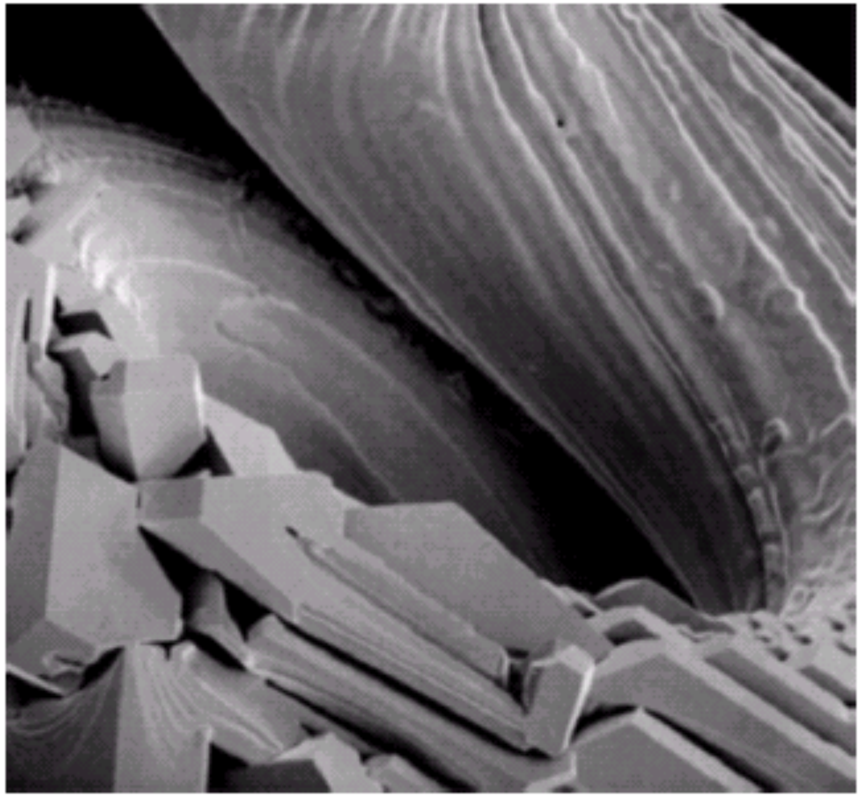


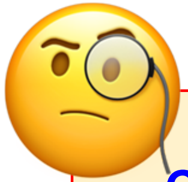
Sharpen

ACK: Prof. Tony Pridmore, UNUK

A Single Operator







Let's take a **CLOSER** look

ACK: Prof. Tony Pridmore, UNUK

Variations on the Theme

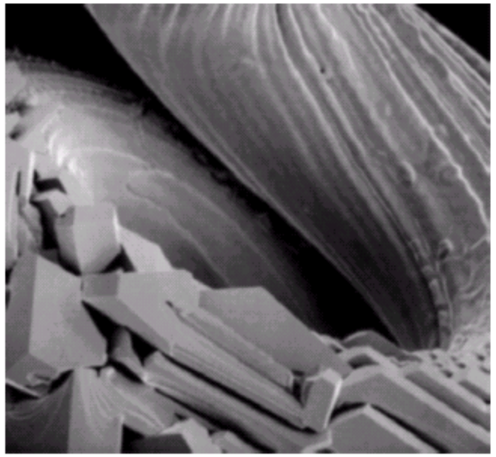
0	1	0
1	-4	1
0	1	0

Basic

Some formulations take 2nd derivatives measured across the diagonals into account

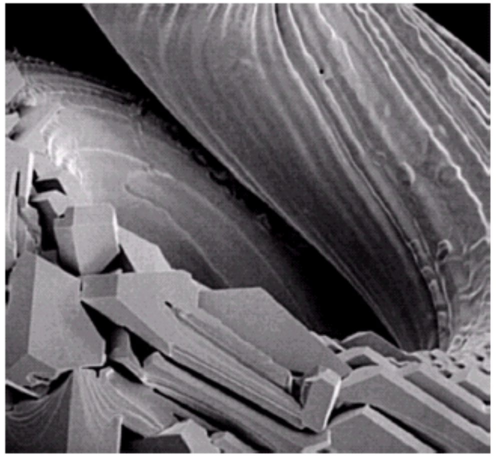
1	1	1
1	-8	1
1	1	1

Extended



Input

-1	-1	-1
-1	9	-1
-1	-1	-1



Sharpen

ACK: Prof. Tony Pridmore, UNUK

University of Nottingham
UK | CHINA | MALAYSIA

What is Edge Detection?

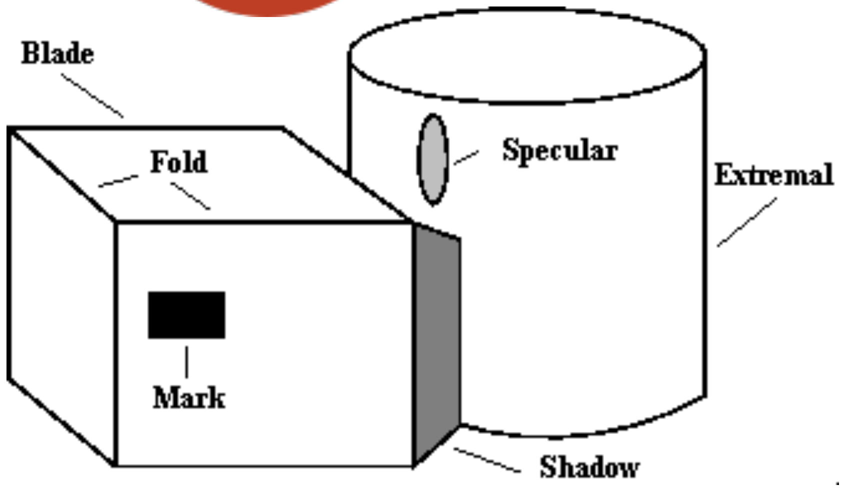
Edge Detection

First Step

In many image analysis and computer vision processes and applications

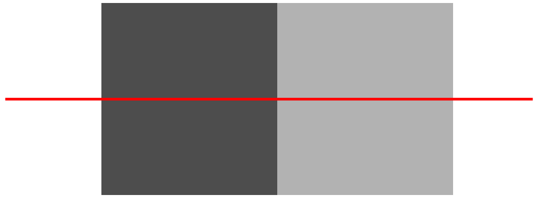
To mark points at which image intensity changes sharply - **edges**

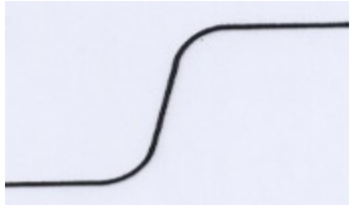
- Sharp changes in image properties reflect events/changes in the world
- This is only an assumption, but it is usually true



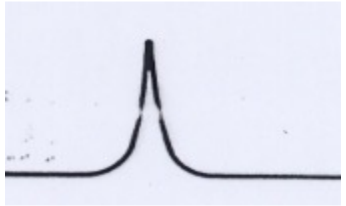
ACK: Prof. Tony Pridmore, UNUK

The Theory

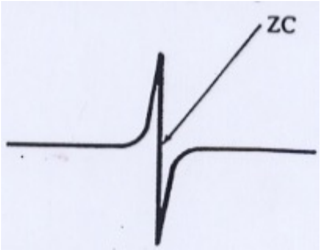




An idealised image of an edge



1st derivative: **a peak**



2nd derivative: **a zero-crossing**

To detect edges find peaks in the 1st derivative or intensity or zero-crossings in the 2nd derivative

ACK: Prof. Tony Pridmore, UNUK

The Result

```
>> im = imread('cameraman.tif');
>> edges = edge(im, 'Canny');
>> imshowpair(im, edges, 'montage');
```

ACK: Prof. Tony Pridmore, UNUK

Break

Edge Detection using 1st Derivative Filters

1st Derivative Filters

Roberts' Cross Operators

G_x:

1	0
0	-1

G_y:

0	1
-1	0

Sobel Operators

G_x:

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

G_y:

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

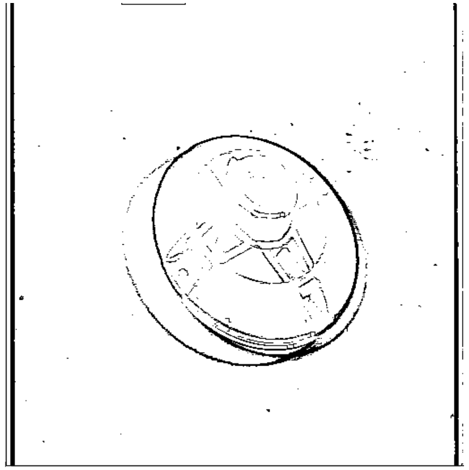
Applied separately and results combined to estimate magnitude

ACK: Prof. Tony Pridmore, UNUK

Detection & Thresholding

- Significant peaks in magnitude of 1st derivative are high
- Apply a threshold, all peaks higher than the threshold value are significant, all other are ignored

Too low



Too high

ACK: Prof. Tony Pridmore, UNUK

Edge Magnitude & Direction

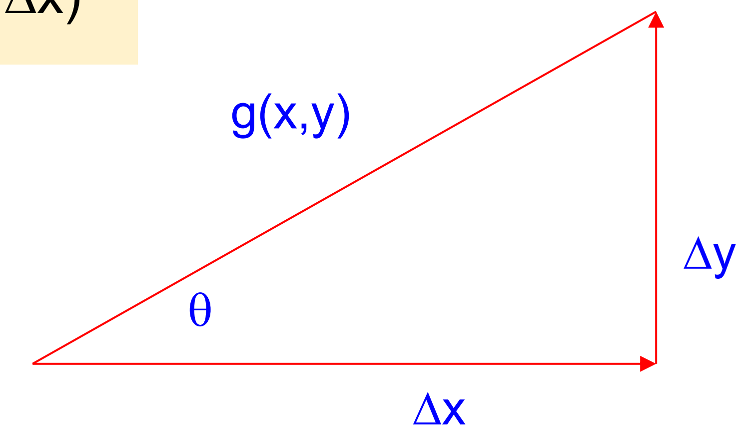
For an image function, *I(x,y)*, the gradient magnitude, *g(x,y)* is given by

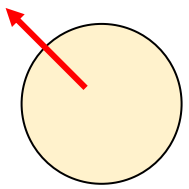
$$g(x,y) \cong (\Delta x^2 + \Delta y^2)^{1/2}$$

The gradient direction, *θ(x,y)*, gives the direction of steepest image gradient

$$\theta(x,y) \cong \mathrm{atan}(\Delta y/\Delta x)$$

This gives the direction of a line perpendicular to the edge



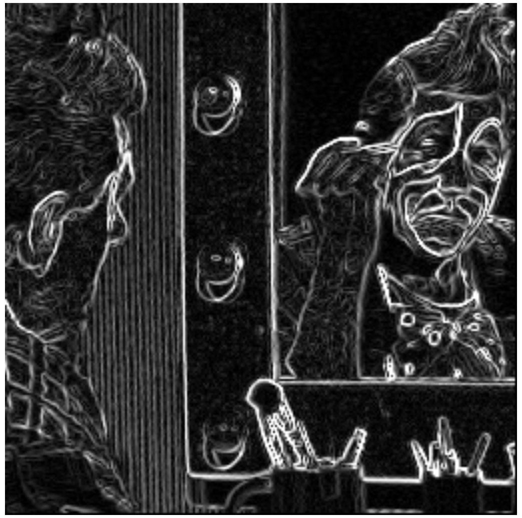


ACK: Prof. Tony Pridmore, UNUK

Roberts' Cross Operator

Very quick to compute – 4 pixels, only subtractions and additions, **but is very sensitive to noise** and only gives a strong response to very sharp edges

Original



Cross Operator

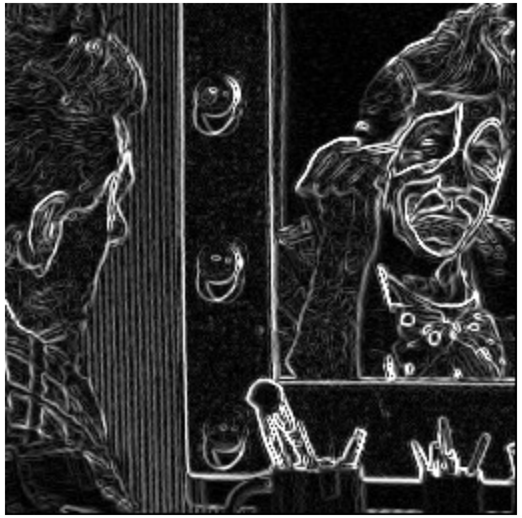
Thresholded

ACK: Prof. Tony Pridmore, UNUK

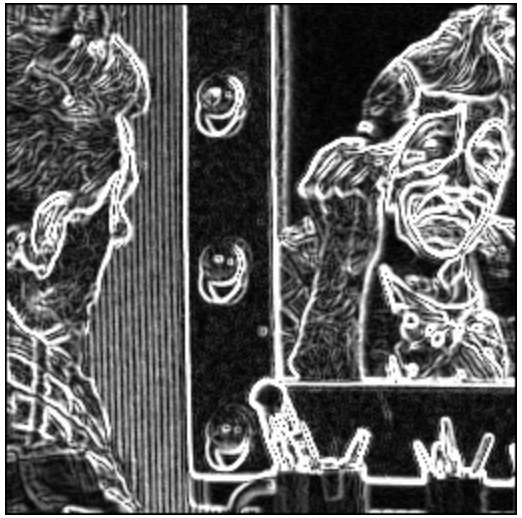
Sobel vs Roberts

- Both use a super-supplied threshold. Sobel is still in use. Roberts is less common, nowadays.
- Larger Sobel operators are more stable in noise

Original



Roberts



Sobel

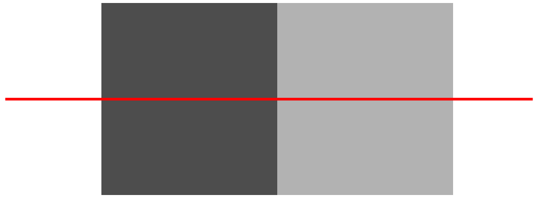
ACK: Prof. Tony Pridmore, UNUK

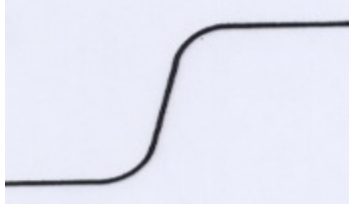
University of Nottingham
UK | CHINA | MALAYSIA

Edge Detection using 2nd Derivative Filters

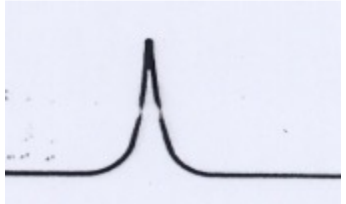
The Theory

QUICK RECAP!

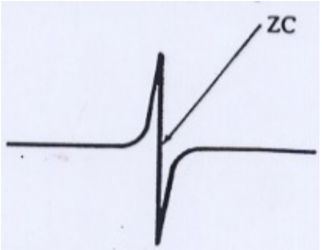




An idealised image of an edge



1st derivative: **a peak**



2nd derivative: **a zero-crossing**

To detect edges find peaks in the 1st derivative or intensity or zero-crossings in the 2nd derivative

ACK: Prof. Tony Pridmore, UNUK

2nd Derivatives: Marr-Hildreth

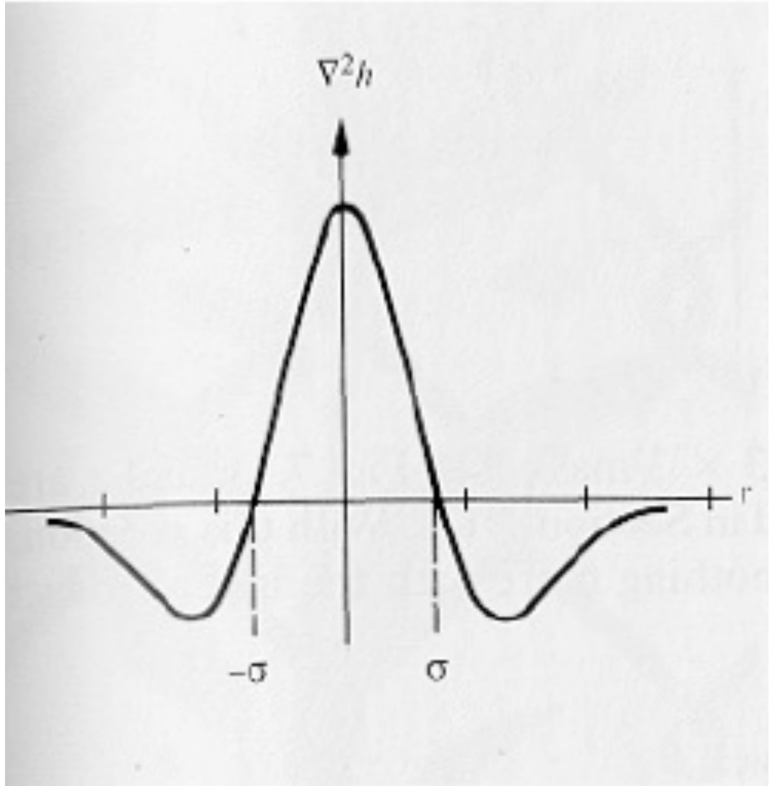
Biologically inspired

Gaussian smooth, compute Laplacian

OR

Convolve with the **Laplacian of Gaussian**

$$\nabla^2[f(x,y) * G(x,y)] = \nabla^2 G(x,y) * f(x,y)$$



ACK: Prof. Tony Pridmore, UNUK

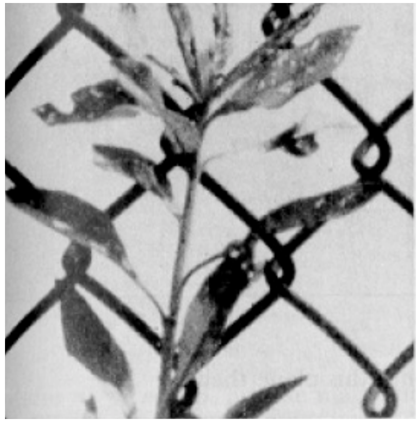
Laplacian of Gaussian (LoG)

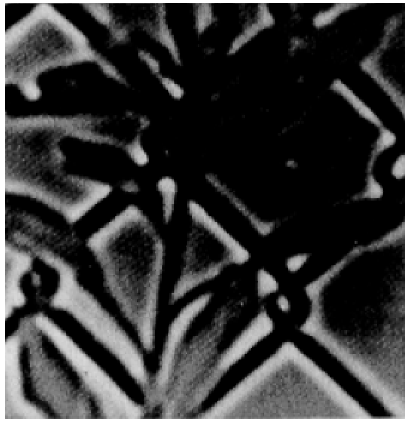
5 × 5 Laplacian of Gaussian mask

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

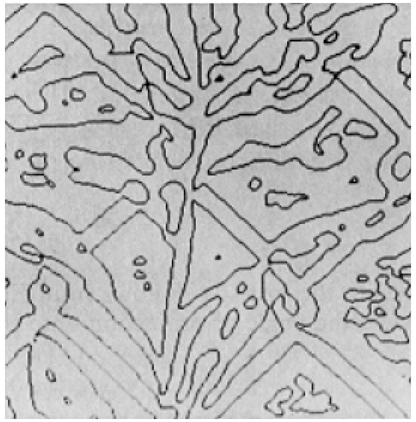
17 × 17 Laplacian of Gaussian mask

0	0	0	0	0	0	−1	−1	−1	−1	−1	0	0	0	0	0	0
0	0	0	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	0	0	0
0	0	−1	−1	−1	−2	−3	−3	−3	−3	−3	−2	−1	−1	−1	0	0
0	0	−1	−1	−2	−3	−3	−3	−3	−3	−3	−3	−2	−1	−1	0	0
0	−1	−1	−2	−3	−3	−3	−2	−3	−2	−3	−3	−3	−2	−1	−1	0
0	−1	−2	−3	−3	−3	0	2	4	2	0	−3	−3	−3	−2	−1	0
−1	−1	−3	−3	−3	0	4	10	12	10	4	0	−3	−3	−3	−1	−1
−1	−1	−3	−3	−2	2	10	18	21	18	10	2	−2	−3	−3	−1	−1
−1	−1	−3	−3	−3	4	12	21	24	21	12	4	−3	−3	−3	−1	−1
−1	−1	−3	−3	−2	2	10	18	21	18	10	2	−2	−3	−3	−1	−1
−1	−1	−3	−3	−3	0	4	10	12	10	4	0	−3	−3	−3	−1	−1
0	−1	−2	−3	−3	−3	0	2	4	2	0	−3	−3	−3	−2	−1	0
0	−1	−1	−2	−3	−3	−3	−2	−3	−2	−3	−3	−3	−2	−1	−1	0
0	0	−1	−1	−2	−3	−3	−3	−3	−3	−3	−3	−2	−1	−1	0	0
0	0	−1	−1	−1	−2	−3	−3	−3	−3	−3	−2	−1	−1	−1	0	0
0	0	0	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	0	0	0
0	0	0	0	0	0	−1	−1	−1	−1	−1	0	0	0	0	0	0





Filtering



Zero-Crossings

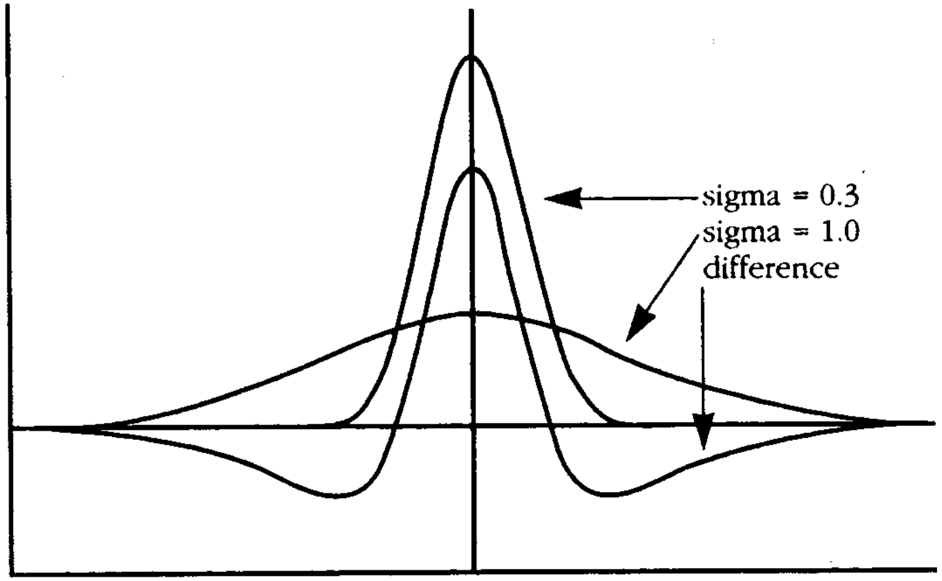
ACK: Prof. Tony Pridmore, UNUK

Difference of Gaussians

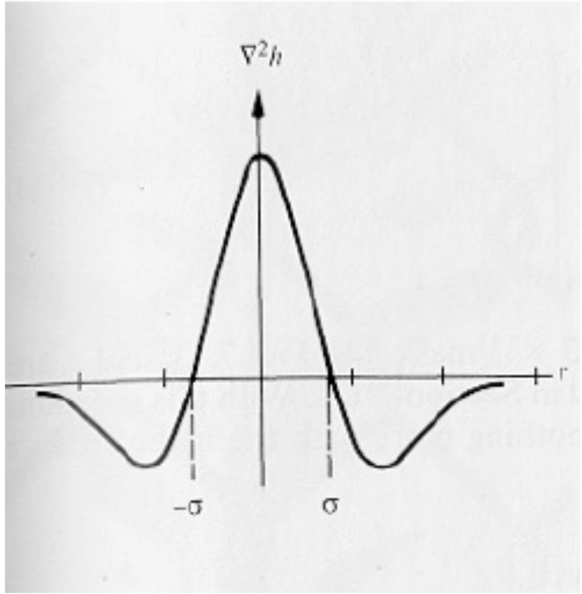
The **Laplacian of Gaussian** can be approximated by the difference between two Gaussian functions:

$$\nabla^2 G \approx G(x, y; \sigma_1) - G(x, y; \sigma_2)$$

approximation



Actual LoG



ACK: Prof. Tony Pridmore, UNUK

DoG Filtering

$$\nabla^2 G \approx G(x, y; \sigma_1) - G(x, y; \sigma_2)$$

(a) σ = 1

(b) σ = 2

(b) – (a)

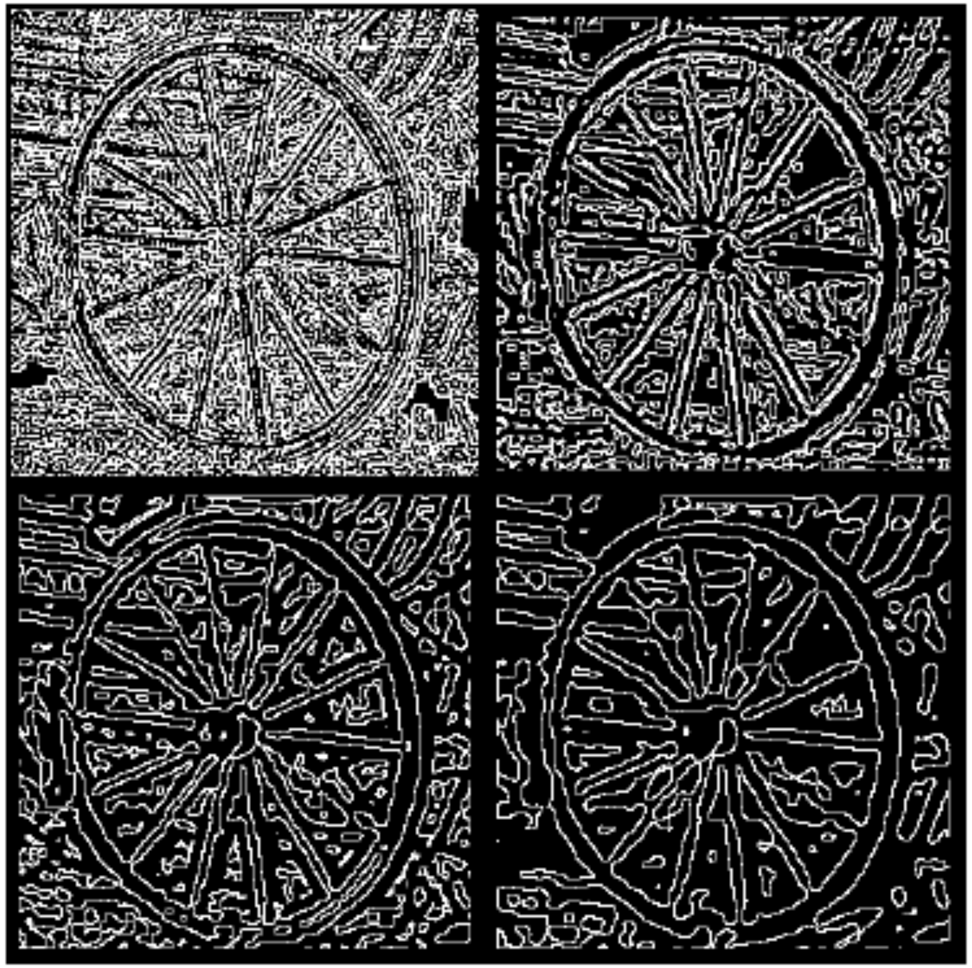
Ratio (σ_1/σ_2) for best approximation is about 1.6.
(Some people like $\sqrt{2}$.)

ACK: Prof. Tony Pridmore, UNUK

Marr-Hildreth

Choice of σ gives flexibility

Input Image



$\sigma = 1$	$\sigma = 2$
$\sigma = 3$	$\sigma = 4$

ACK: Prof. Tony Pridmore, UNUK

1st vs 2nd Derivative Methods

Peaks in 1st Derivative

- Strong response at edges, but also respond to noise
- Peak detection and threshold selection need care

VS

Zero crossings in 2nd derivative

- Well-defined, easy to detect
- Must form smooth, connected contours
- Tend to round off corners

1st derivative methods are much more common in practical applications,

In part because of John Canny

ACK: Prof. Tony Pridmore, UNUK

The Canny Operator

What Canny Did

John Canny tried to find the optimal edge detector, assuming a perfect step edge in Gaussian noise

Optimal

- **Good Detection** → It should mark all the edges and only all the edges
- **Good Localisation** → The points marked should be as close to the real edge as possible
- **Minimal Response** → Each edge should be reported only once

Canny used the *Calculus of Variations:* finds the function which best satisfies some functional

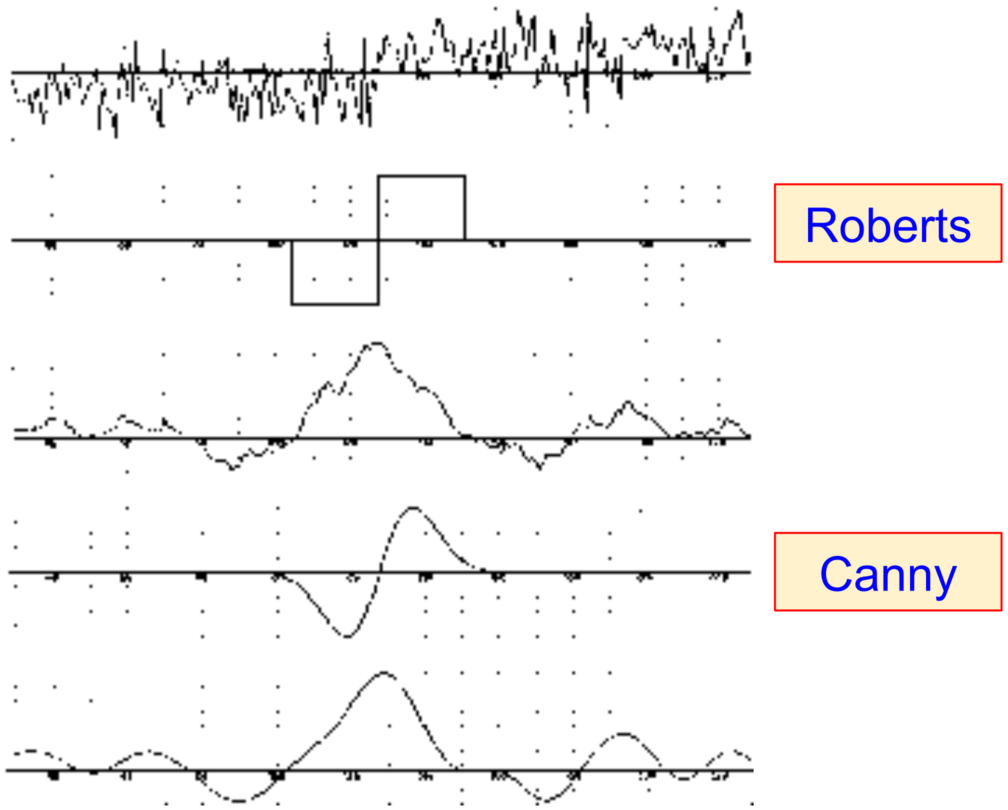
ACK: Prof. Tony Pridmore, UNUK

The Canny Operator

- The optimal detector was a sum of 4 exponential terms, but is very closely approximated by the 1^{st} derivative of a Gaussian

i.e., 1^{st} derivative of a Gaussian smoothed image

- Gives a cleaner response to a noisy edge than square operators
- Most implementations are 2D Gaussian smoothing + Roberts style derivative



ACK: Prof. Tony Pridmore, UNUK

Non-Maximal Suppression

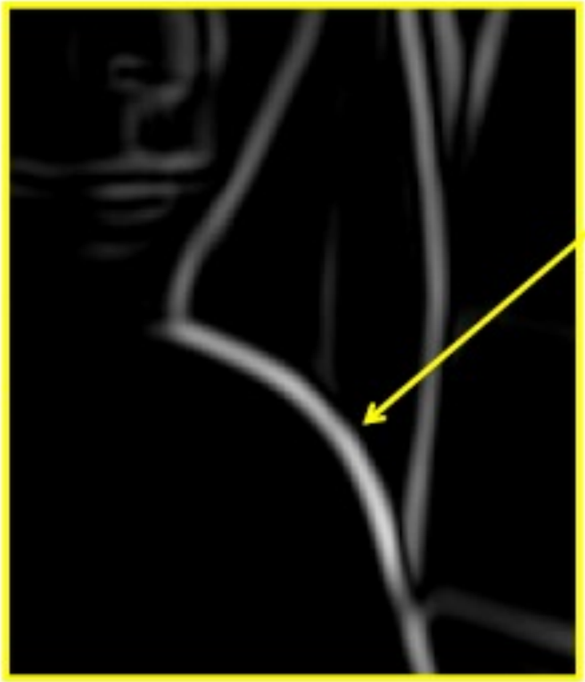
The Canny operator’s response is cleaner than Sobel or Roberts, but it needs an explicit step to enforce Minimal Response

Canny operator

ACK: Prof. Tony Pridmore, UNUK

Non-Maximal Suppression

Thresholding raw operator response would leave thick lines

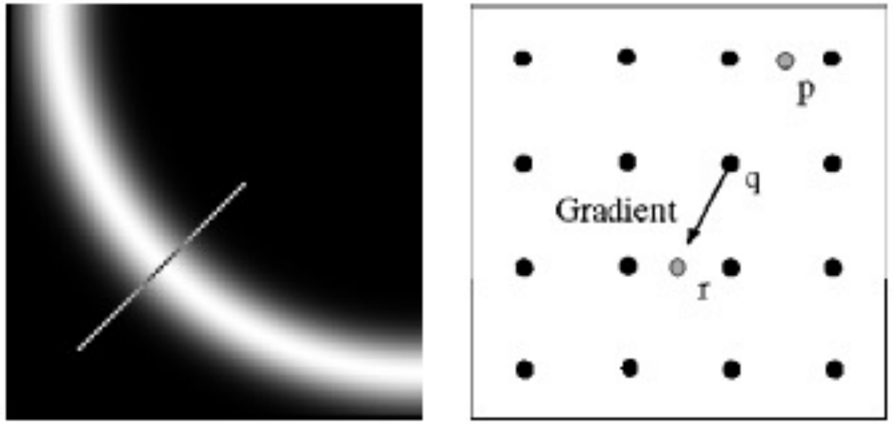


How to turn
these thick
regions of the
gradient into
curves?

ACK: Prof. Tony Pridmore, UNUK

Non-Maximal Suppression

1. Check if pixel is a local maximum along the gradient direction
2. Select a single maximum across the width of the edge

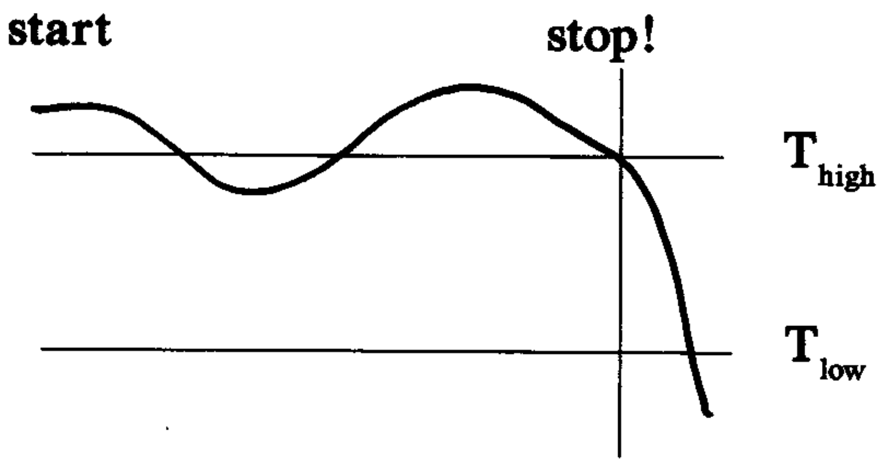


ACK: Prof. Tony Pridmore, UNUK

Thresholding with Hysteresis

- Simple thresholding tests each pixel independently: *edges aren't really independent, they make up lines*
- The industry standard edge thresholding method

- Allows a band of variation, but assumes continuous edges
- User still selects parameters, but its easier, less precise

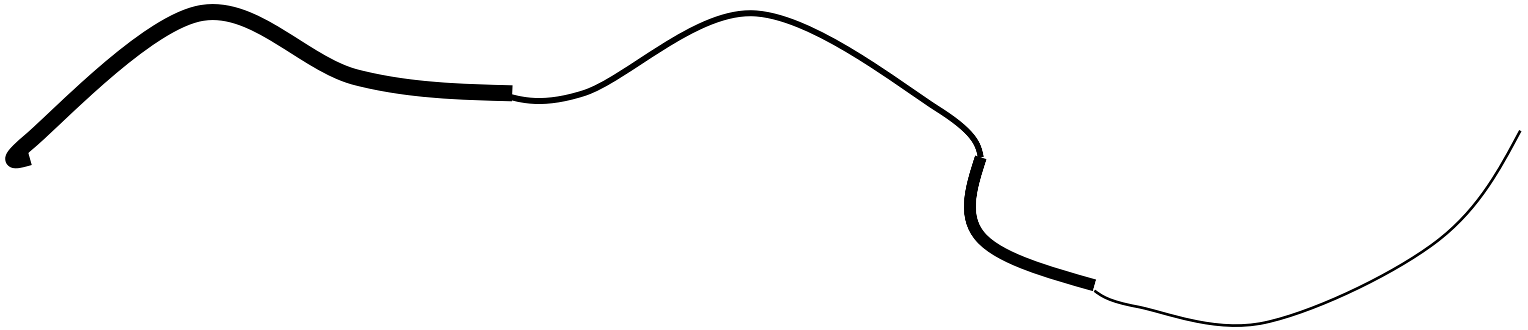


ACK: Prof. Tony Pridmore, UNUK

Thresholding with Hysteresis

The effect is to keep weak edges if they connect strong edges, as long as

- The strong edges are really strong
- The weak edges aren't really weak



ACK: Prof. Tony Pridmore, UNUK

Thresholding with Hysteresis

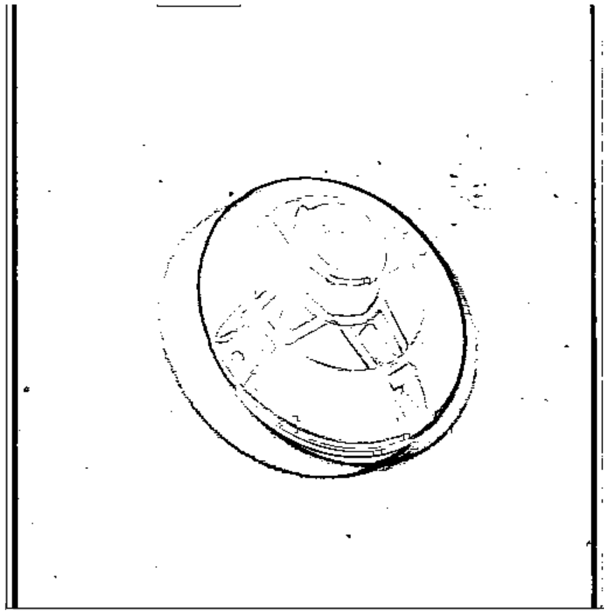
Hysteresis fills in most of the gaps

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

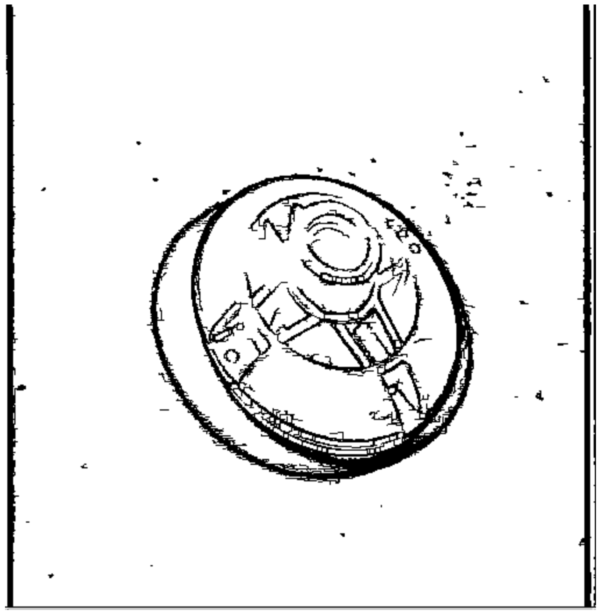
ACK: Prof. Tony Pridmore, UNUK

Thresholding with Hysteresis

Low threshold



High threshold



Hysteresis

ACK: Prof. Tony Pridmore, UNUK

What Canny Did

Showed that 1st derivative of a Gaussian smoothed image is the optimal way to detect step edges in noise

Explained why 1st derivatives are a good idea

Designed the industry standard thresholding method

- Non-maximal suppression
- Thresholding with hysteresis

Effectively **solved** the edge detection problem

ACK: Prof. Tony Pridmore, UNUK

Summary

1. Derivative Filters
2. Sharpening
3. What is Edge Detection?
4. Edge Detection using 1st Derivatives
5. Edge Detection using 2nd Derivatives
6. The Canny Operator

University of Nottingham
UK | CHINA | MALAYSIA

Questions

University of Nottingham
UK | CHINA | MALAYSIA

NEXT:

Whole Image Methods