Digital Image Processing

Spatial Filtering

Christophoros Nikou
cnikou@cs.uoi.gr

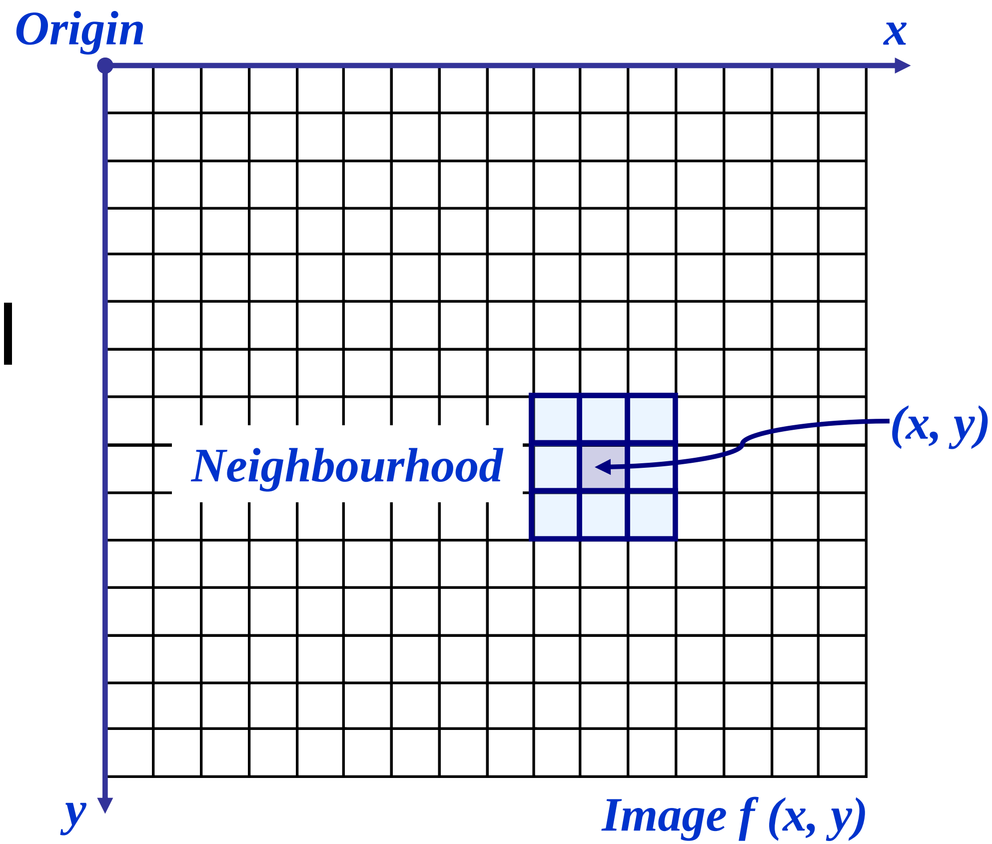
Contents

In this lecture we will look at spatial filtering techniques:

– Neighbourhood operations
– What is spatial filtering?
– Smoothing operations
– What happens at the edges?
– Correlation and convolution
– Sharpening filters
– Combining filtering techniques

Neighbourhood Operations

- Neighbourhood operations simply operate on a larger neighbourhood of pixels than point operations
- Neighbourhoods are mostly a rectangle around a central pixel
- Any size rectangle and any shape filter are possible

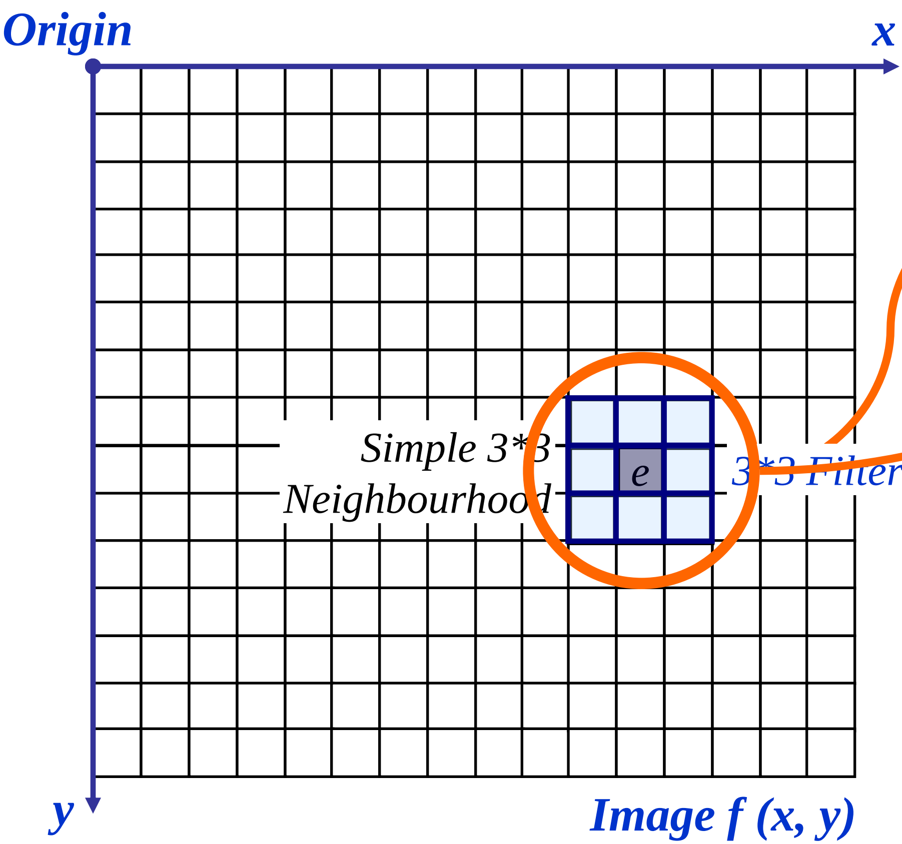


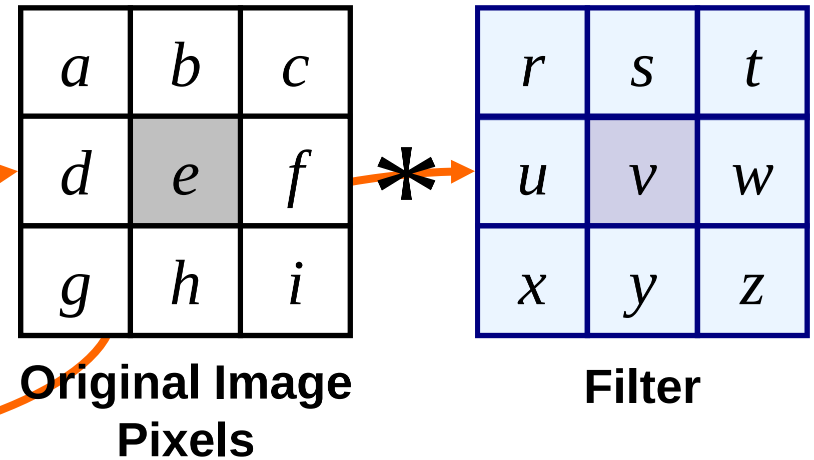
Simple Neighbourhood Operations

Some simple neighbourhood operations include:

- **Min:** Set the pixel value to the minimum in the neighbourhood
- **Max:** Set the pixel value to the maximum in the neighbourhood
- **Median:** The median value of a set of numbers is the midpoint value in that set (e.g. from the set [1, 7, 15, 18, 24] the median is 15). Sometimes the median works better than the average

The Spatial Filtering Process

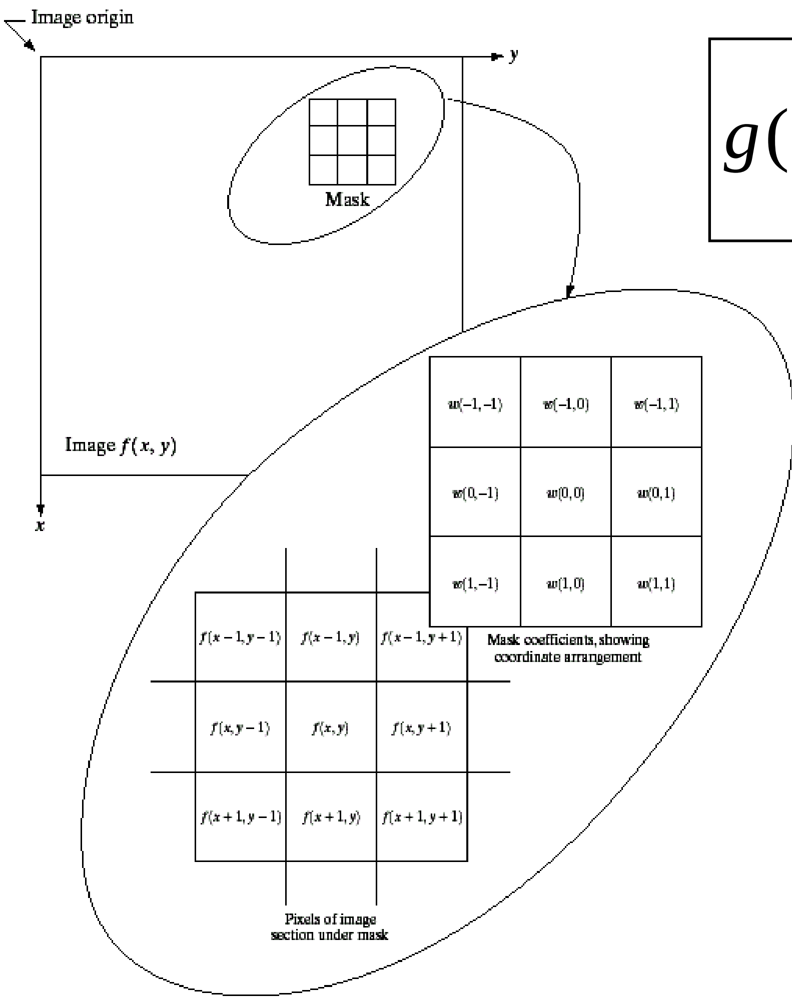




$$e_{processed} = v*e + r*a + s*b + t*c + u*d + w*f + x*g + y*h + z*i$$

The above is repeated for every pixel in the original image to generate the filtered image

Spatial Filtering: Equation Form



$$\boxed{g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)}$$

Filtering can be given in equation form as shown above

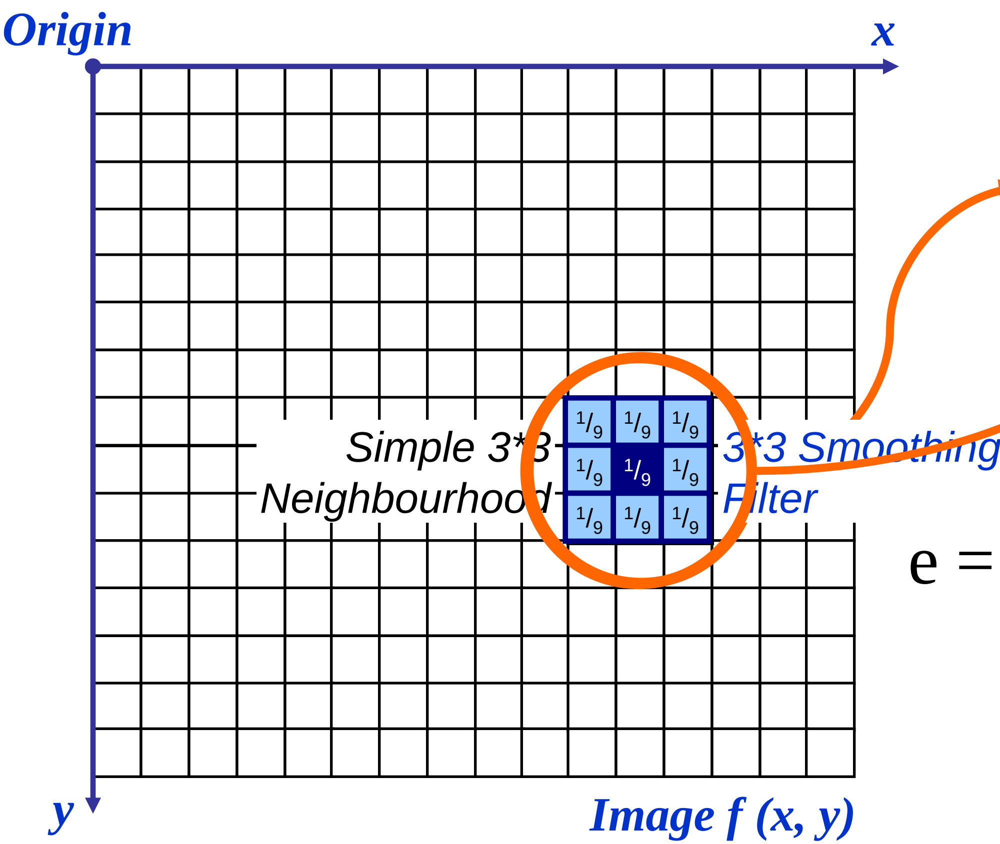
Notations are based on the image shown to the left

Smoothing Spatial Filters

- One of the simplest spatial filtering operations we can perform is a smoothing operation
  - Simply **average** all of the pixels in a neighbourhood around a central value
  - Especially useful in removing noise from images
  - Also useful for highlighting gross detail

$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$

Smoothing Spatial Filtering



104	100	108
99	106	98
95	90	85

Original Image Pixels

$*$

$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$

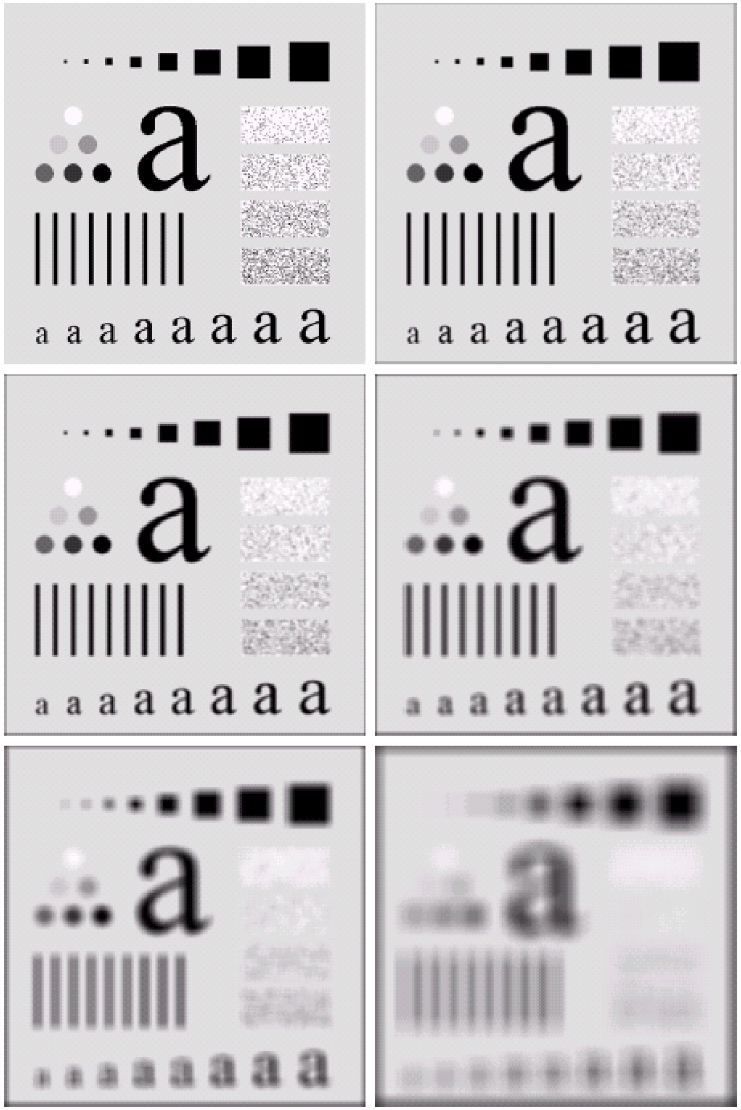
Filter

$$
\begin{aligned}
e = {} & {}^1/_9*106 + \\
& {}^1/_9*104 + {}^1/_9*100 + {}^1/_9*108 + \\
& {}^1/_9*99 + {}^1/_9*98 + \\
& {}^1/_9*95 + {}^1/_9*90 + {}^1/_9*85 \\
= {} & 98.3333
\end{aligned}
$$

The above is repeated for every pixel in the original image to generate the smoothed image.

Image Smoothing Example

- The image at the top left is an original image of size 500*500 pixels
- The subsequent images show the image after filtering with an averaging filter of increasing sizes
  - 3, 5, 9, 15 and 35
- Notice how detail begins to disappear



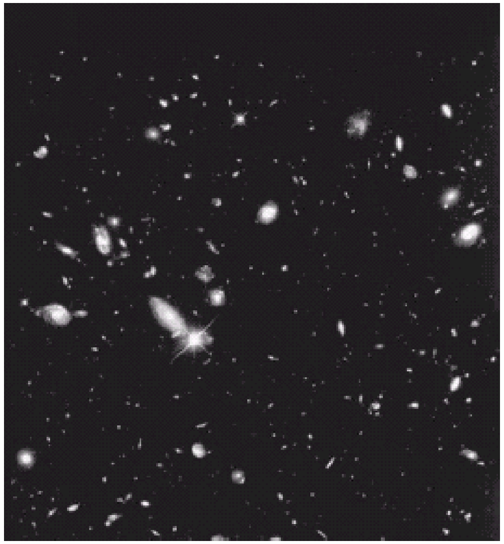
Weighted Smoothing Filters

- More effective smoothing filters can be generated by allowing different pixels in the neighbourhood different weights in the averaging function
  - Pixels closer to the central pixel are more important
  - Often referred to as a *weighted averaging*

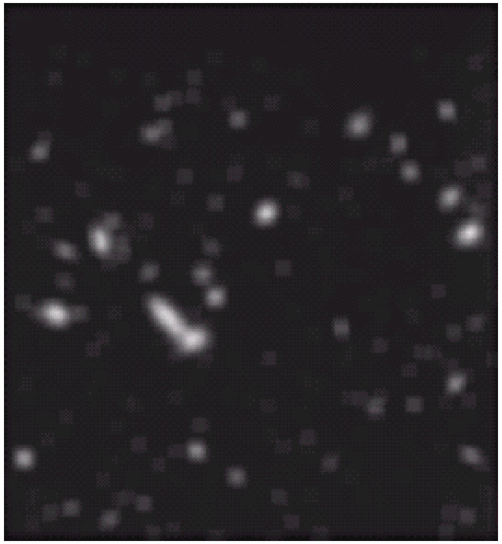
$^1/_{16}$	$^2/_{16}$	$^1/_{16}$
$^2/_{16}$	$^4/_{16}$	$^2/_{16}$
$^1/_{16}$	$^2/_{16}$	$^1/_{16}$

Another Smoothing Example

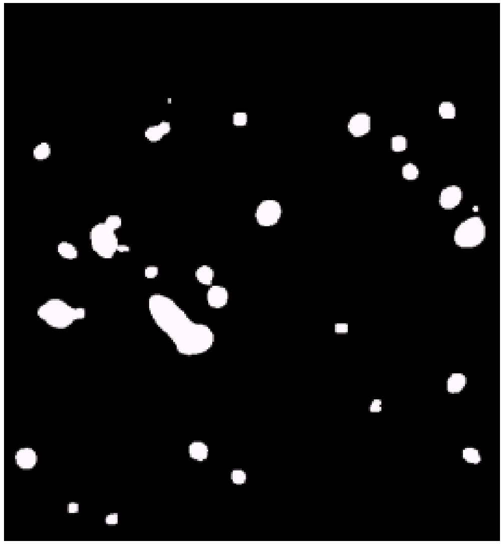
- By smoothing the original image we get rid of lots of the finer detail which leaves only the gross features for thresholding



Original Image

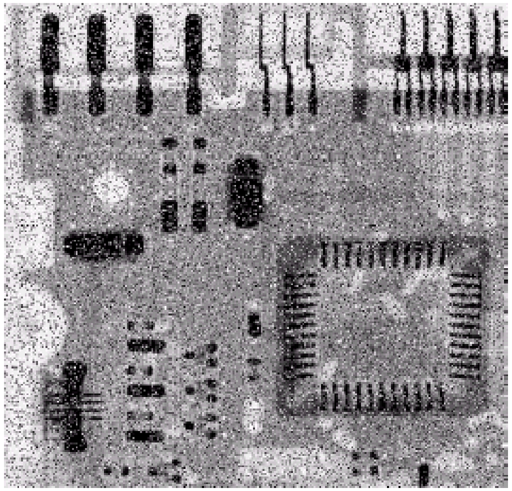


Smoothed Image



Thresholded Image

Averaging Filter vs. Median Filter Example



Original Image With Noise

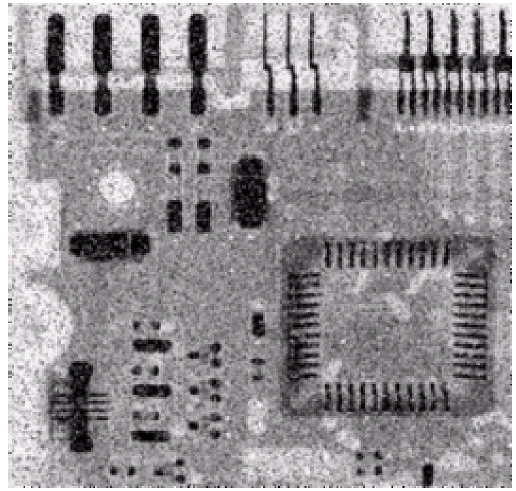


Image After Averaging Filter

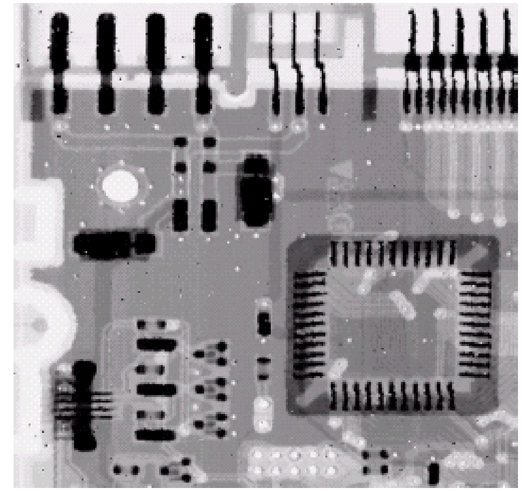


Image After Median Filter

- Filtering is often used to remove noise from images
- Sometimes a median filter works better than an averaging filter

Spatial smoothing and image approximation

- Spatial smoothing may be viewed as a process for estimating the value of a pixel from its neighbours.
- What is the value that “best” approximates the intensity of a given pixel given the intensities of its neighbours?
- We have to define “best” by establishing a criterion.

Spatial smoothing and image approximation (cont...)

A standard criterion is the the sum of squares differences.

$$E = \sum_{i=1}^{N} \left[x(i) - m \right]^2 \Leftrightarrow m = \arg\min_{m} \left\{ \sum_{i=1}^{N} \left[x(i) - m \right]^2 \right\}$$

$$\frac{\partial E}{\partial m} = 0 \Leftrightarrow -2\sum_{i=1}^{N} \left(x(i) - m \right) = 0 \Leftrightarrow \sum_{i=1}^{N} x(i) = \sum_{i=1}^{N} m$$

$$\Leftrightarrow \sum_{i=1}^{N} x(i) = Nm \quad \Leftrightarrow m = \frac{1}{N} \sum_{i=1}^{N} x(i)$$ The average value

Spatial smoothing and image approximation (cont...)

Another criterion is the the sum of absolute differences.

$$E = \sum_{i=1}^{N} \left| x(i) - m \right| \quad \Leftrightarrow m = \arg\min_{m} \left\{ \sum_{i=1}^{N} \left| x(i) - m \right| \right\}$$

$$\frac{\partial E}{\partial m} = 0 \Leftrightarrow -\sum_{i=1}^{N} sgn\left(x(i) - m \right) = 0, \quad sign(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$$

There must be equal in quantity positive and negative values.

$$m = \text{median}\{x(i)\}$$

Spatial smoothing and image approximation (cont...)

- The median filter is non linear:

$$\text{median}\{x+y\} \neq \text{median}\{x\} + \text{median}\{y\}$$

- It works well for impulse noise (e.g. salt and pepper).
- It requires sorting of the image values.
- It preserves the edges better than an average filter in the case of impulse noise.
- It is robust to impulse noise at 50%.

Spatial smoothing and image approximation (cont...)

Example

x[n]	**1**	**1**	**1**	**1**	**1**	**2**	**2**	**2**	**2**	**2**

edge

Impulse noise

x[n]	**1**	**3**	**1**	**1**	**1**	**2**	**3**	**2**	**2**	**3**

Median (N=3)

x[n]	**-**	**1**	**1**	**1**	**1**	**2**	**2**	**2**	**2**	**-**

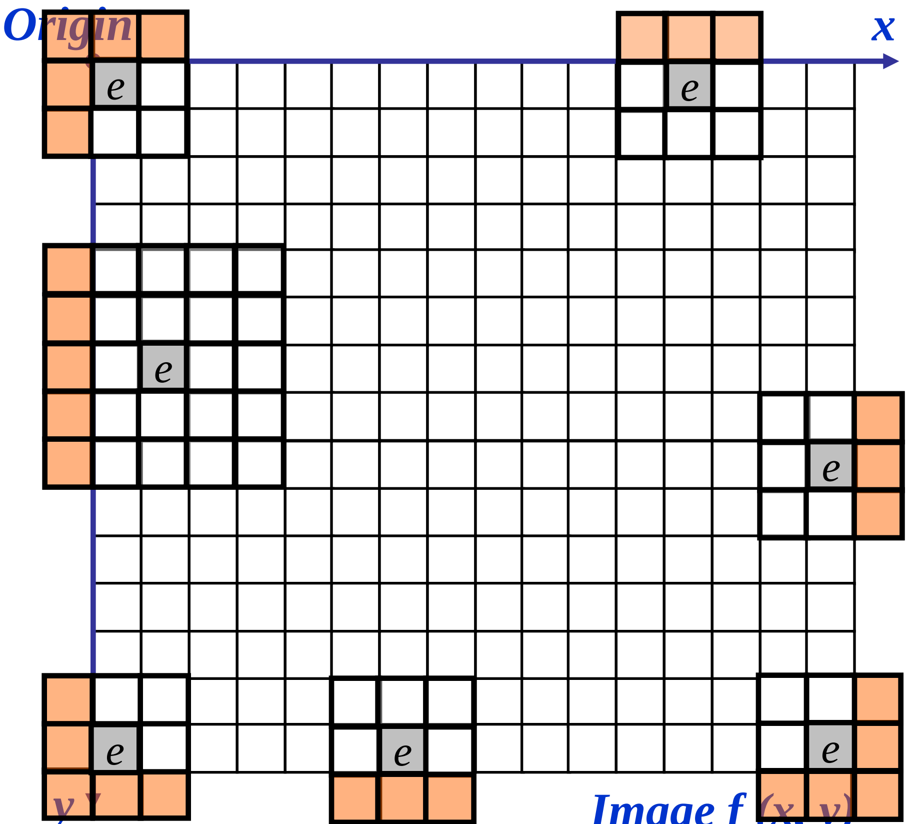
Average (N=3)

x[n]	**-**	**1.7**	**1.7**	**1**	**1.3**	**2**	**2.3**	**2.3**	**2.2**	**-**

The edge is smoothed

Strange Things Happen At The Edges!

At the edges of an image we are missing pixels to form a neighbourhood



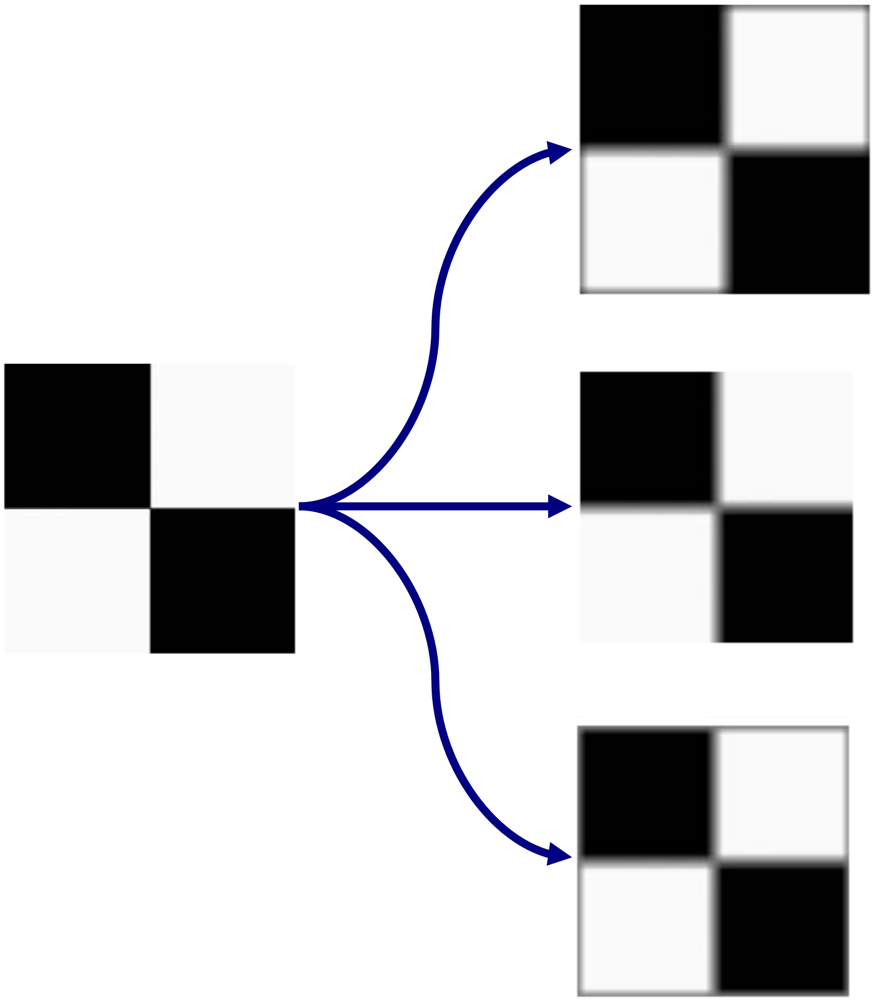
Strange Things Happen At The Edges! (cont…)

There are a few approaches to dealing with missing edge pixels:

- Omit missing pixels
  - Only works with some filters
  - Can add extra code and slow down processing
- Pad the image
  - Typically with either all white or all black pixels
- Replicate border pixels
- Truncate the image
- Allow pixels *wrap around* the image
  - Can cause some strange image artefacts

Strange Things Happen At The Edges! (cont…)

Original Image



Filtered Image: Zero Padding

Filtered Image: Replicate Edge Pixels

Filtered Image: Wrap Around Edge Pixels

Correlation & Convolution

- The filtering we have been talking about so far is referred to as *correlation* with the filter itself referred to as the *correlation kernel*
- *Convolution* is a similar operation, with just one subtle difference

a	b	c
d	e	e
f	g	h

Original Image Pixels

$*$

r	s	t
u	v	w
x	y	z

Filter

$$\begin{aligned} e_{processed} = {} & v*e + \\ & z*a + y*b + x*c + \\ & w*d + u*e + \\ & t*f + s*g + r*h \end{aligned}$$

- For symmetric filters it makes no difference.

Correlation & Convolution (cont.)

Correlation

Origin f w

(a) 0 0 0 1 0 0 0 0 1 2 3 2 8

(b) 0 0 0 1 0 0 0 0
1 2 3 2 8
Starting position alignment

Zero padding

(c) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
1 2 3 2 8

(d) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
1 2 3 2 8
Position after one shift

(e) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
1 2 3 2 8
Position after four shifts

(f) 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
1 2 3 2 8
Final position

Full correlation result

(g) 0 0 0 8 2 3 2 1 0 0 0 0

Cropped correlation result

(h) 0 8 2 3 2 1 0 0

Convolution

Origin f w rotated 180°

0 0 0 1 0 0 0 0 8 2 3 2 1 (i)

0 0 0 1 0 0 0 0 (j)
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (k)
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (l)
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (m)
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (n)
8 2 3 2 1

Full convolution result

0 0 0 1 2 3 2 8 0 0 0 0 (o)

Cropped convolution result

0 1 2 3 2 8 0 0 (p)

FIGURE 3.29 Illustration of 1-D correlation and convolution of a filter with a discrete unit impulse. Note that correlation and convolution are functions of *displacement*.

Correlation & Convolution (cont.)

Origin $f(x, y)$

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

$w(x, y)$

1	2	3
4	5	6
7	8	9

(a)

Padded f

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(b)

Initial position for w

1	2	3	0	0	0	0	0	0
4	5	6	0	0	0	0	0	0
7	8	9	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(c)

Full correlation result

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	9	8	7	0	0	0
0	0	0	6	5	4	0	0	0
0	0	0	3	2	1	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(d)

Cropped correlation result

0	0	0	0	0
0	9	8	7	0
0	6	5	4	0
0	3	2	1	0
0	0	0	0	0

(e)

Rotated w

9	8	7	0	0	0	0	0	0
6	5	4	0	0	0	0	0	0
3	2	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(f)

Full convolution result

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	1	2	3	0	0	0
0	0	0	4	5	6	0	0	0
0	0	0	7	8	9	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(g)

Cropped convolution result

0	0	0	0	0
0	1	2	3	0
0	4	5	6	0
0	7	8	9	0
0	0	0	0	0

(h)

Effect of Low Pass Filtering on White Noise

Let f be an observed instance of the image f_0 corrupted by noise w:

$$f = f_0 + w$$

with noise samples having mean value $E[w(n)]=0$ and being uncorrelated with respect to location:

$$E[w(m)w(n)] = \begin{cases} \sigma^2, & m = n \\ 0, & m \neq n \end{cases}$$

Effect of Low Pass Filtering on White Noise (cont...)

Applying a low pass filter h (e.g. an average filter) by convolution to the degraded image:

$$g = h * f = h * (f_0 + w) = h * f_0 + h * w$$

The expected value of the output is:

$$E[g] = E[h * f_0] + E[h * w] = h * f_0 + h * E[w]$$

$$= h * f_0 + h * 0 = h * f_0$$

The noise is removed in average.

Effect of Low Pass Filtering on White Noise (cont...)

What happens to the standard deviation of g?

Let $g = h * f_0 + h * w = \overline{f_0} + \overline{w}$

where the bar represents filtered versions of the signals, then

$$\sigma_g^2 = E[g^2] - \left(E[g]\right)^2 = \mathrm{E}[(\overline{f_0} + \overline{w})^2] - (\overline{f_0})^2$$

$$= \mathrm{E}[(\overline{f_0})^2 + (\overline{w})^2 + 2\overline{f_0}\,\overline{w}] - (\overline{f_0})^2$$

$$= E[(\overline{w})^2] + 2E[\overline{f_0}]E[\overline{w}] = E[(\overline{w})^2]$$

Effect of Low Pass Filtering on White Noise (cont...)

Considering that h is an average filter, we have at pixel n:

$$\overline{w}(n) = (h * w)(n) = \frac{1}{N} \sum_{k \in \Gamma(n)} w(k)$$

Therefore,

$$E[(\overline{w}(n))^2] = E\left[\left(\frac{1}{N} \sum_{k \in \Gamma(n)} w(k) \right)^2 \right]$$

Effect of Low Pass Filtering on White Noise (cont...)

$$E\left[\left(\frac{1}{N}\sum_{k\in\Gamma(n)} w(k)\right)^2\right]$$

$$=\frac{1}{N^2}\sum_{k\in\Gamma(n)} E\left[\left\{w(k)\right\}^2\right]$$

Sum of squares

$$+\frac{2}{N^2}\sum_{l\in\Gamma(n)}\sum_{\substack{m\in\Gamma(n)\\ m\neq l}} E\left[w(n-l)w(n-m)\right]$$

Cross products

Effect of Low Pass Filtering on White Noise (cont...)

Sum of squares

$$\frac{1}{N^2}\sum_{k\in\Gamma(n)} E\left[\left\{w(k)\right\}^2\right] = \frac{1}{N^2}\sum_{k\in\Gamma(n)} \sigma^2$$

Cross products (uncorrelated as $m \neq l$)

$$+\frac{2}{N^2}\sum_{l\in\Gamma(n)}\sum_{\substack{m\in\Gamma(n)\\ m\neq l}} E\left[w(n-l)w(n-m)\right] = 0$$

Effect of Low Pass Filtering on White Noise (cont...)

Finally, substituting the partial results:

$$\sigma_g^2 = E\left[\left(\frac{1}{N}\sum_{k\in\Gamma(n)} w(k)\right)^2\right] = \frac{1}{N^2}\sum_{k\in\Gamma(n)} \sigma^2$$

$$= \frac{1}{N^2} N\sigma^2 = \frac{\sigma^2}{N}$$

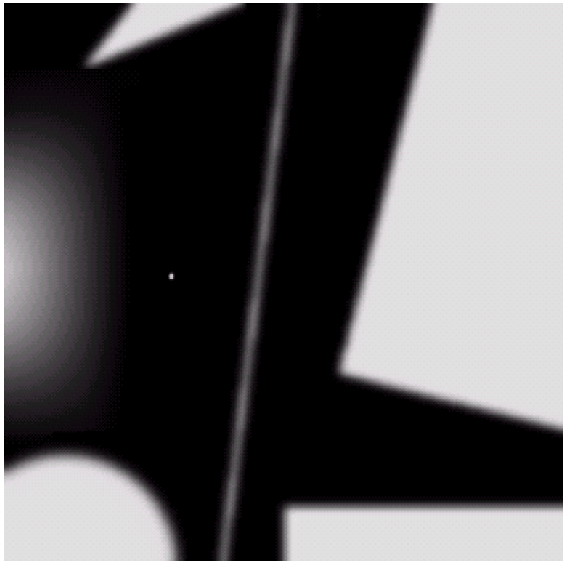
The effect of the noise is reduced.
This processing is not optimal as it also smoothes image edges.

Sharpening Spatial Filters

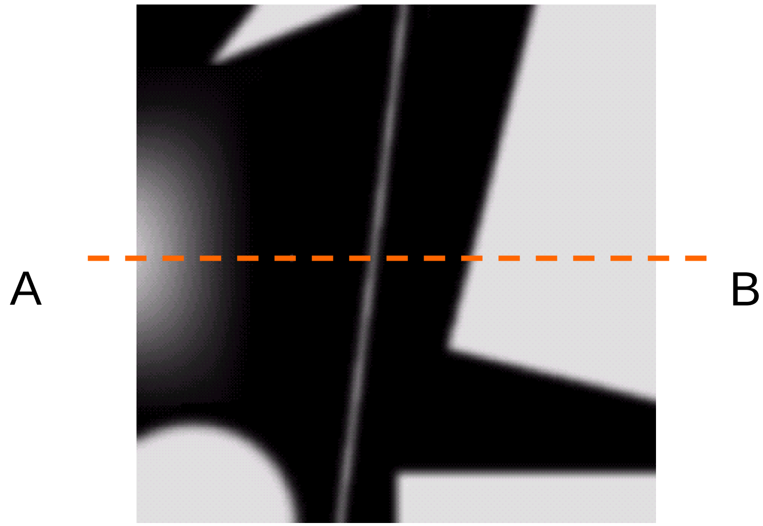
- Previously we have looked at smoothing filters which remove fine detail
- *Sharpening spatial filters* seek to highlight fine detail
  - Remove blurring from images
  - Highlight edges
- Sharpening filters are based on spatial differentiation

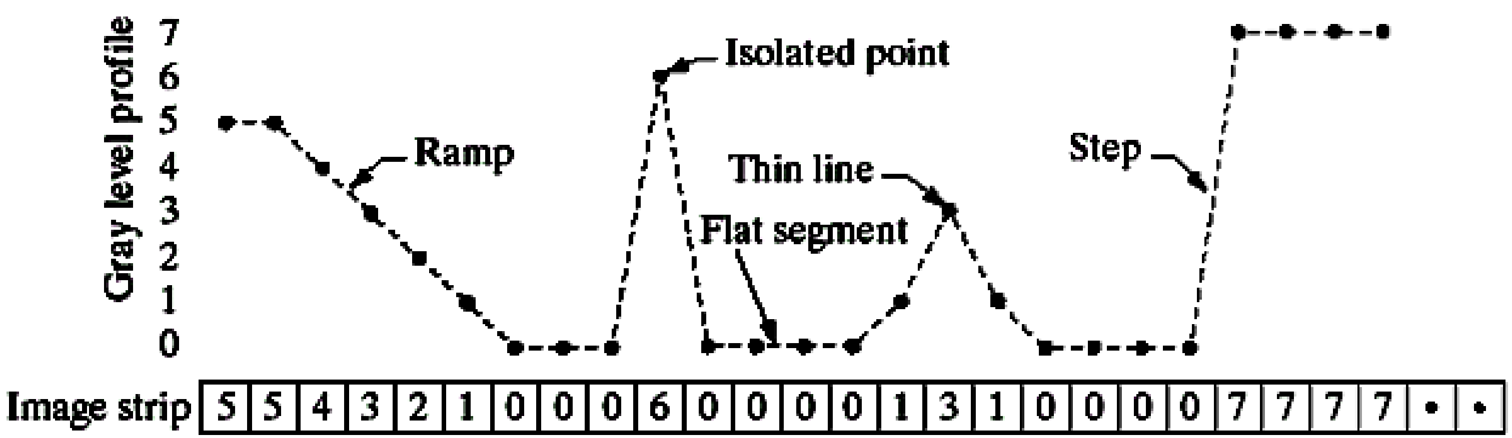
Spatial Differentiation

- Differentiation measures the rate of change of a function
- Let's consider a simple 1 dimensional example



Spatial Differentiation





Derivative Filters Requirements

- First derivative filter output
  - Zero at constant intensities
  - Non zero at the onset of a step or ramp
  - Non zero along ramps

- Second derivative filter output
  - Zero at constant intensities
  - Non zero at the onset and end of a step or ramp
  - Zero along ramps of constant slope

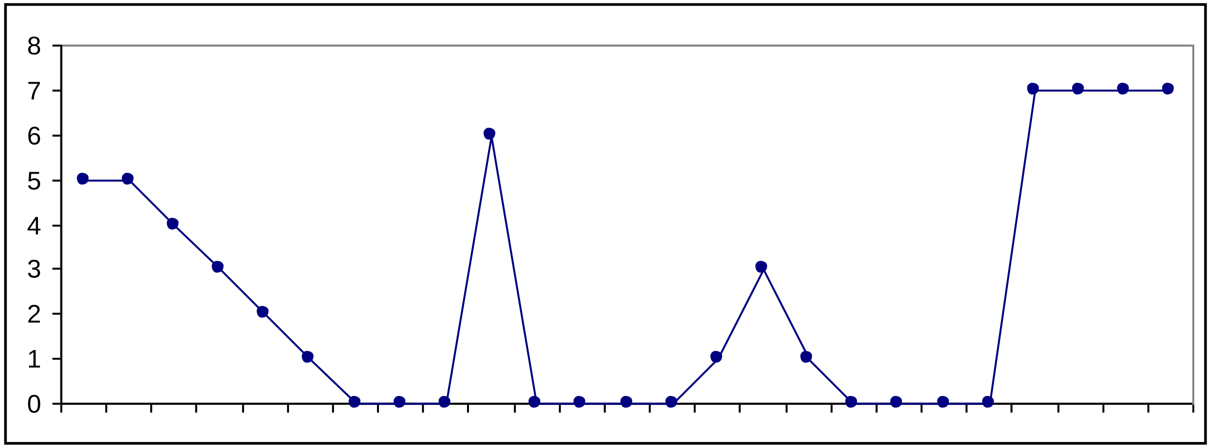
1^{st} Derivative

- Discrete approximation of the 1st derivative

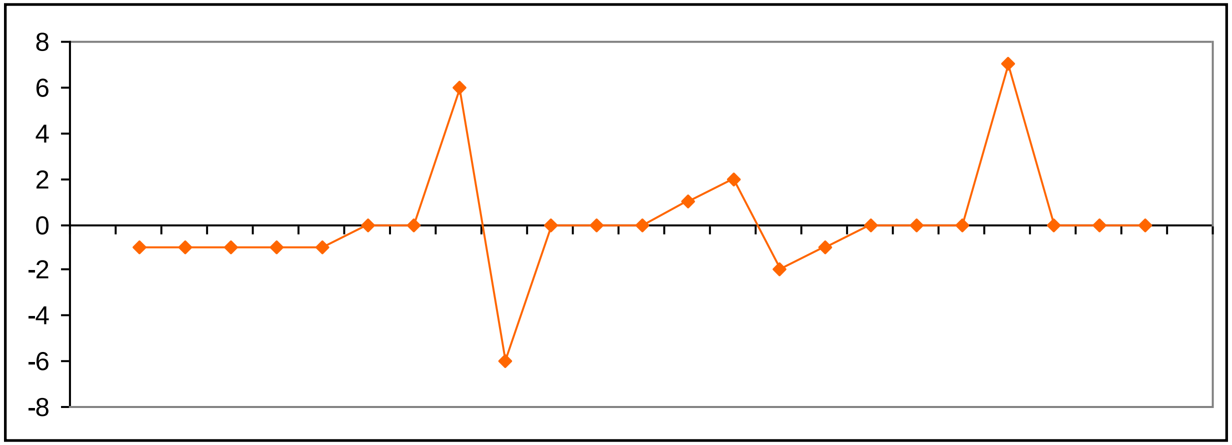
$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

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

1st Derivative (cont…)

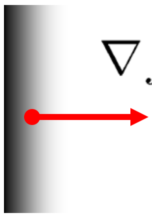


5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7
	0	-1	-1	-1	-1	-1	0	0	6	-6	0	0	0	1	2	-2	-1	0	0	0	7	0	0	0

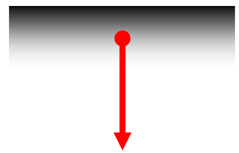


1st Derivative (cont.)

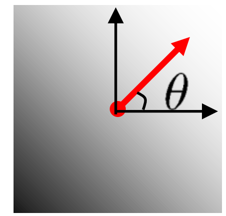
- The gradient of an image: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

- 

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



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



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

The gradient points in the direction of most rapid increase in intensity.

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

The *edge strength* is given by the gradient magnitude

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

Source: Steve Seitz

1^{st} Derivative (cont.)

$$\|\nabla f\|$$

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

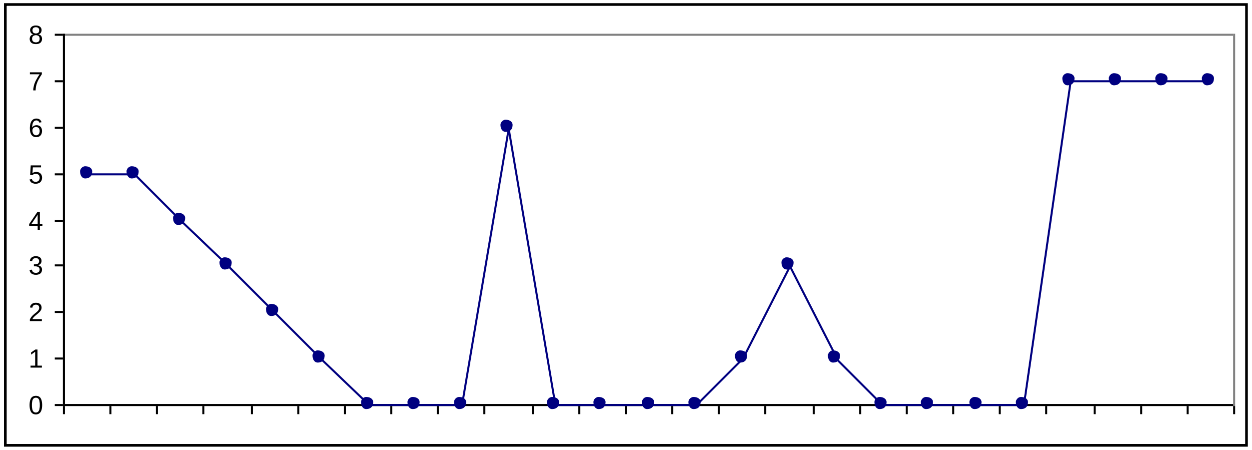
$$\frac{\partial f}{\partial y}$$

2^{nd} Derivative

- Discrete approximation of the 2^{nd} derivative:

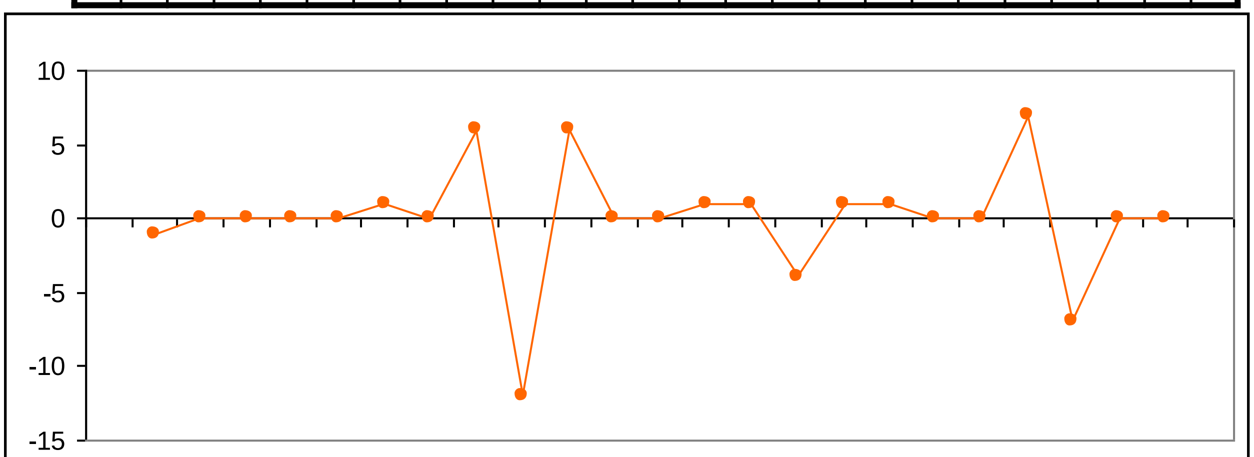
$$\frac{\partial^2 f}{\partial^2 x} = f(x+1) + f(x-1) - 2f(x)$$

2^{nd} Derivative (cont…)



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

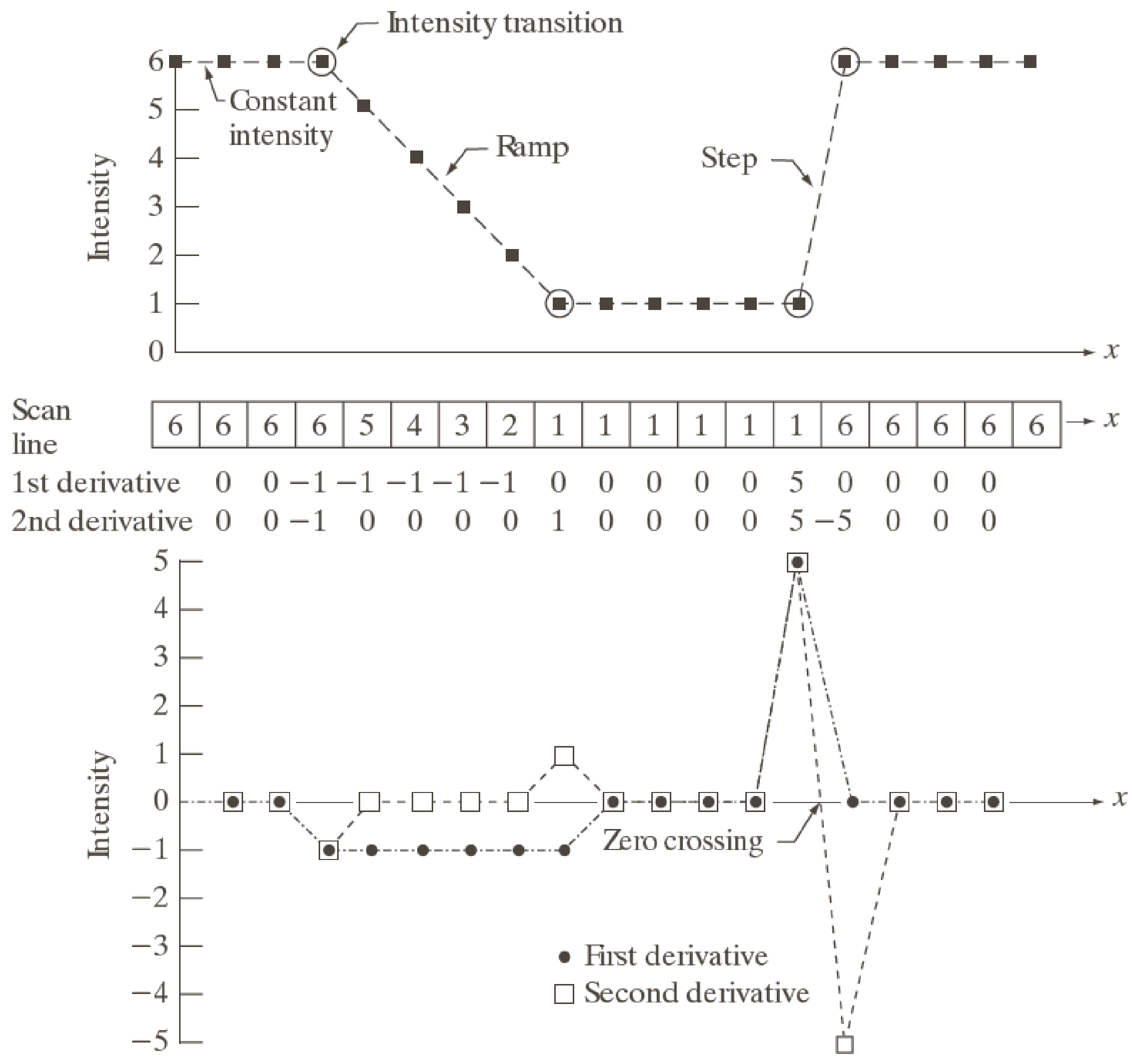
	-1	0	0	0	0	1	0	6	-12	6	0	0	1	1	-4	1	1	0	0	7	-7	0	0	



Using Second Derivatives For Image Enhancement

- Edges in images are often ramp-like transitions
  - 1st derivative is constant and produces thick edges
  - 2nd derivative zero crosses the edge (double response at the onset and end with opposite signs)

Derivatives



Using Second Derivatives For Image Enhancement

- A common sharpening filter is the **Laplacian**
  - Isotropic
    - Rotation invariant: Rotating the image and applying the filter is the same as applying the filter and then rotating the image.
    - In other words, the Laplacian of a rotated image is the rotated Laplacian of the original image.
  - One of the simplest sharpening filters
  - We will look at a digital implementation

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

The Laplacian

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

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

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

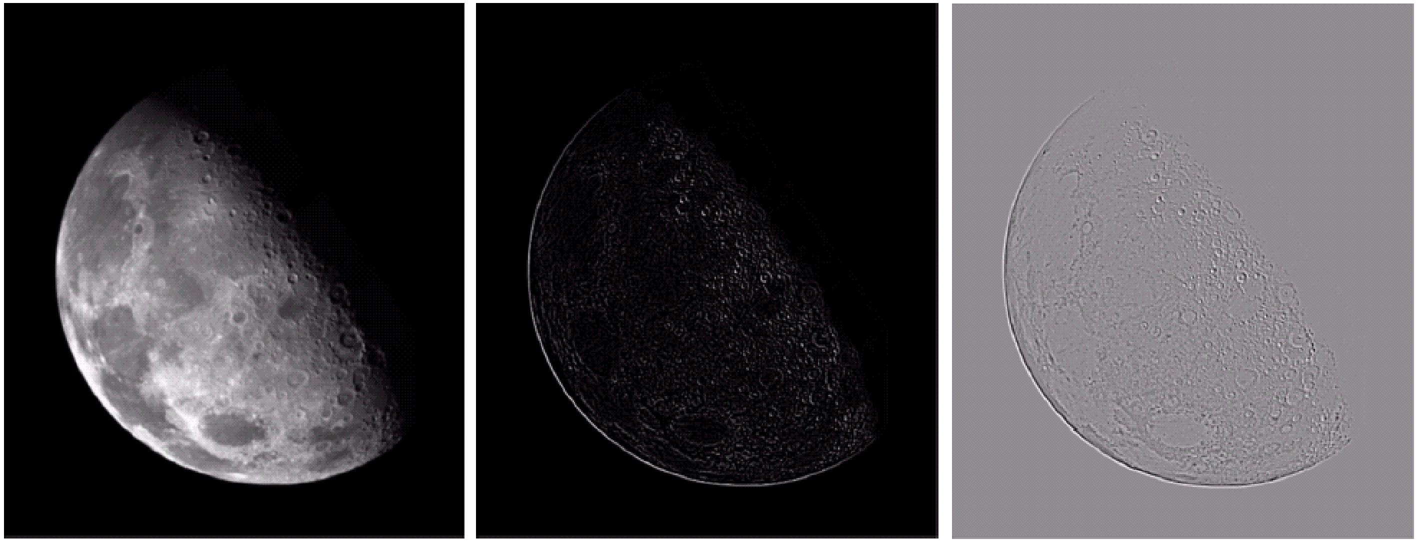
The Laplacian (cont…)

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

0	1	0
1	-4	1
0	1	0

The Laplacian (cont…)

- Applying the Laplacian to an image we get a new image that highlights edges and other discontinuities



Original Image

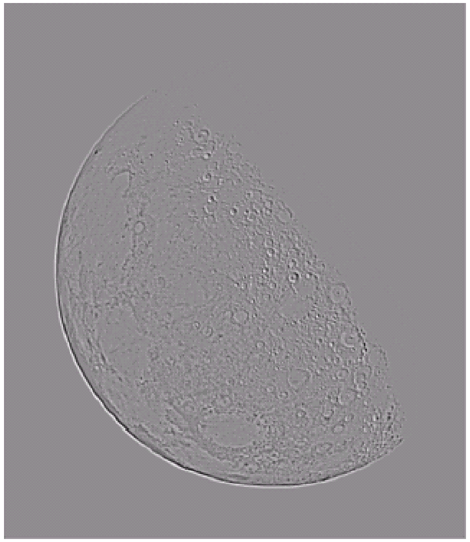
Laplacian Filtered Image

Laplacian Filtered Image Scaled for Display

The Laplacian (cont…)

- The result of a Laplacian filtering is not an enhanced image
- We have to do more work
- Subtract the Laplacian result from the original image to generate our final sharpened enhanced image

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

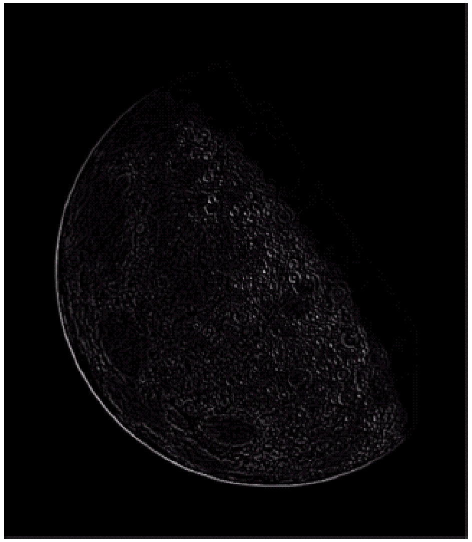


Laplacian Filtered Image Scaled for Display

Laplacian Image Enhancement

Original Image

-



Laplacian Filtered Image

=

Sharpened Image

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

Laplacian Image Enhancement

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

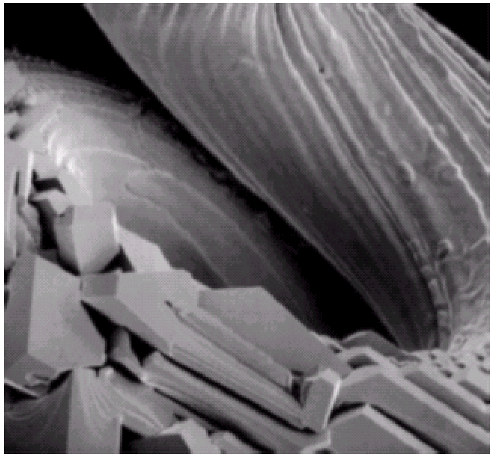
Simplified Image Enhancement

- The entire enhancement can be combined into a single filtering operation:

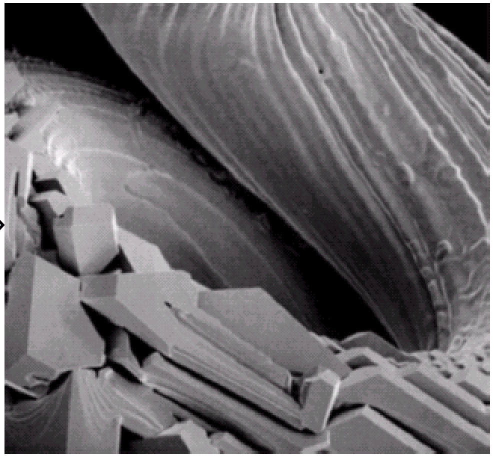
$$
\begin{aligned}
g(x,y) &= f(x,y) - \nabla^2 f \\
&= 5f(x,y) - f(x+1,y) - f(x-1,y) \\
&\quad - f(x,y+1) - f(x,y-1)
\end{aligned}
$$

Simplified Image Enhancement (cont…)

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

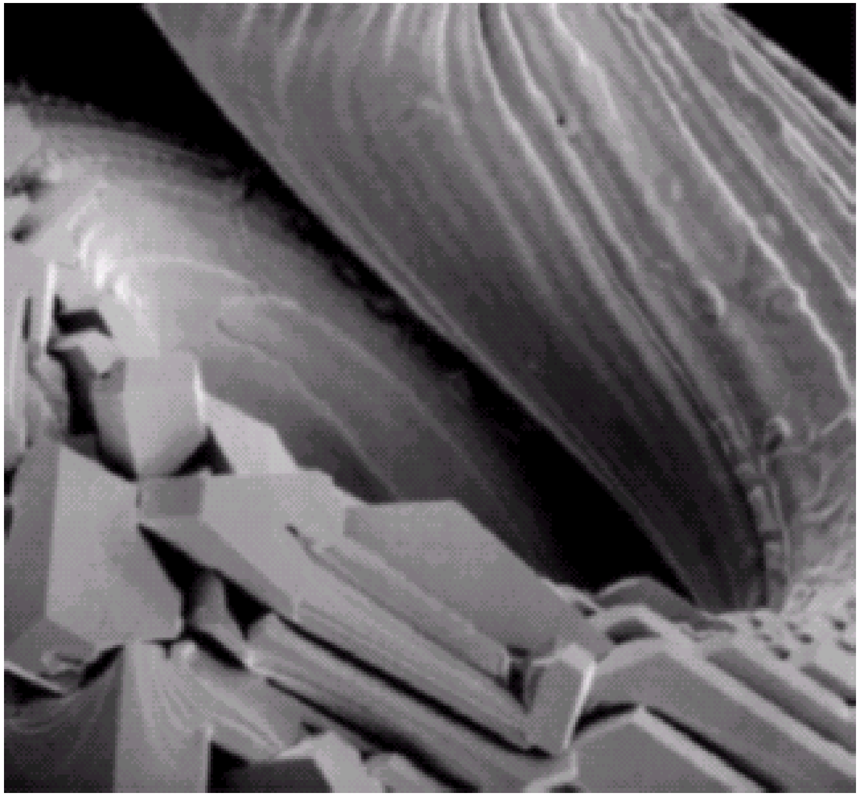


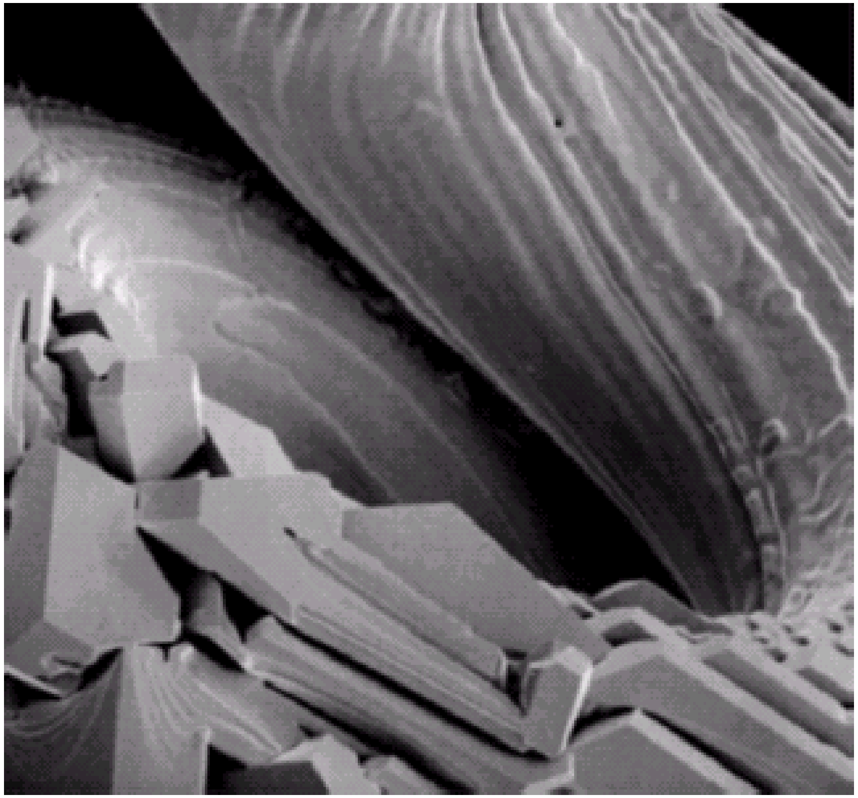
0	-1	0
-1	5	-1
0	-1	0



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Simplified Image Enhancement (cont…)





Variants On The Simple Laplacian

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

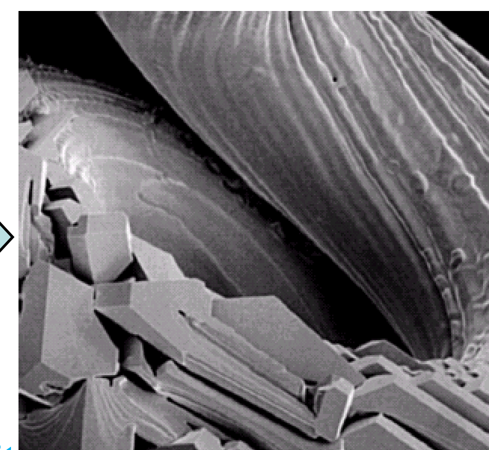
0	1	0
1	-4	1
0	1	0

Standard Laplacian

1	1	1
1	-8	1
1	1	1

Variant of Laplacian

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



Unsharp masking

- Used by the printing industry
- Subtracts an unsharped (smooth) image from the original image $f(x,y)$.
  - Blur the image

    $$b(x,y)=Blur\{f(x,y)\}$$

  - Subtract the blurred image from the original (the result is called the *mask*)

    $$g_{mask}(x,y)=f(x,y)-b(x,y)$$

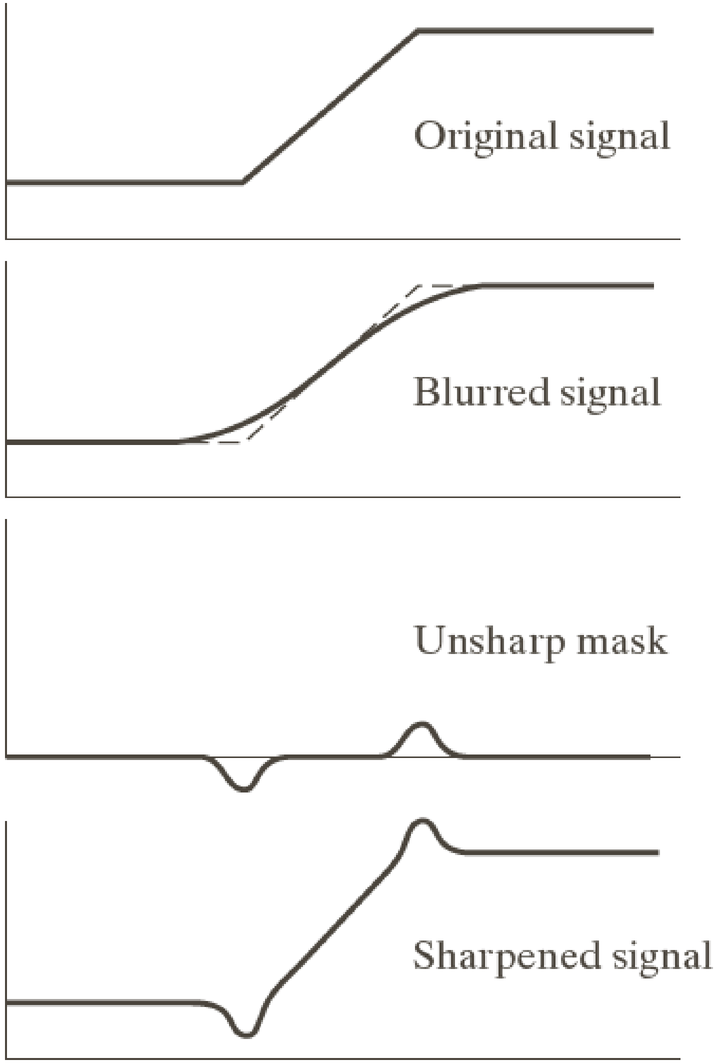
  - Add the mask to the original

    $g(x,y)=f(x,y)+k\,g_{mask}(x,y)$, k being non negative

Unsharp masking (cont...)

Sharpening mechanism

If $k>1$, the process is referred to as **highboost filtering**



Unsharp masking (cont...)

Original image

Blurred image
(Gaussian 5x5, σ=3)

Mask

Unsharp masking (k=1)

Highboost filtering (k=4.5)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Using First Derivatives For Image Enhancement

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

- Although the derivatives are linear operators, the gradient magnitude is not.
- Also, the partial derivatives are not rotation invariant (isotropic).
- The magnitude of the gradient vector is isotropic.

Using First Derivatives For Image Enhancement (cont…)

- In some applications it is more computationally efficient to approximate:

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

- This expression preserves relative changes in intensity but it is not isotropic.
- Isotropy is preserved only for a limited number of rotational increments which depend on the filter masks (e.g. 90 deg.).

Sobel Operators

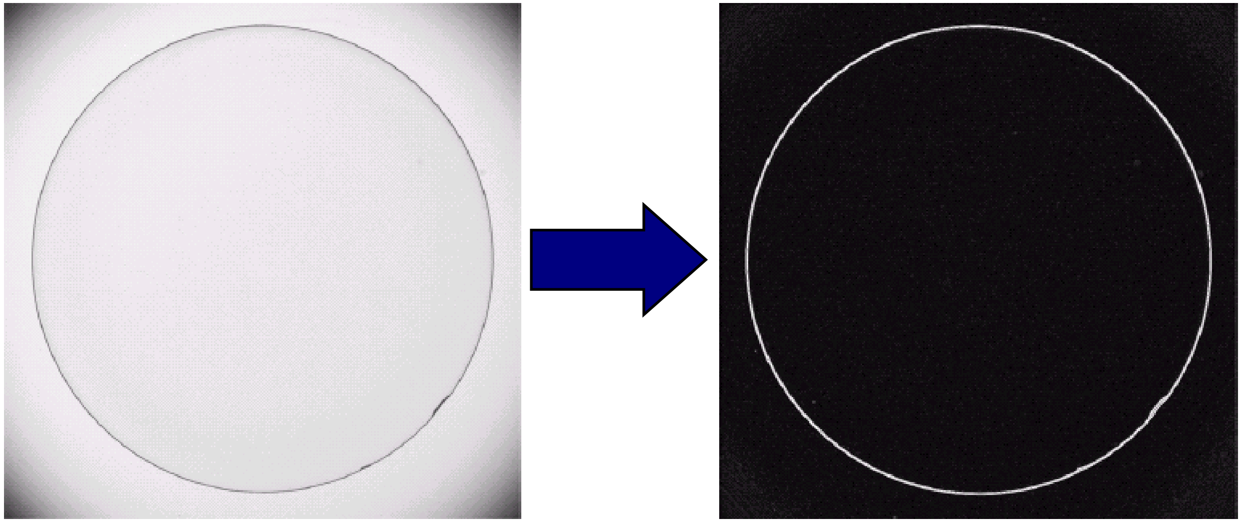
- **Sobel operators** introduce the idea of smoothing by giving more importance to the center point:

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

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

- Note that the coefficients sum to 0 to give a 0 response at areas of constant intensity.

Sobel operator Example



An image of a contact lens which is enhanced in order to make defects (at four and five o'clock in the image) more obvious

- Sobel gradient aids to eliminate constant or slowly varying shades of gray and assist automatic inspection.
- It also enhances small discontinuities in a flat gray filed.

1st & 2nd Derivatives

Comparing the 1st and 2nd derivatives we can conclude the following:

- 1st order derivatives generally produce thicker edges (if thresholded at ramp edges)
- 2nd order derivatives have a stronger response to fine detail e.g. thin lines
- 1st order derivatives have stronger response to gray level step
- 2nd order derivatives produce a double response at step changes in grey level (which helps in detecting zero crossings)

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 to the right

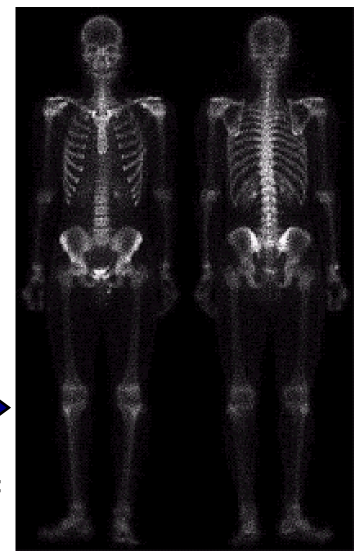
Combining Spatial Enhancement Methods (cont…)

(a)

Laplacian filter of bone scan (a)

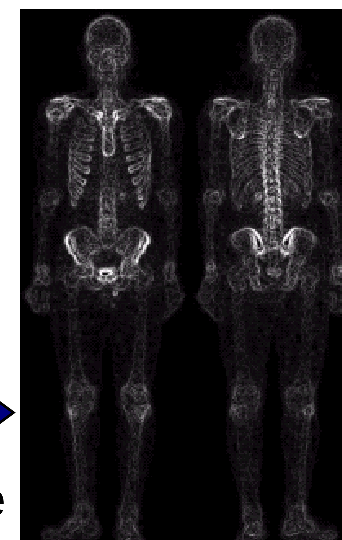
(b)

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



(c)

Sobel filter of bone scan (a)



(d)

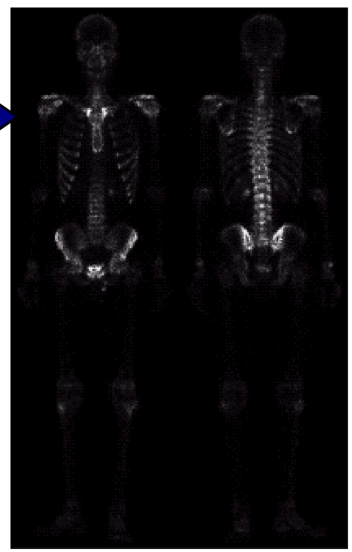
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Combining Spatial Enhancement Methods (cont…)

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

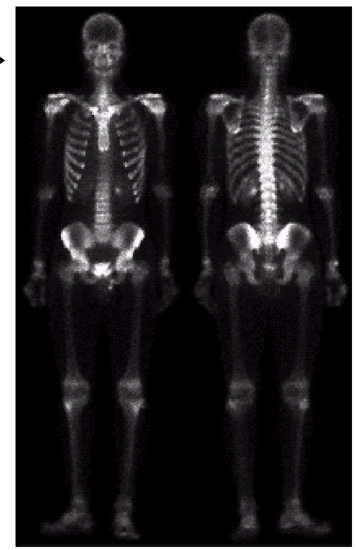
(e)

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 (cont…)

Compare the original and final images

Summary

In this lecture we have looked at the idea of spatial filtering and in particular:

- Neighbourhood operations
- The filtering process
- Smoothing filters
- Dealing with problems at image edges when using filtering
- Correlation and convolution
- Sharpening filters
- Combining filtering techniques