CSE4227 Digital Image Processing

Chapter 03 – **Sharpening Spatial Filter**

Dr. Kazi A Kalpoma
Professor, Department of CSE

Ahsanullah University of Science & Technology (AUST)

Today's Contents

- **Sharpening spatial filter**
- **Derivatives of Image**
  - 1st derivative
  - 2nd derivative
- **Laplacian Filter**
- **Laplacian Image Enhancement**
- **Gradient Operators**
- **Difference filters**
- **Combining filtering techniques**

Chapter 3 from R.C. Gonzalez and R.E. Woods, Digital Image Processing (3rd Edition), Prentice Hall, 2008 [**Section 3.6, 3.7**]

Sharpening Spatial Filters

Previously we have looked at Smoothing filters which **remove fine details.**

Sharpening spatial filters seek to **highlight fine details**.

- Remove **blurring** from images
- Highlight edges
- Useful for emphasizing **transitions** in image intensity

Sharpening Spatial Filters

Some Applications

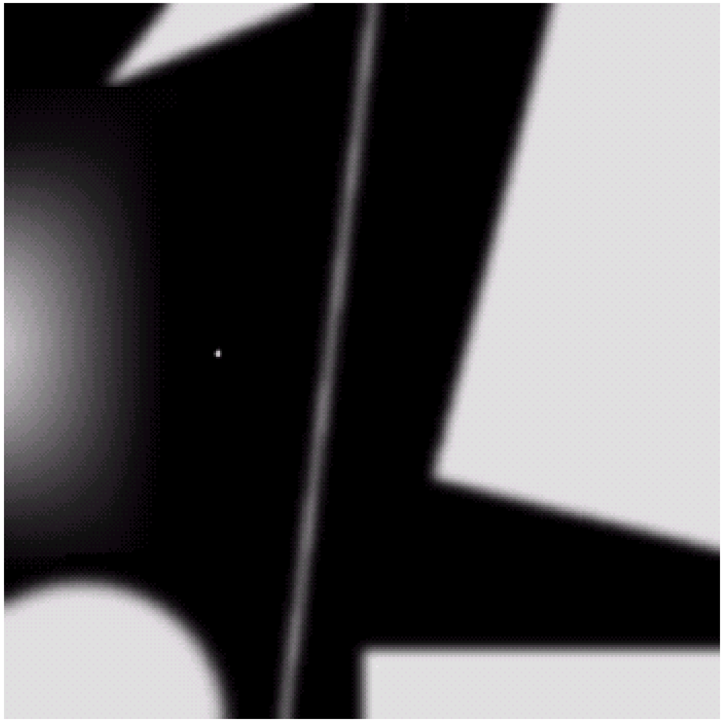
- Photo Enhancement
- Medical image visualization
- Industrial defect detection
- Electronic printing
- Autonomous guidance in military systems

Spatial Differentiation

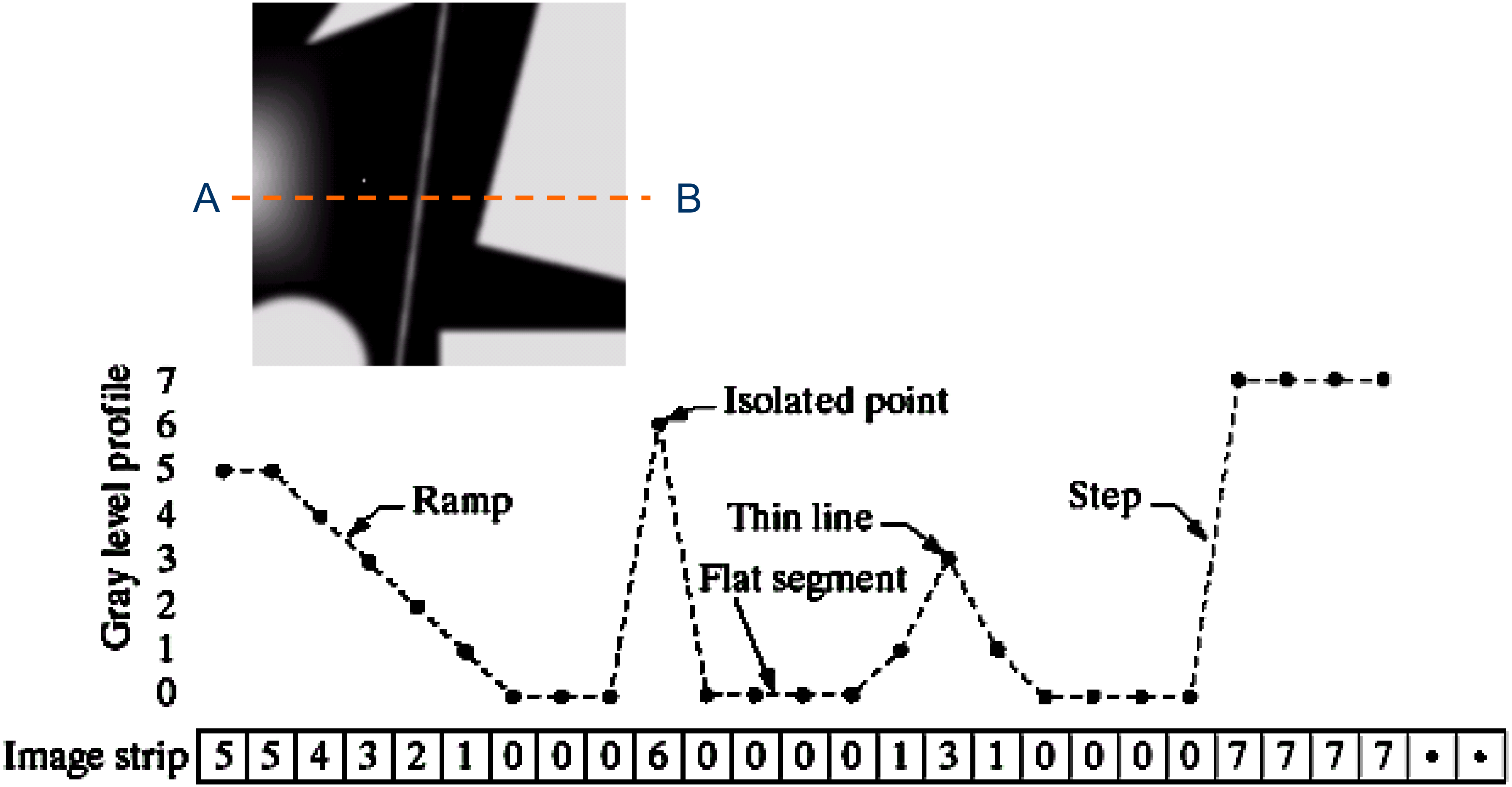
- Sharpening filters are based on first- and second-order derivatives of image.
- The derivatives of a digital function are defined in terms of differences.
- Differentiation measures the rate of change of a function.
- *i.e.* Sharpening filters are based on ***spatial differentiation***

Spatial Differentiation

- Let's consider a simple 1 dimensional example



Spatial Differentiation



1^{st} Derivative in Digital Form

- The 1^{st} derivative of a function is given by:

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

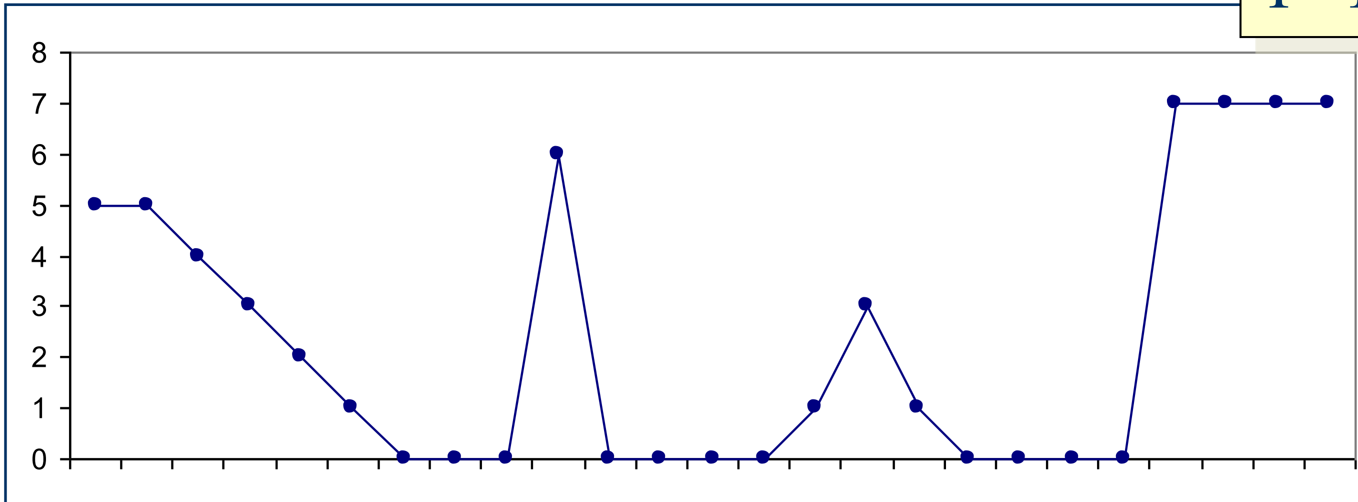
forward

or $f(x) - f(x-1)$

backward

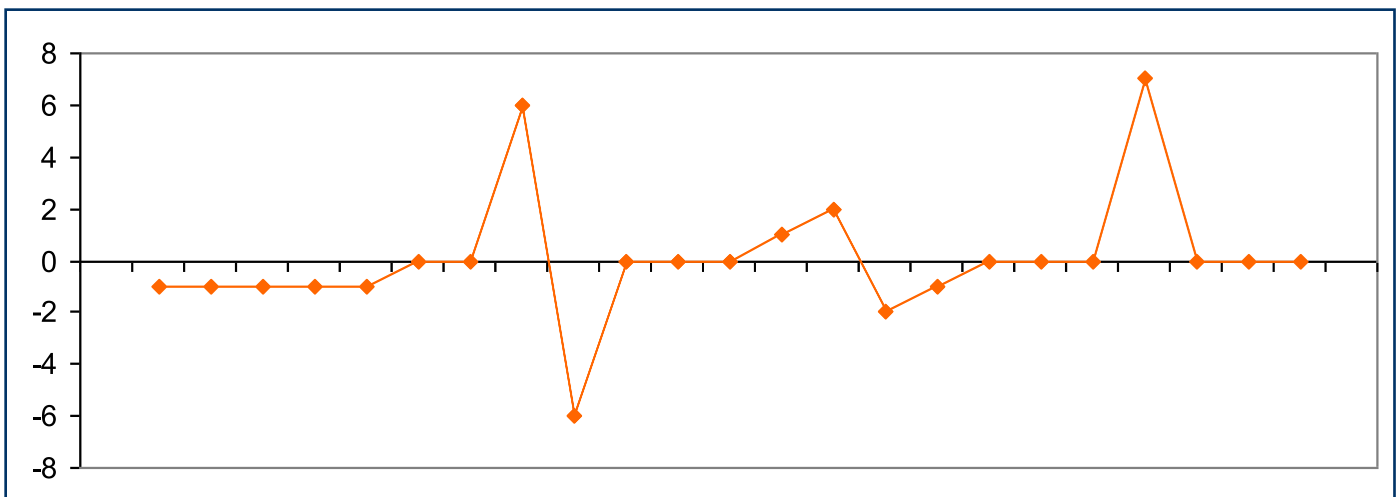
- Its just the difference between subsequent values and measures the rate of change of the function

1^{st} Derivative



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

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



Derivative is nonzero along the entire ramp, zero in flat area,.

2^{nd} Derivative in Digital Form

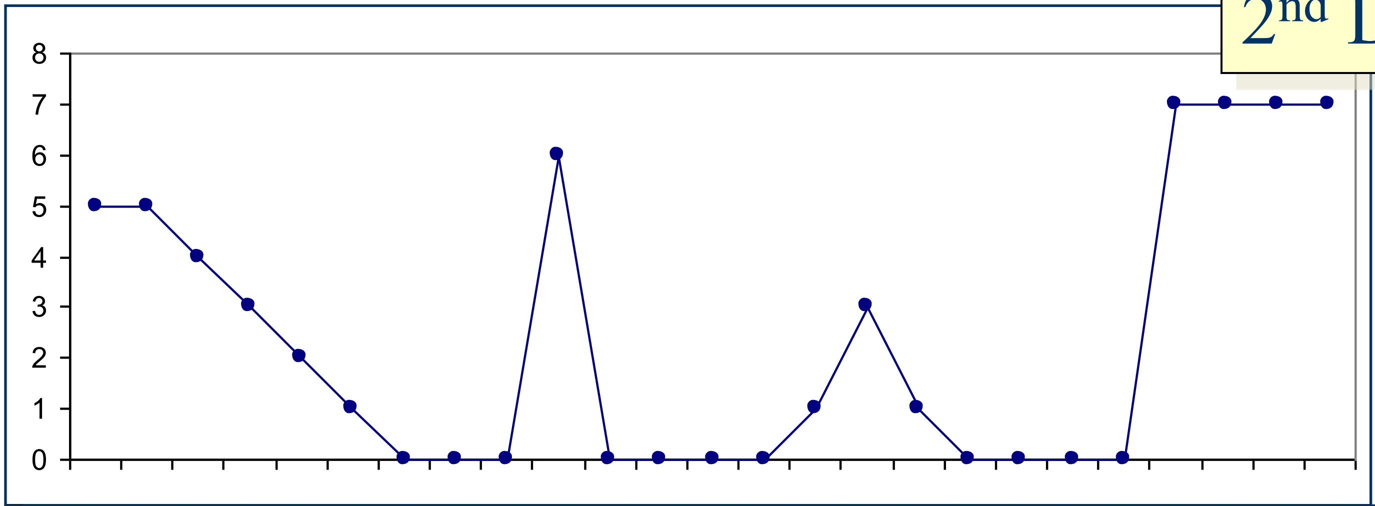
The 2nd derivative of a function is given by:

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

or $2f(x) - f(x-1) - f(x+1)$

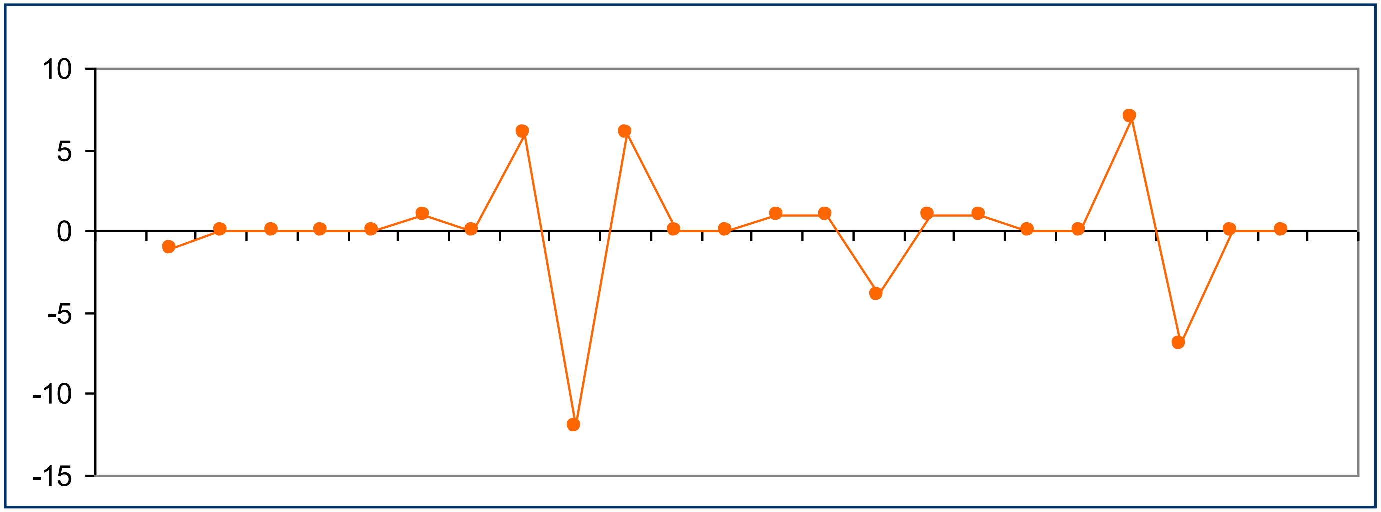
Simply takes into account the values both before (backward) and after (forward) the current value

2^{nd} Derivative



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

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



Derivative is nonzero at the onset and end of ramp, stronger response at and around the point.

1^{st} Derivative for Two Dimensional

$$f(x+1,y)-f(x,y)$$

and $f(x,y+1)-f(x,y)$

$$f(x,y)-f(x-1,y)$$

and $f(x,y)-f(x,y-1)$

2^{nd} Derivative for Two Dimensional

$$f(x+1,y)+f(x-1,y)-2f(x,y)$$

and $f(x,y+1)+f(x,y-1)-2f(x,y)$

OR

$$2f(x,y)-f(x-1,y)-f(x+1,y)$$

and $2f(x,y)-f(x,y-1)-f(x,y+1)$

Sharpening Spatial Filters

1. LAPLACIAN
   - Use of 2^{nd} Derivative for Image Enhancement
2. SOBEL (Gradient Operators)
   - Use of 1st Derivative for Image Enhancement

Use of 2nd Derivative for Image Enhancement

The 2nd derivative is more useful for image enhancement than the 1st derivative - *Stronger response to fine detail*

The first sharpening filter we will look at is the ***Laplacian***

2nd derivatives for image Sharpening - For Two Dimensional

- 2-D 2nd derivatives => Laplacian

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

=>discrete formulation

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

2nd Derivative in Two Dimension

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x) \rightarrow \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$

x kernel

$$\frac{\partial^2 f}{\partial y^2} = f(y+1) + f(y-1) - 2f(y) \rightarrow \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

y kernel

0	0	0
1	-2	1
0	0	0

+

0	1	0
0	-2	0
0	1	0

=

0	1	0
1	-4	1
0	1	0

1. Laplacian Filter

So, the Laplacian can be given as follows:

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

We can implement it using this filter.

0	1	0
1	-4	1
0	1	0

Types of Laplacian Kernels

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

a	b
c	d

FIGURE 3.39 (a) Filter mask used to implement the digital Laplacian, as defined in Eq. (3.7-4). (b) Mask used to implement an extension of this equation that includes the diagonal neighbors. (c) and (d) Two other implementations of the Laplacian.

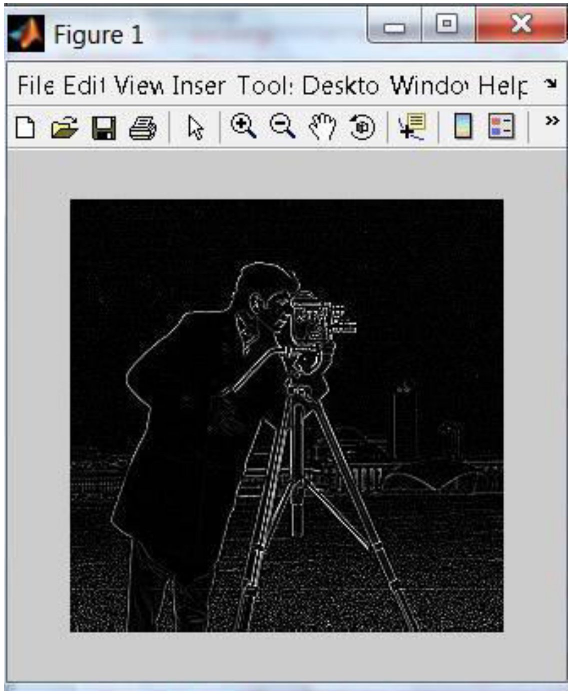
Laplacian Image Enhancement

Another Example:

0	-1	0
-1	4	-1
0	-1	0

Original Image

+



Laplace Sharpened image

=

Laplace filtered image

Laplacian Filter

Example: apply the following Laplacian filter on the highlighted and underlined pixel

0	-1	0
-1	4	-1
0	-1	0

153	157	156	153	155
159	156	158	156	159
155	158	**<u>154</u>**	156	160
154	157	158	160	160
157	157	157	156	155

Step 1:

154*4 – 158- 156-158-158 = -14

So the value after filter = **-14**

We call the resultant image: **sharpened image.**

Step 2:

Filtered image=original + sharpened image

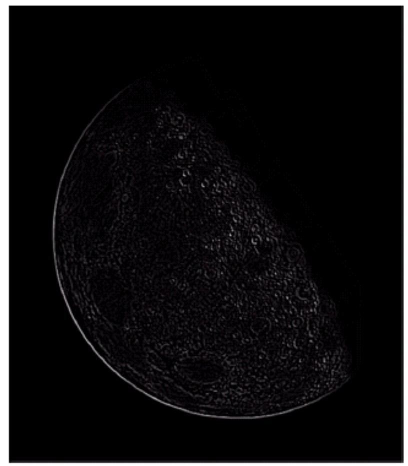
The value in the filtered image=154-14 =130

Laplacian Image Enhancement

0	1	0
1	-4	1
0	1	0

Original Image

-



Laplacian Filtered Image

=

Sharpened Image

In the final sharpened image edges and fine detail are much more obvious

Simplified Image Enhancement

- The result of a Laplacian filtering is not an enhanced image.
- The entire enhancement can be combined into a **single filtering operation**

$$g(x,y) = f(x,y) - \nabla^2 f$$

$$= f(x,y) - [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)]$$

0	1	0
1	-4	1
0	1	0

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f, & w_5 < 0 \\ f(x,y) + \nabla^2 f, & w_5 > 0 \end{cases}$$

Simplified Image Enhancement

- The entire enhancement or sharpening can be done in one PASS.

$$g(x,y) = f(x,y) - \nabla^2 f$$

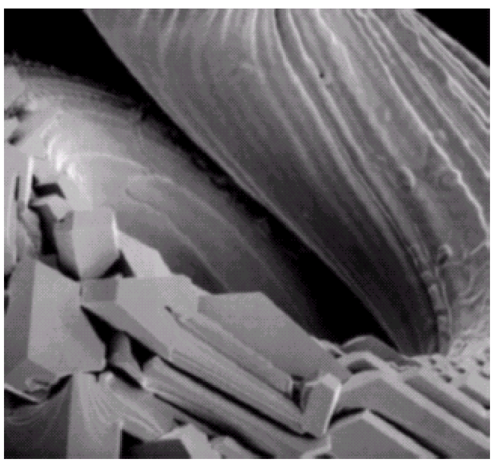
$$= 5f(x,y) - f(x+1,y) - f(x-1,y) - f(x,y+1) - f(x,y-1)$$

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

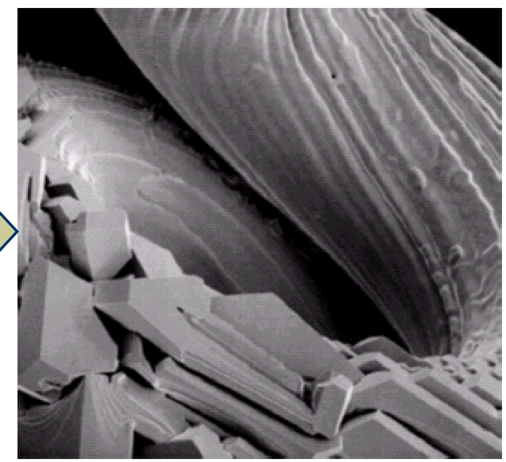
We can implement it using this filter.

Simplified Image Enhancement

- This gives us a new filter which does the whole job for us in one step



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



Variants On The Simple Laplacian

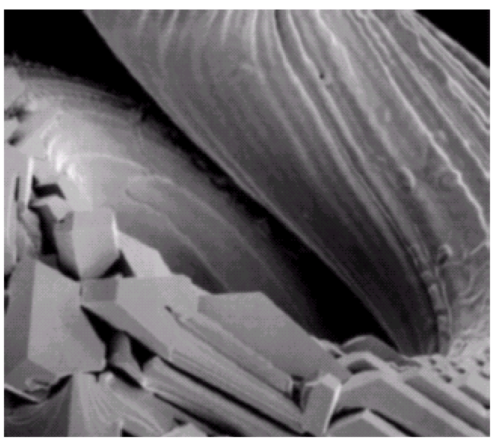
There are lots of slightly different versions of the Laplacian that can be used:

0	1	0
1	-4	1
0	1	0

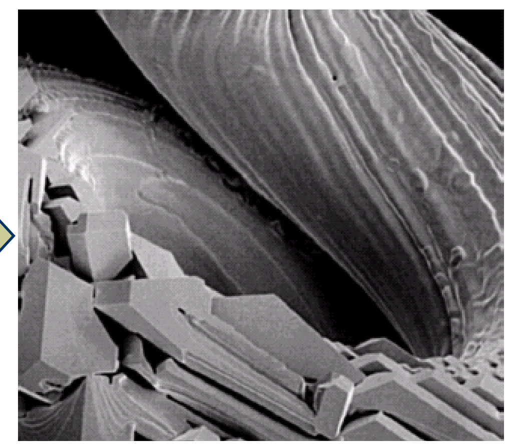
Simple Laplacian

1	1	1
1	-8	1
1	1	1

Variant of Laplacian



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



Use of 1st Derivatives for Image Enhancement

The another Sharpening Spatial filters is SOBEL (Gradient Operators).

What is Gradient of a Digital Image?

The Gradient of a Digital Image

The Gradient (1st order derivative)

- First Derivatives in image processing are implemented using the magnitude of the gradient.
- The gradient of function f(x,y) is

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

The Gradient of a Digital Image

- The magnitude of this vector is given by

$$mag(\nabla f) = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y|$$

G_x

-1	1

This mask is simple, and no isotropic. Its result only horizontal and vertical.

G_y

1
-1

The Gradient – First-order Derivative

How can we compute first-order discrete image derivatives?

- There are various ways…
  - One dimensional forward differences
  - Roberts cross gradient operators
  - One dimensional central differences
  - Prewitt operators
  - Sobel operators

There is some debate as to how best to calculate these gradients.

Gradient Operators

Robert's Method

- The simplest approximations to a first-order derivative that satisfy the conditions stated in that section are

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$G_x = (z_9 - z_5)$ and $G_y = (z_8 - z_6)$

$$\nabla f = \sqrt{(z_9 - z_5)^2 + (z_8 - z_6)^2}$$

$$\nabla f \approx |z_9 - z_5| + |z_8 - z_6|$$

Gradient Operators

- These mask are referred to as the Roberts cross-gradient operators.

-1	0
0	1

0	-1
1	0

Sharpening Spatial filters :
2) SOBEL (Gradient Operator)

- The Sobel operator provides differencing and smoothing effect of an image.
- Sobel operator consists of 3x3 convolution kernels. Gx is a simple kernel and Gy is rotated by 90˚

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

Gy, Extract horizontal edges

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

Gx, Extract vertical edges

SOBEL Operator on an Image

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

Gx

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

Gy

The Sobel Operator involves estimating the first derivative of an image **by doing a convolution between an image and two special kernels,** one to detect vertical edges and one to detect horizontal edges.

Gradient Operators

Sobel Operator

$$\frac{\partial f}{\partial y} = \begin{array}{|c|c|c|}\hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Gy, Extract horizontal edges

$$\frac{\partial f}{\partial x} = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$$

Gx, Extract vertical edges

$$|G| = |Gx| + |Gy|$$

$$\nabla f \approx \left|(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)\right| + \left|(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)\right|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Pixel Arrangement

Gradient Operators

Prewitt Operator

- is used for detecting edges horizontally and vertically.

$$|G| = |Gx| + |Gy|$$

$$\nabla f \approx \left|(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)\right| + \left|(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)\right|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Pixel Arrangement

$$\frac{\partial f}{\partial y} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Extract horizontal edges

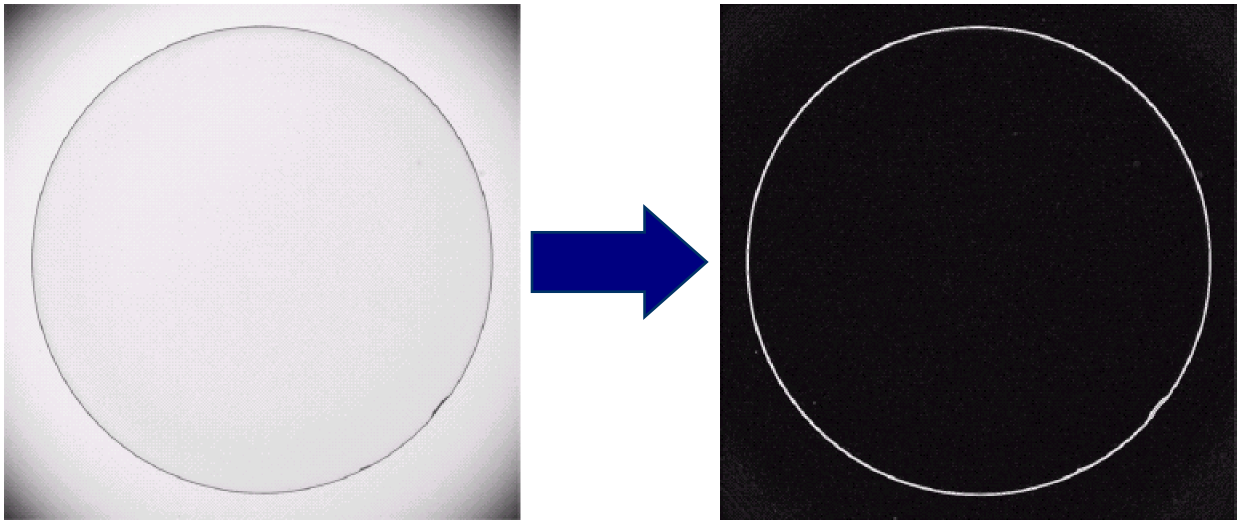
Gy

$$\frac{\partial f}{\partial x} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Extract vertical edges

Gx

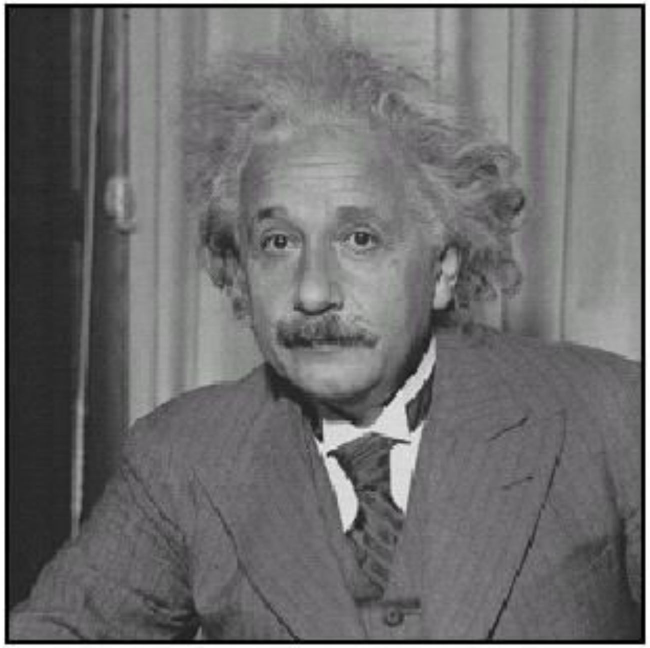
Sobel Operator: Example



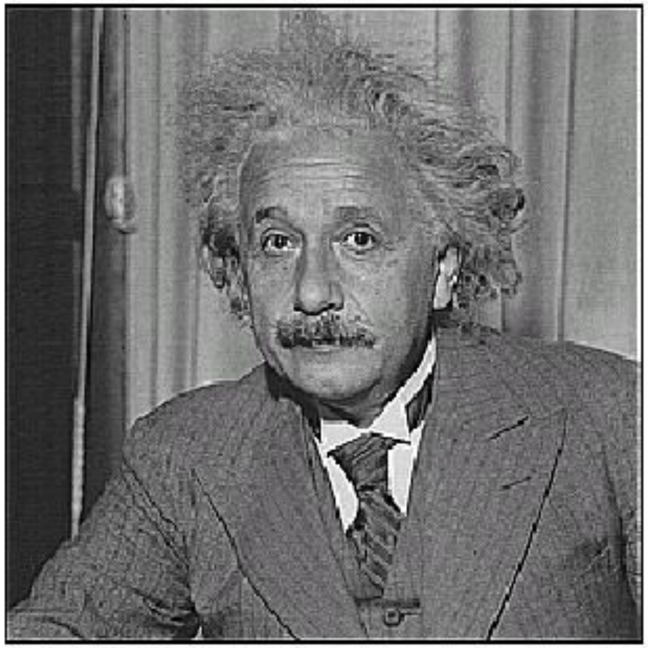
An image of a contact lens which is enhanced in order to make defects more obvious

Sobel filters are typically used for edge detection

Sharpening with Sobel Operator

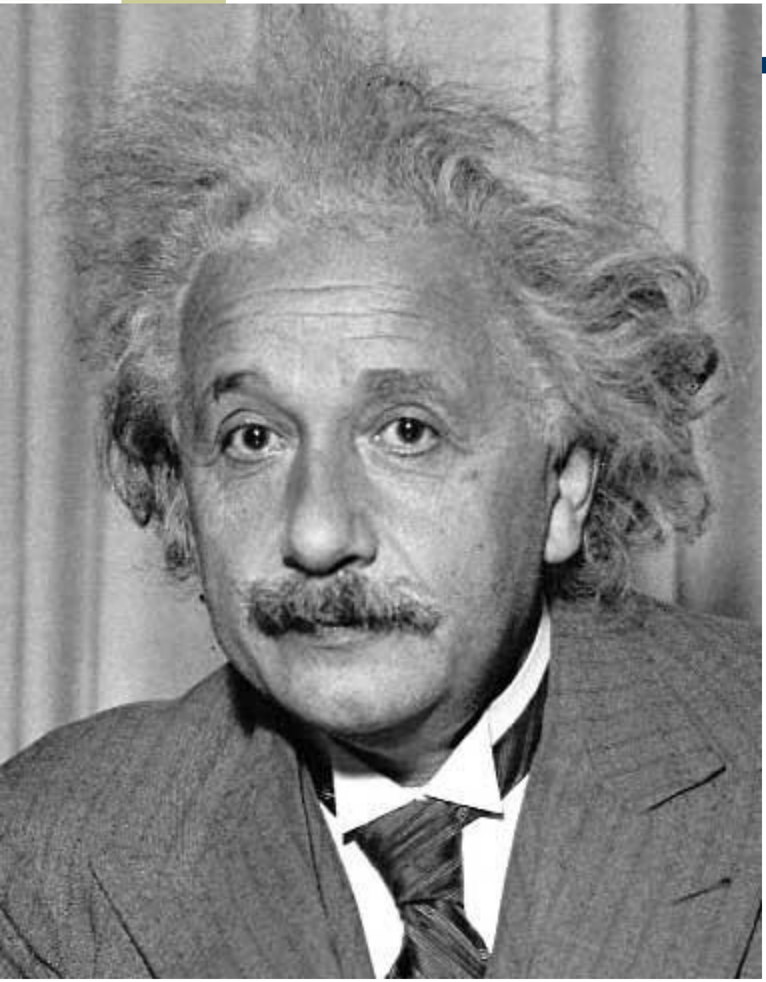


before



after

Sharpening with Sobel Operator

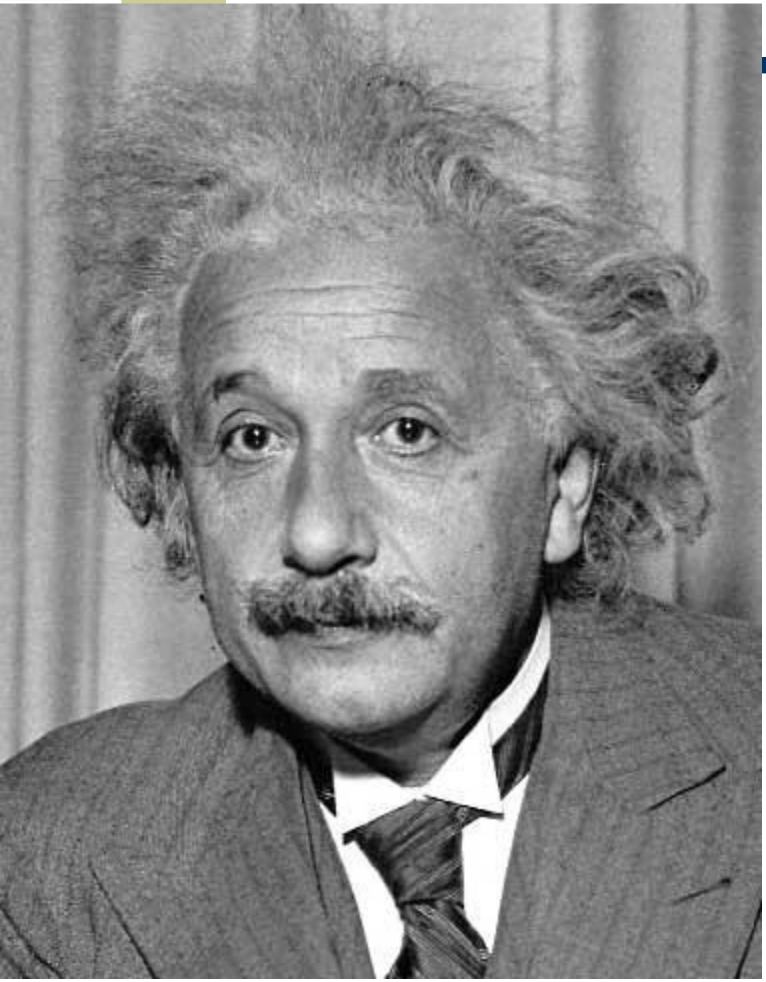


-1	0	1
-2	0	2
-1	0	1

Sobel

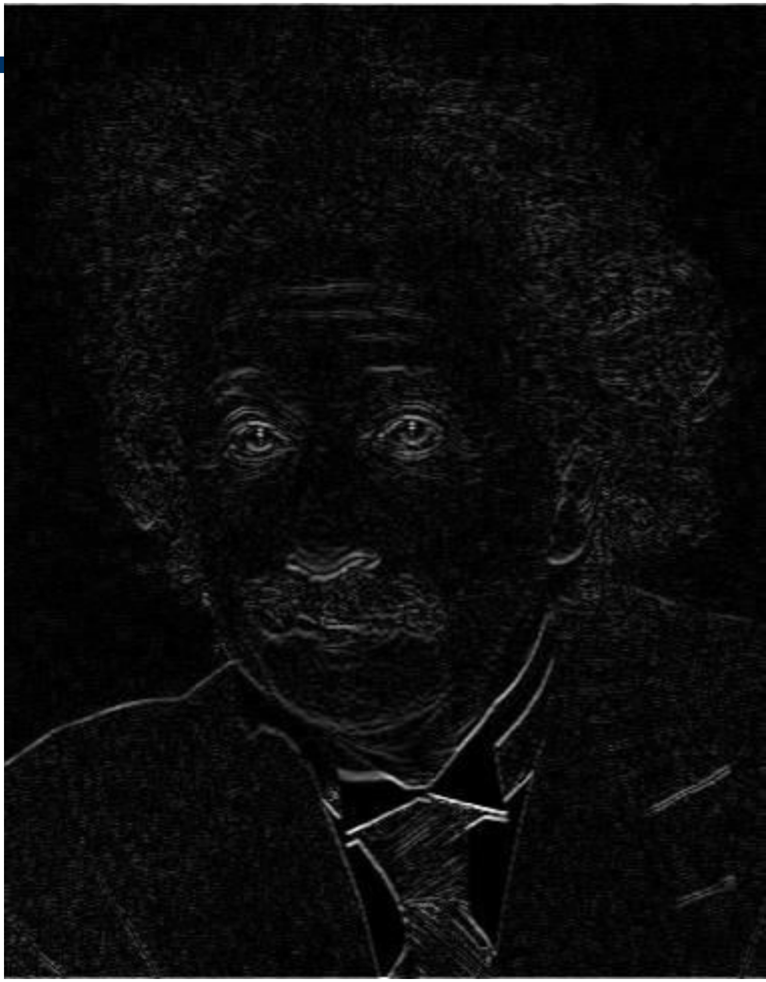
Vertical Edge
(absolute value)

Sharpening with Sobel Operator



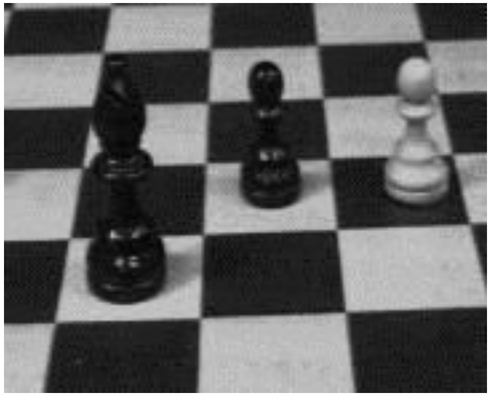
1	2	1
0	0	0
-1	-2	-1

Sobel

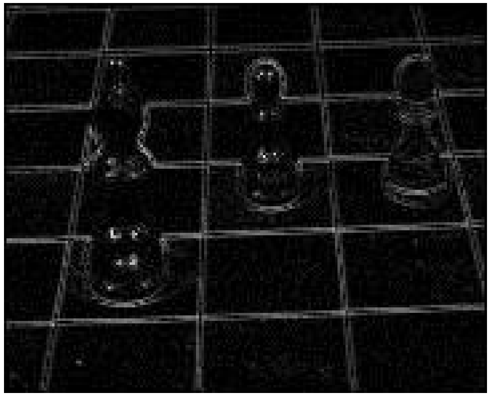


Horizontal Edge
(absolute value)

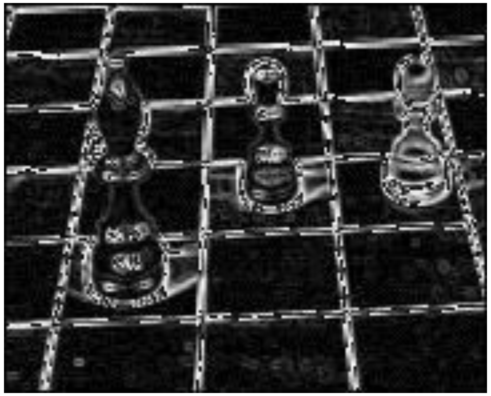
Sharpening Spatial Filters



Laplacian



Sobel



Difference Filter

- Also called as Emboss filters
- Enhances the details in the direction specific to the mask selected
- Four primary difference filter convolution masks, corresponding to the edges in the vertical, horizontal, and two diagonal directions are:

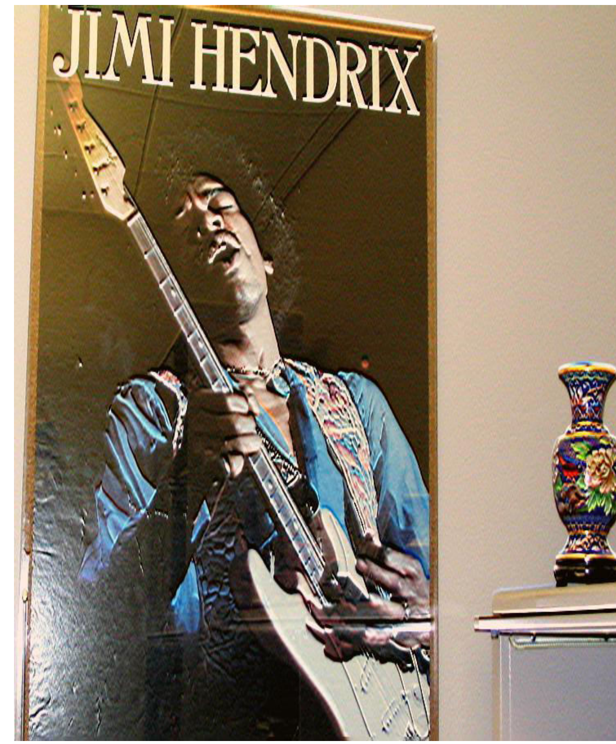
Vertical **Horizontal** **Diagonal 1** **Diagonal 2**

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

Difference Filter

Original image

Difference filtered image



Difference filtered image added to the original image, with contrast enhanced

Combining Spatial Enhancement Methods

- Successful image enhancement is typically not achieved using a single operation
- Rather we combine a range of techniques in order to achieve a final result
- This example will focus on enhancing the bone scan

Combining Spatial Enhancement Methods

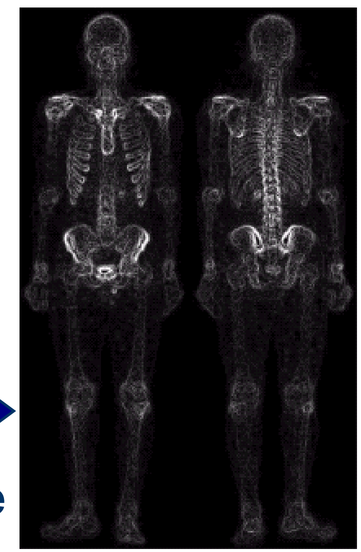
(a)

(b)

Laplacian filter of bone scan (a)

(c)

Sharpened version of bone scan achieved by subtracting (a) and (b)



(d)

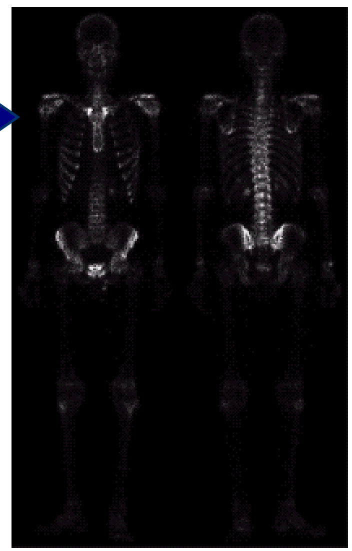
Sobel filter of bone scan (a)

Combining Spatial Enhancement Methods

Image (d) smoothed with a 5*5 averaging filter

(e)

The product of (c) and (e) which will be used as a mask



(f)

Sharpened image which is sum of (a) and (f)

(g)

Result of applying a power-law trans. to (g)

(h)

Combining Spatial Enhancement Methods

Compare the original and final images

Class Work

Consider a 3-bit 4x4 image.

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

Laplacian filter

0	-1	0
-1	+4	-1
0	-1	0

Find the filtered output image using
•this **Laplacian** filter,
•a 3 × 3 **Mean** filter
•a 3x3 **Median** filter and
•a **Sobel** operator

Ignore the border pixels in calculation and put zero in the border of the output image.