ज्ञानेन प्रकाशते जगत्

INDUS UNIVERSITY

Image Processing (EC0703)

Unit-2

B.Tech (Electronics and Communication)

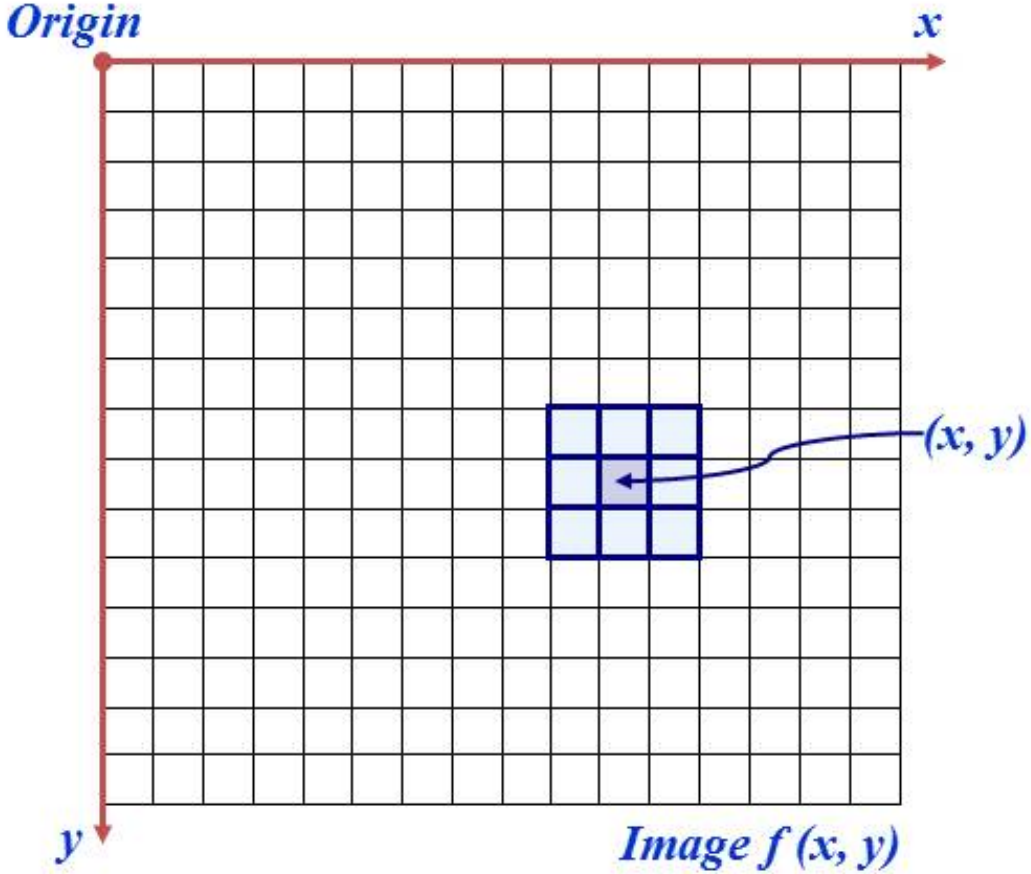
Semester-7

Prof.Hansa Shingrakhia

Academic Year 2019-2020

Basic Spatial Domain Image Enhancement

- Most spatial domain enhancement operations can be reduced to the form
- $g(x, y) = T[f(x, y)]$
- where $f(x, y)$ is the input image, $g(x, y)$ is the processed image and T is some operator defined over some neighbourhood of (x, y)

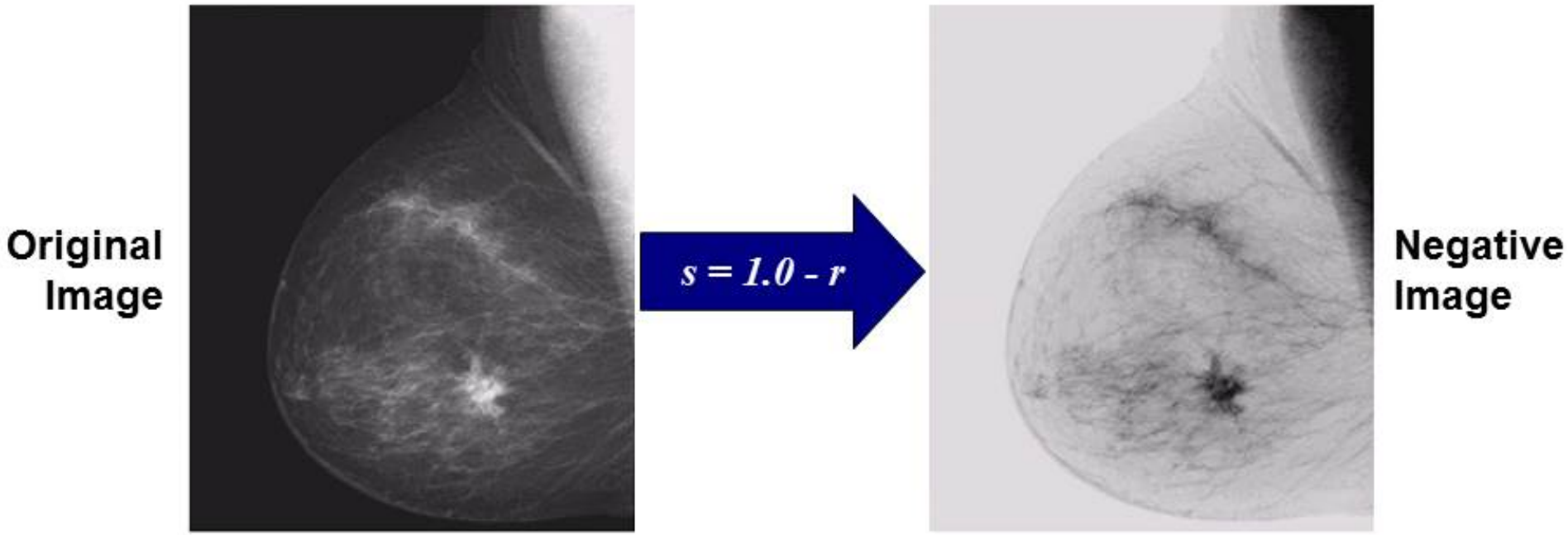


Point Processing

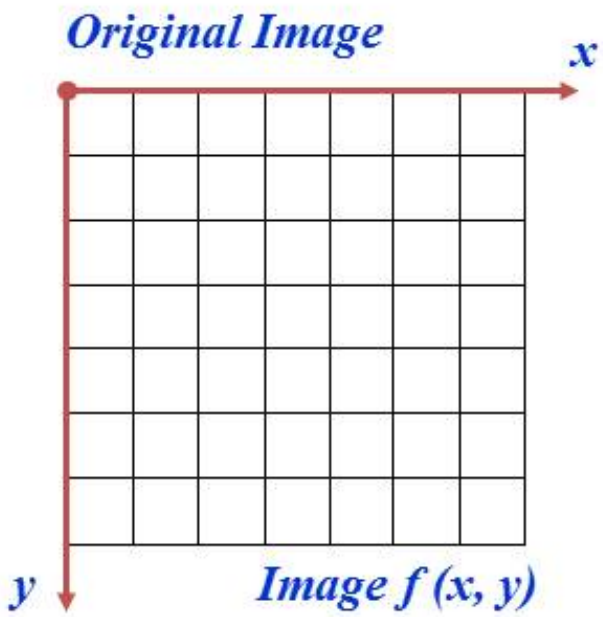
- The simplest spatial domain operations occur when the neighbourhood is simply the pixel itself
- In this case T is referred to as a *grey level transformation function* or a *point processing operation*
- Point processing operations take the form
  - $s = T(r)$
- where s refers to the processed image pixel value and r refers to the original image pixel value

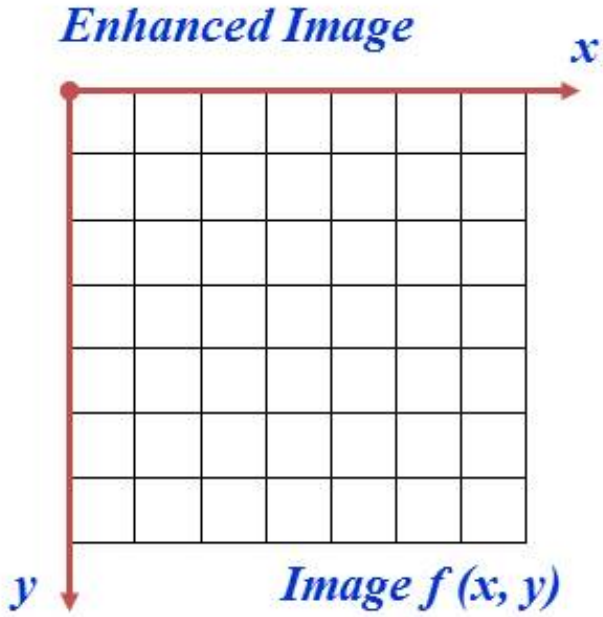
Point Processing Example: Negative Images

- Negative images are useful for enhancing white or grey detail embedded in dark regions of an image
  - Note how much clearer the tissue is in the negative image of the mammogram below



Point Processing Example: Negative Images (cont…)

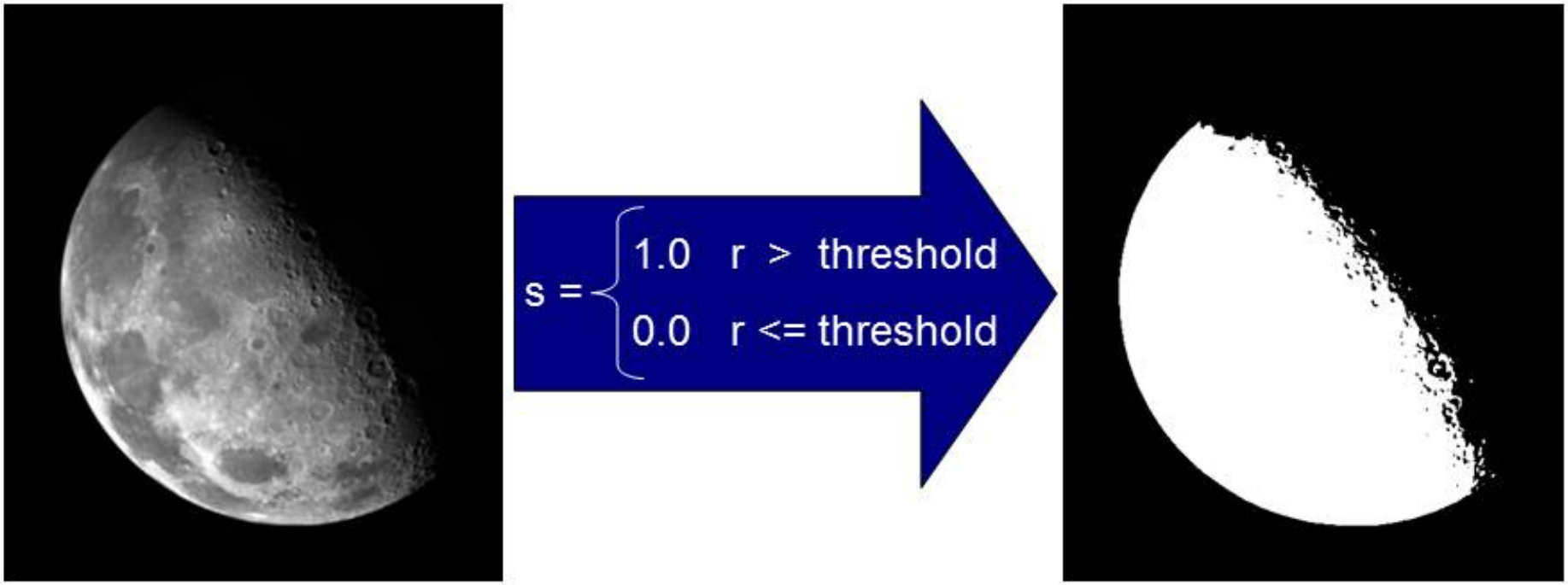




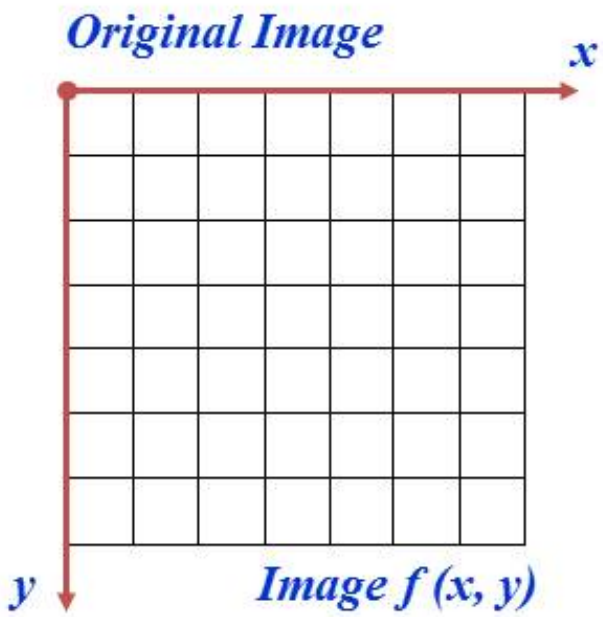
$$s = intensity_{max} - r$$

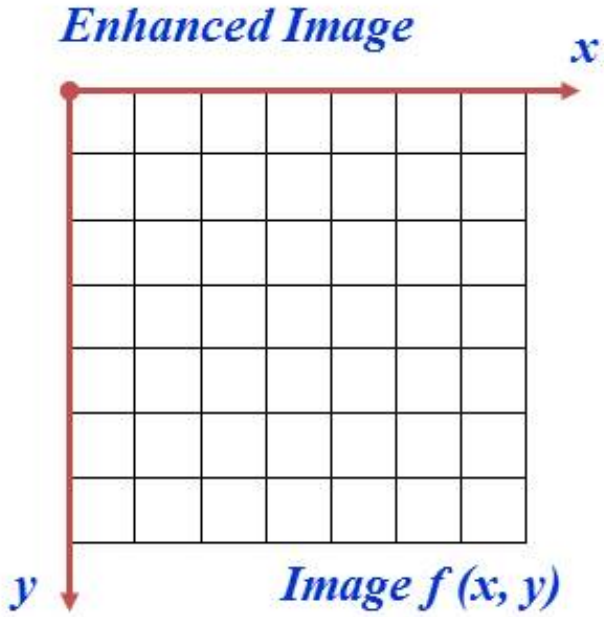
Point Processing Example: Thresholding

- Thresholding transformations are particularly useful for segmentation in which we want to isolate an object of interest from a background



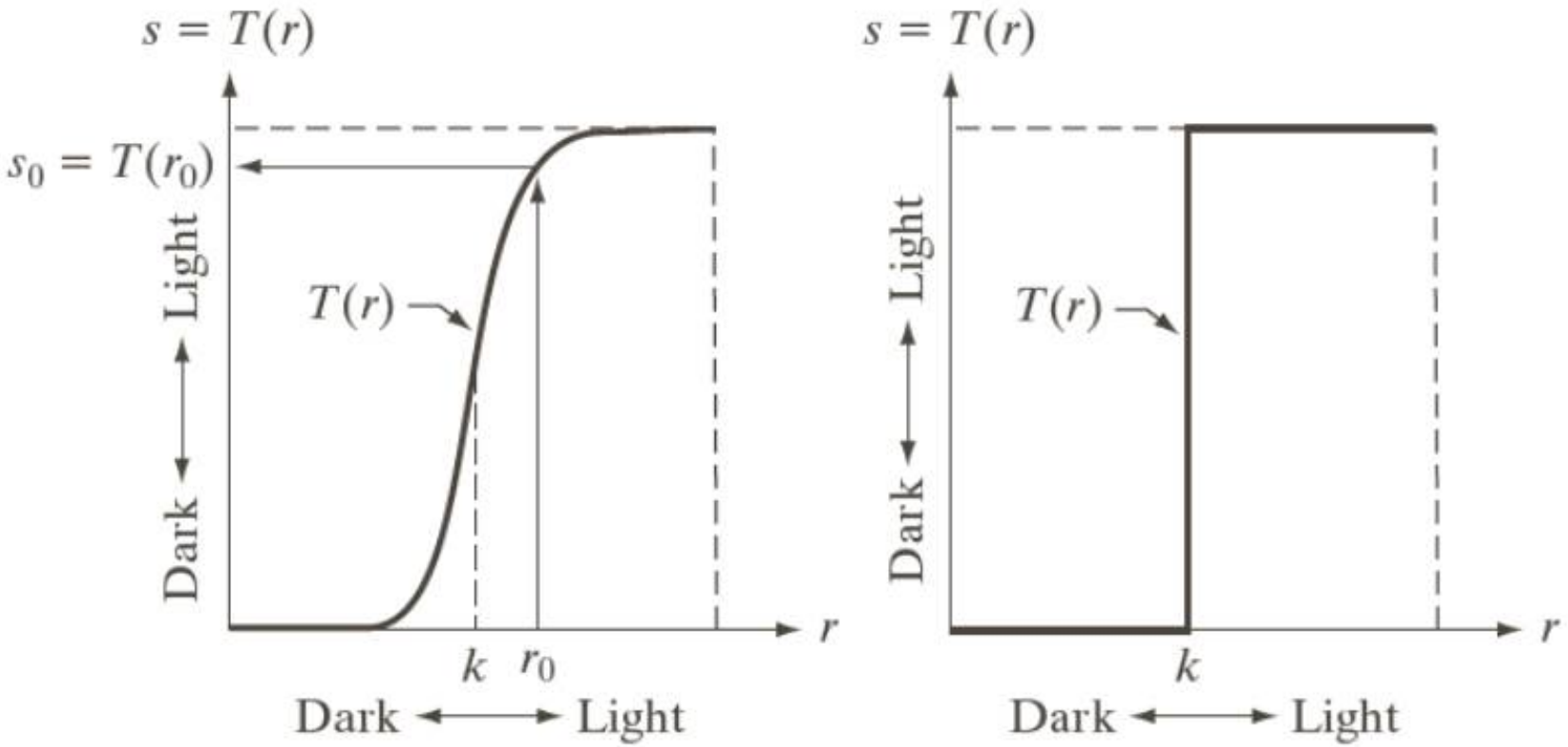
Point Processing Example: Thresholding (cont…)





$$s = \begin{cases} 1.0 & r > threshold \\ 0.0 & r <= threshold \end{cases}$$

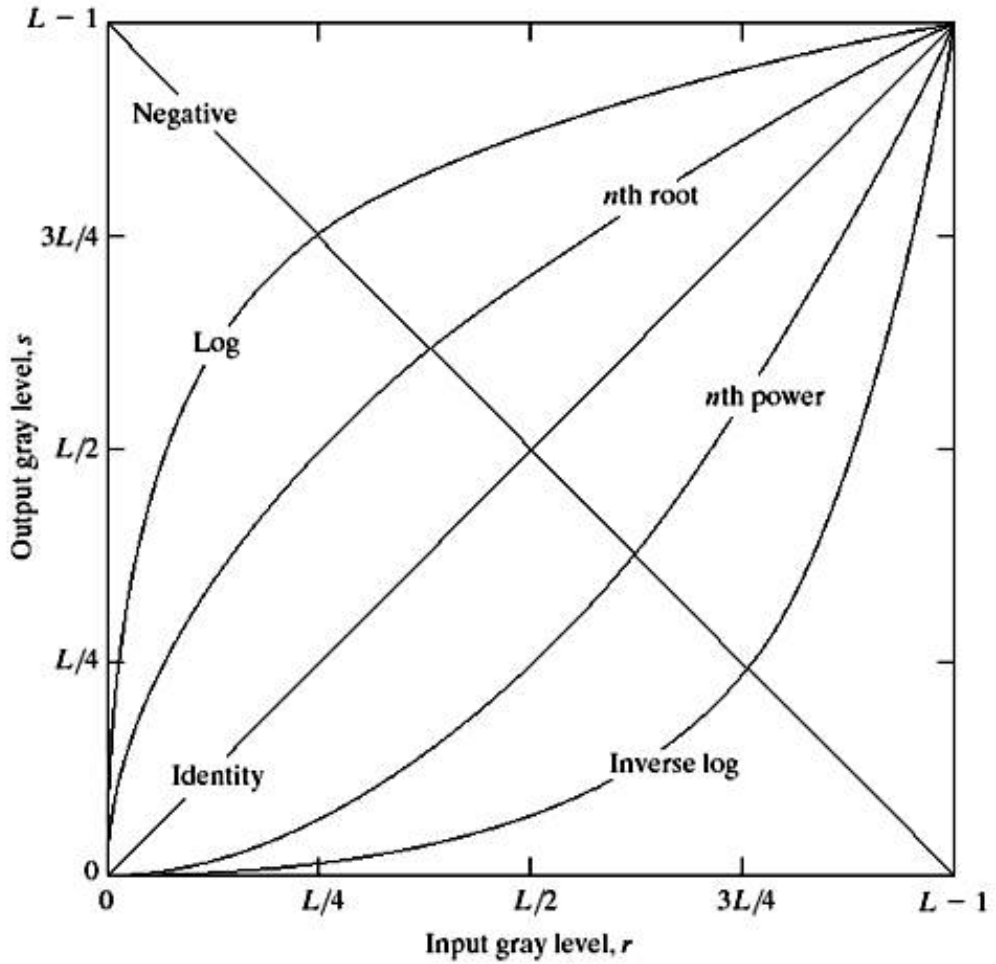
Intensity Transformations



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

Basic Grey Level Transformations

- There are many different kinds of grey level transformations
- Three of the most common are shown here
  - Linear
    - Negative/Identity
  - Logarithmic
    - Log/Inverse log
  - Power law
    - n^{th} power/n^{th} root

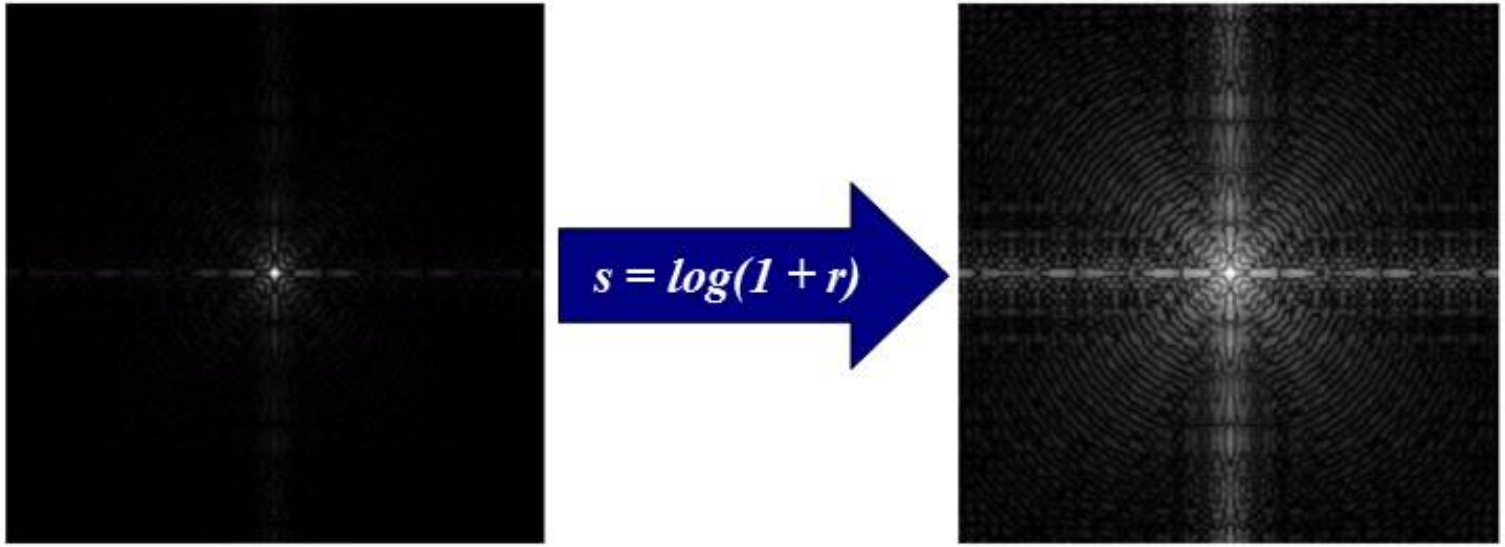


Logarithmic Transformations

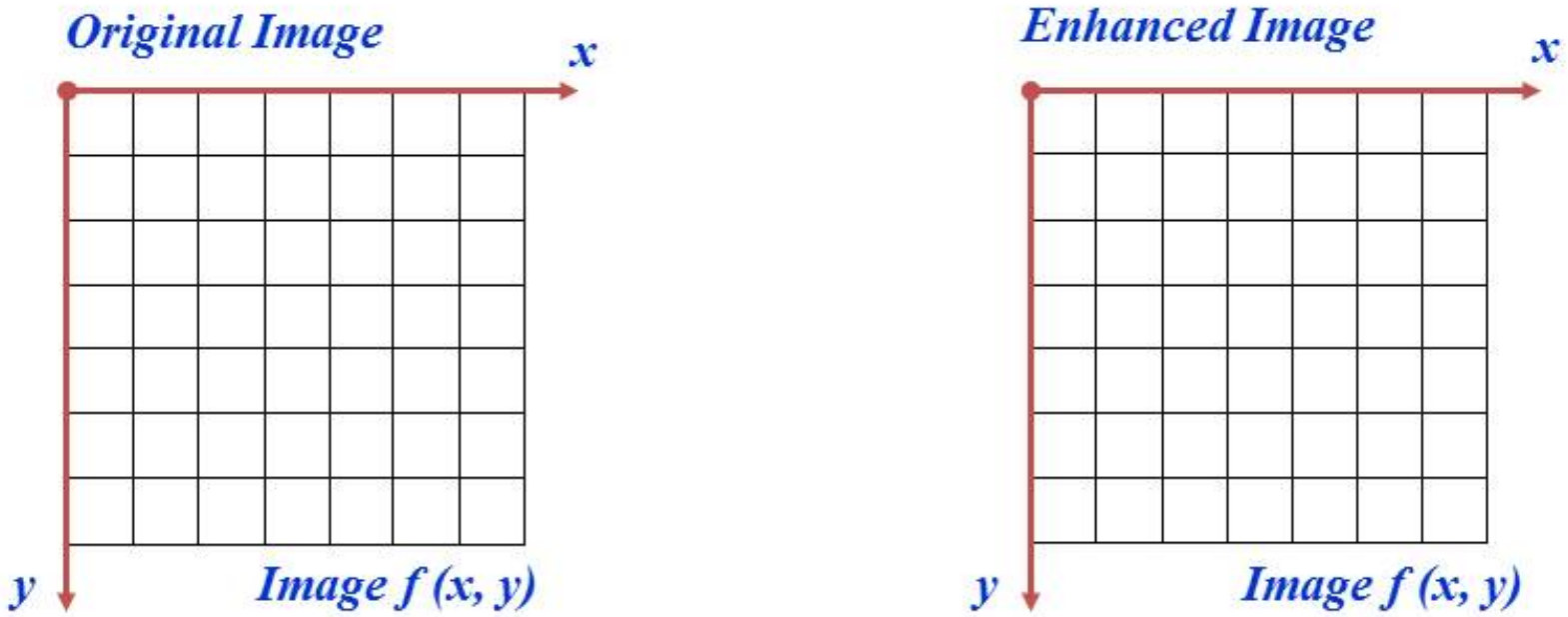
- The general form of the log transformation is
  - $s = c * log(1 + r)$
- The log transformation maps a narrow range of low input grey level values into a wider range of output values
- The inverse log transformation performs the opposite transformation

Logarithmic Transformations (cont…)

- Log functions are particularly useful when the input grey level values may have an extremely large range of values
- In the following example the Fourier transform of an image is put through a log transform to reveal more detail



Logarithmic Transformations (cont…)



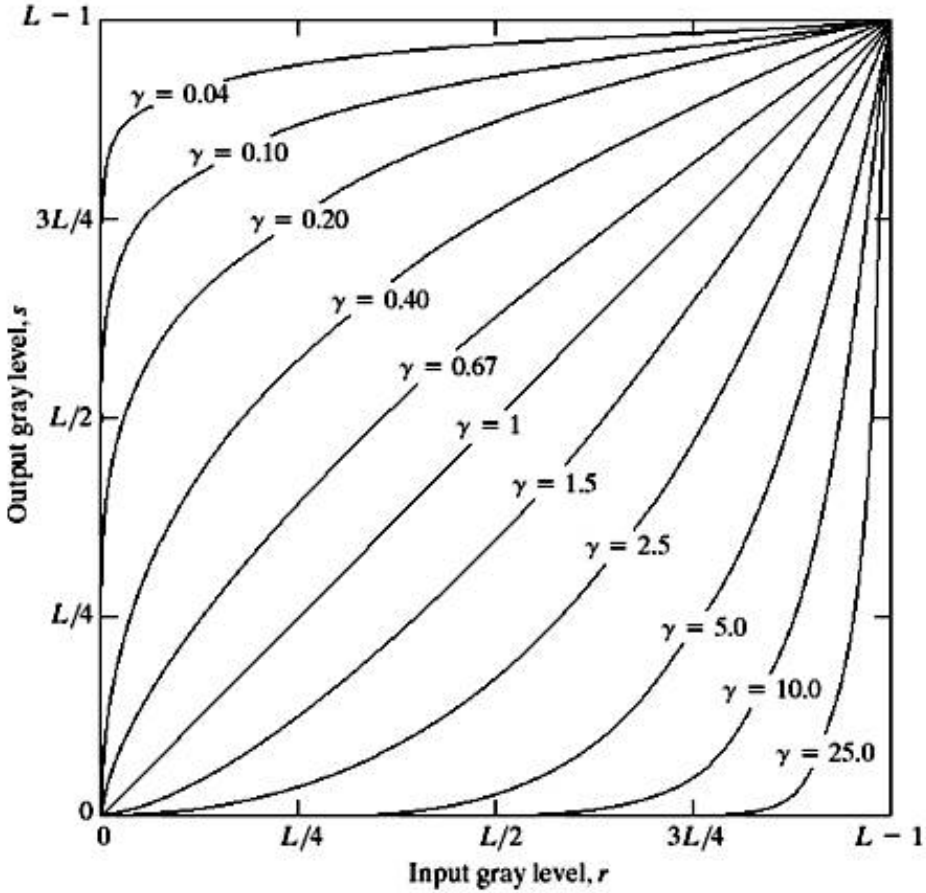
$$s = log(1 + r)$$

We usually set c to 1

Grey levels must be in the range [0.0, 1.0]

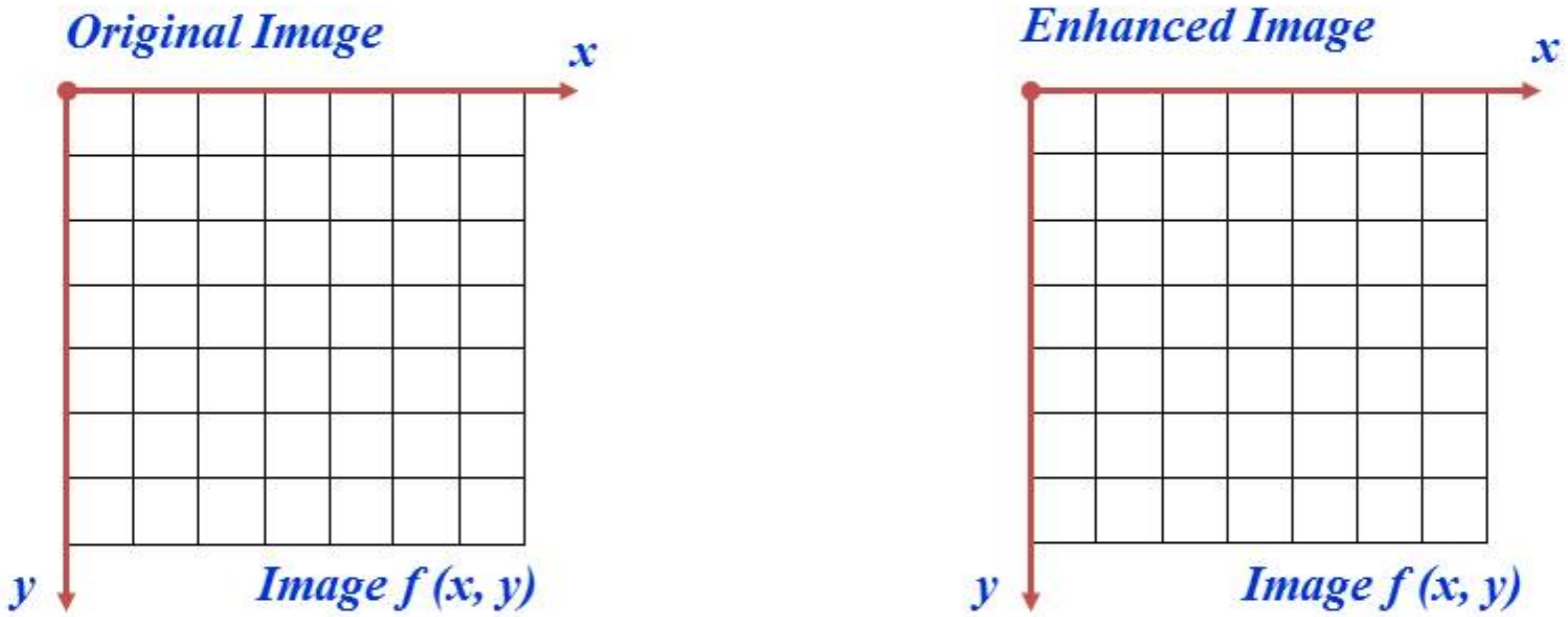
Power Law Transformations

- Power law transformations have the following form
- $s = c * r^{\gamma}$
- Map a narrow range of dark input values into a wider range of output values or vice versa
- Varying γ gives a whole family of curves



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

Power Law Transformations (cont…)



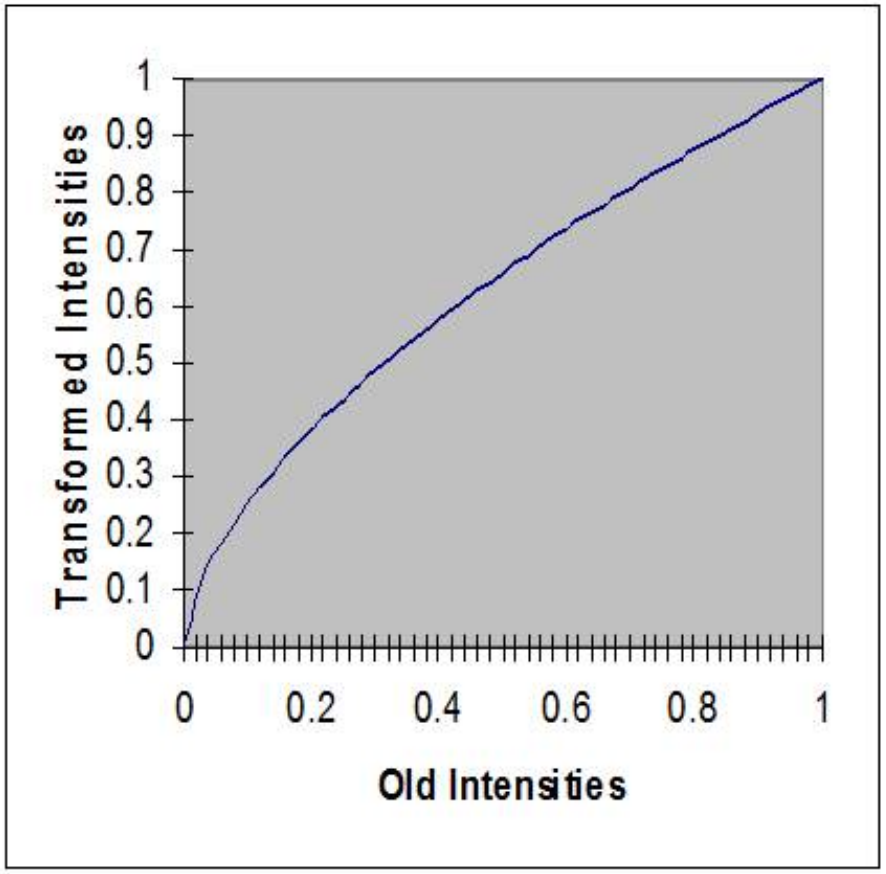
$$s = r^{\gamma}$$

- We usually set c to 1
- Grey levels must be in the range [0.0, 1.0]

Power Law Example

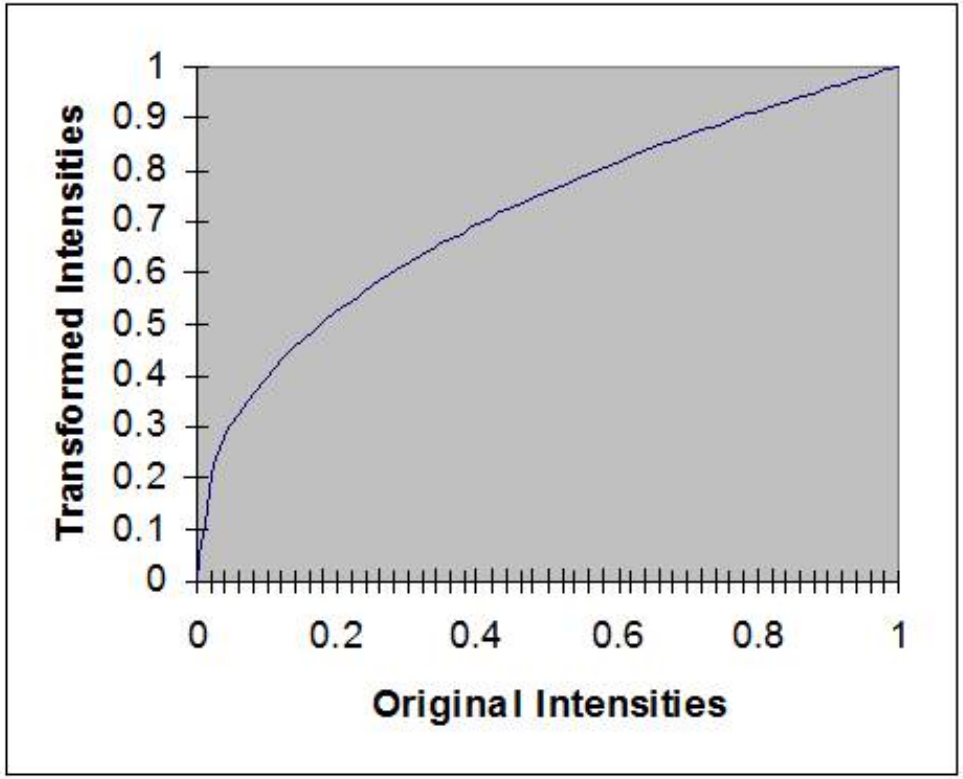
Power Law Example (cont…)

$$\gamma = 0.6$$



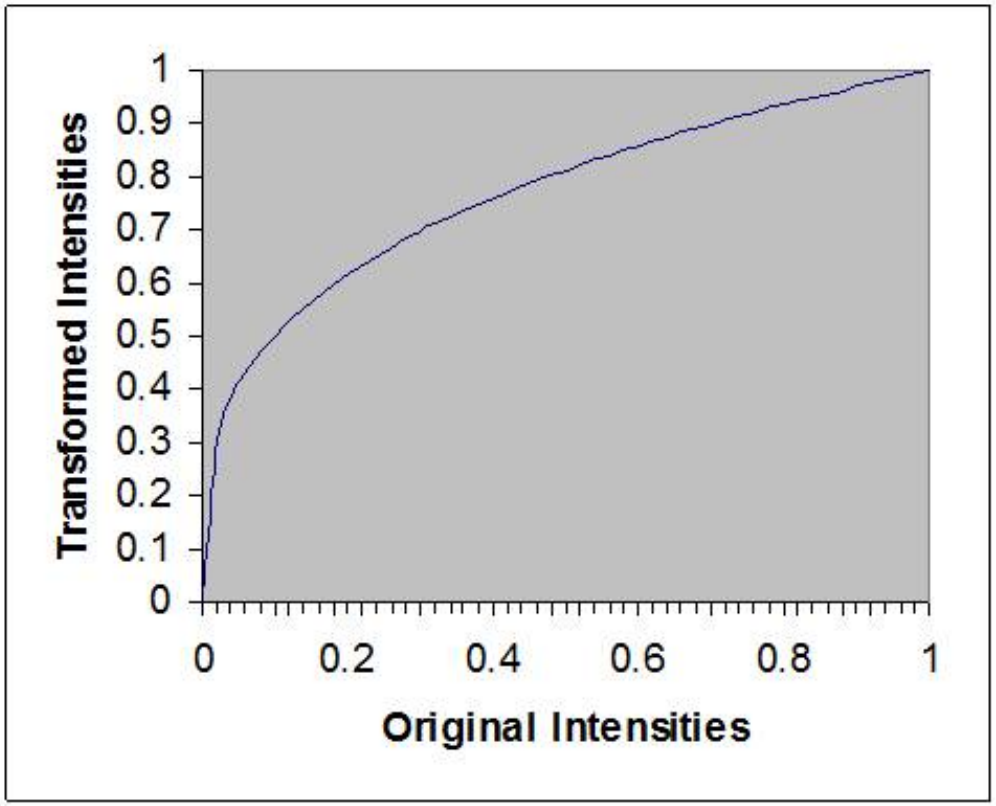
Power Law Example (cont…)

$$\gamma = 0.4$$



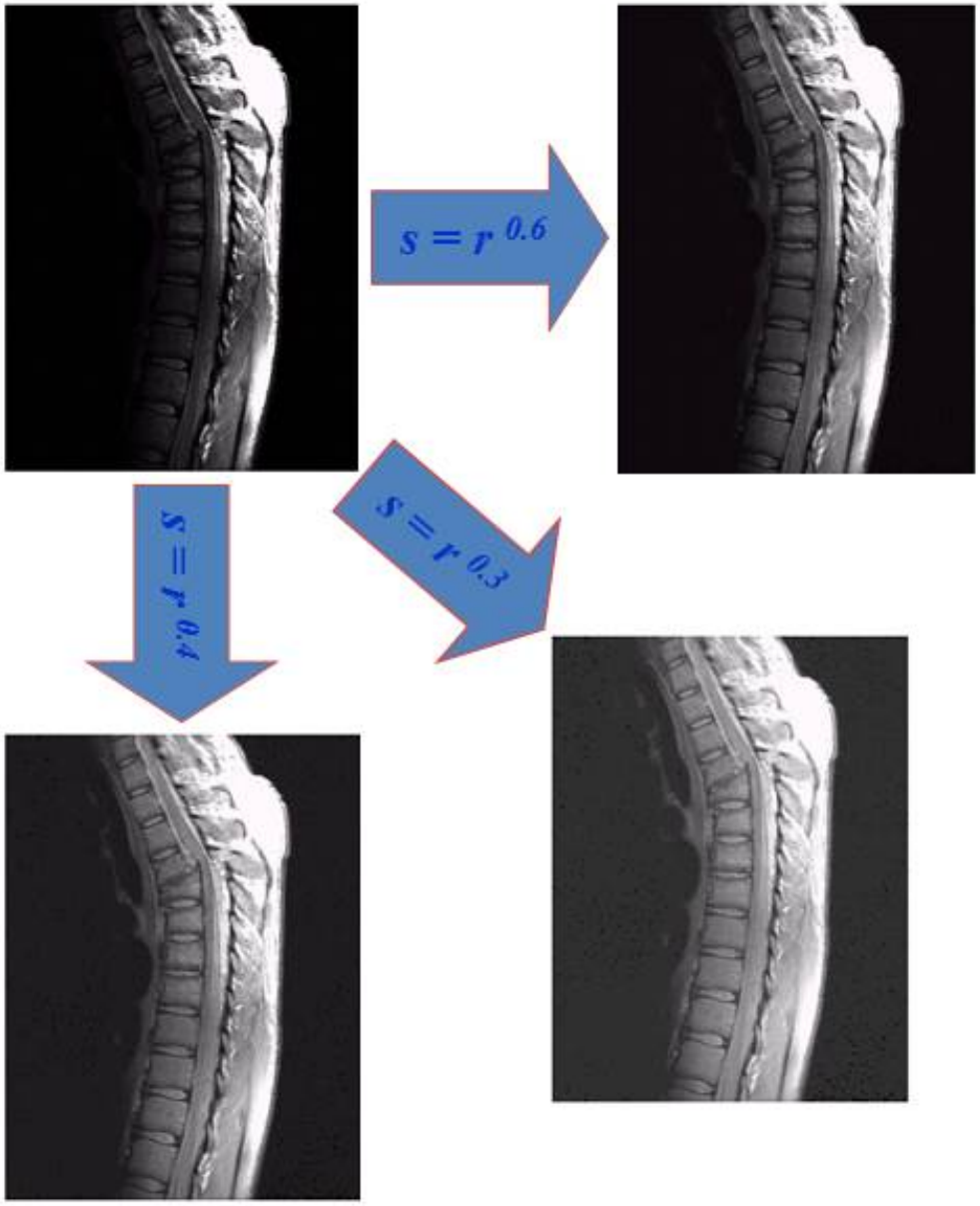
Power Law Example (cont…)

$\gamma = 0.3$



Power Law Example (cont…)

- The images to the right show a magnetic resonance (MR) image of a fractured human spine
- Different curves highlight different detail

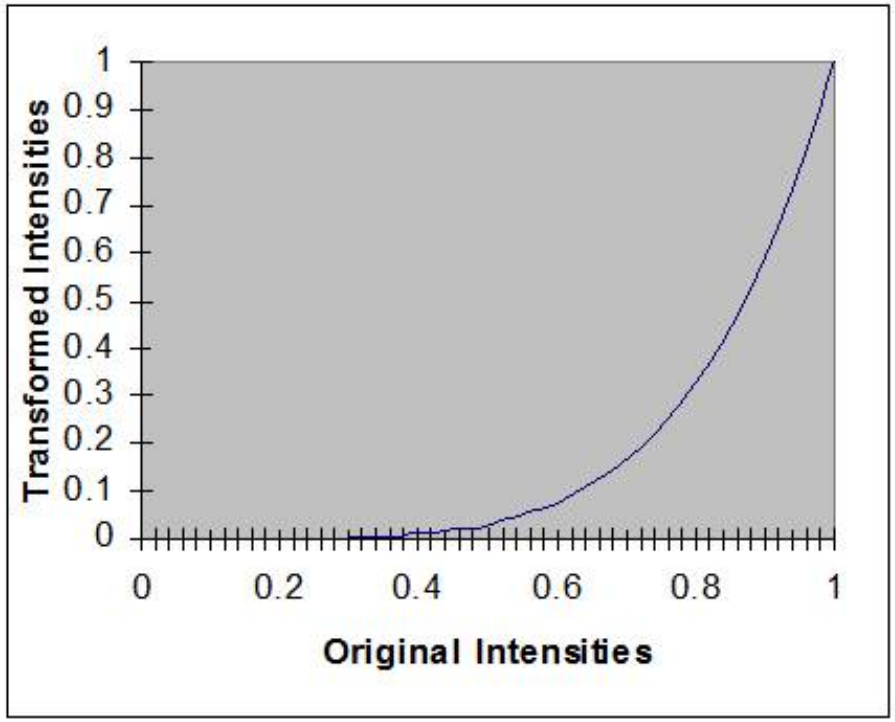


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Power Law Example

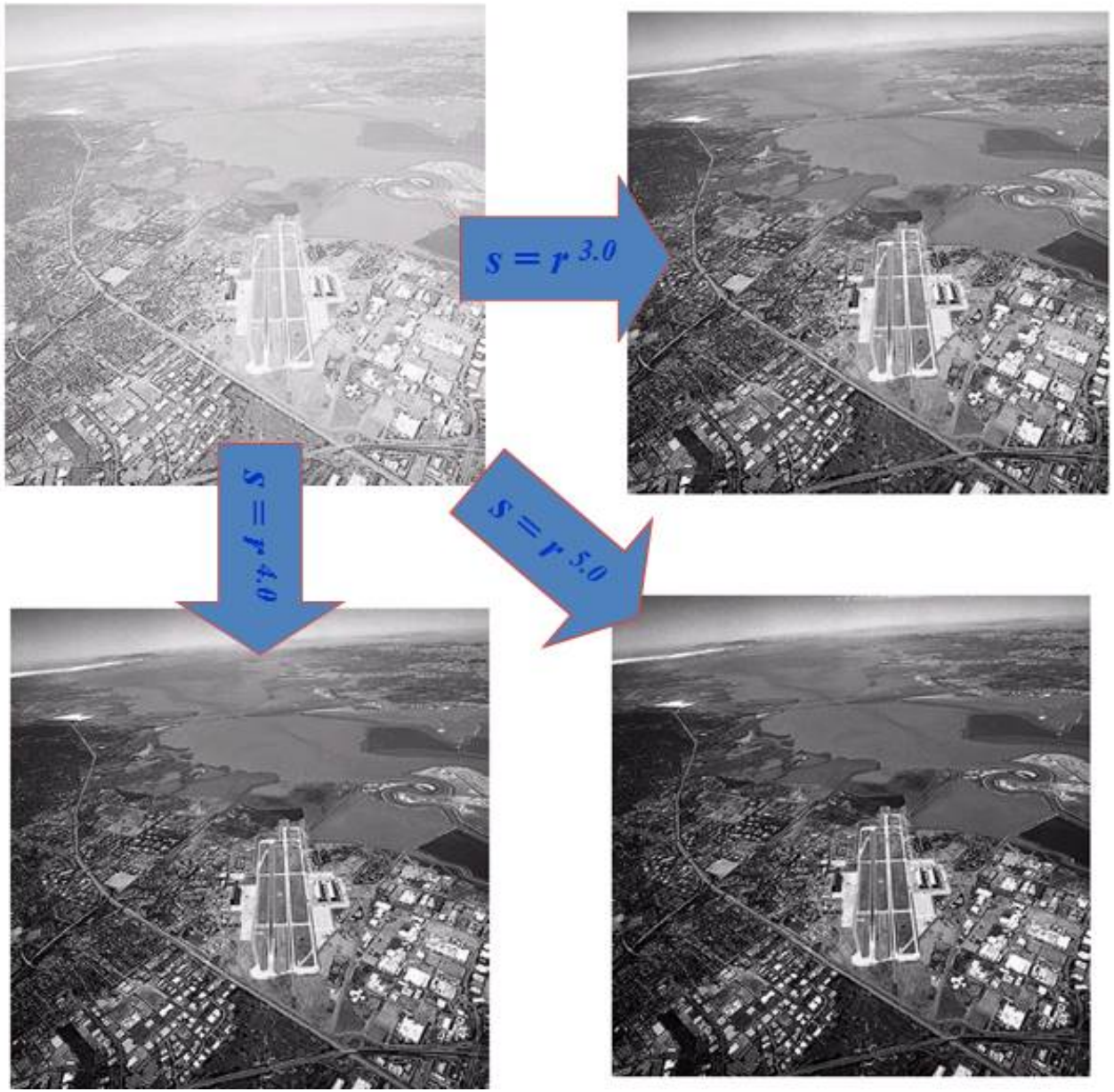
Power Law Example (cont…)

$$\gamma = 5.0$$



Power Law Transformations (cont…)

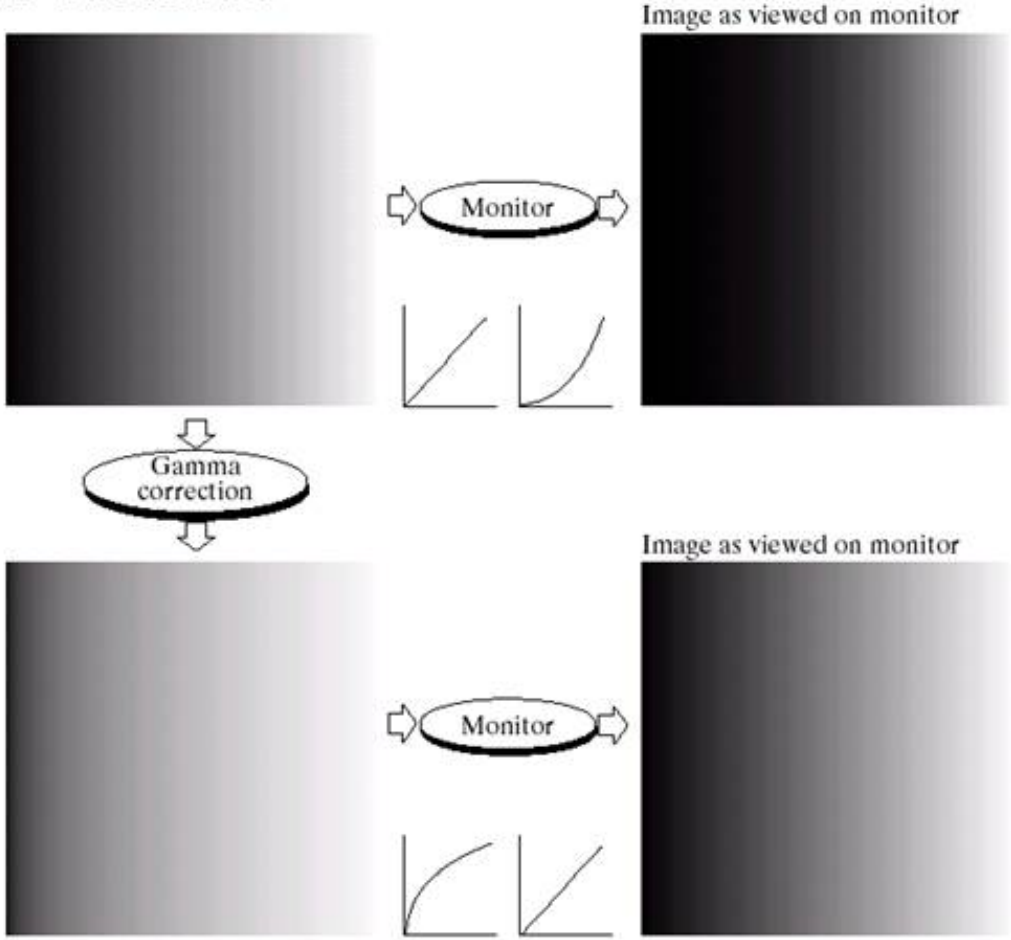
- An aerial photo of a runway is shown
- This time power law transforms are used to darken the image
- Different curves highlight different detail



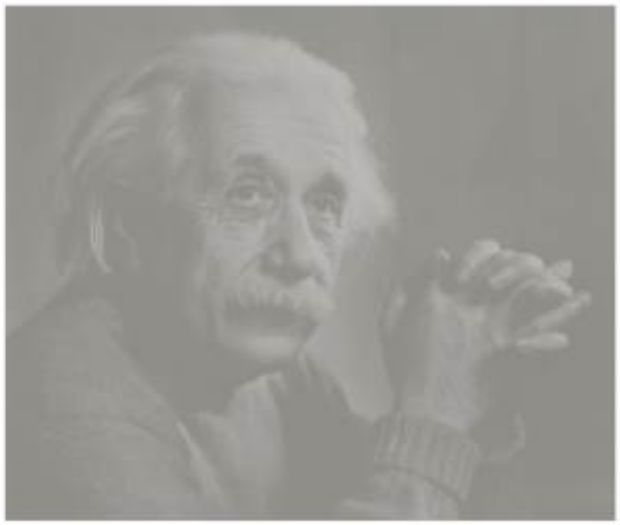
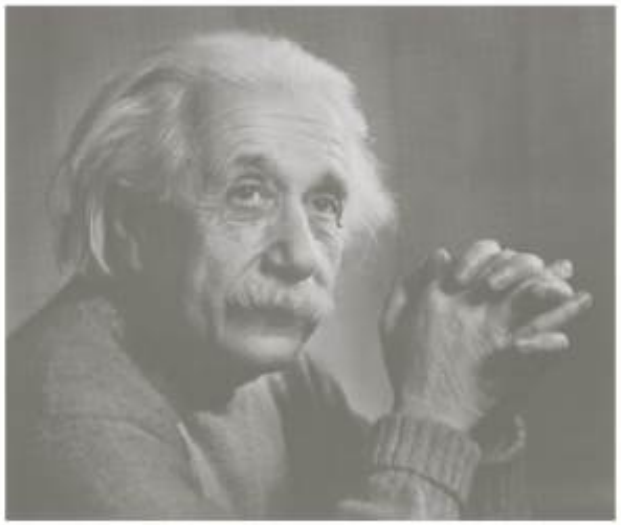
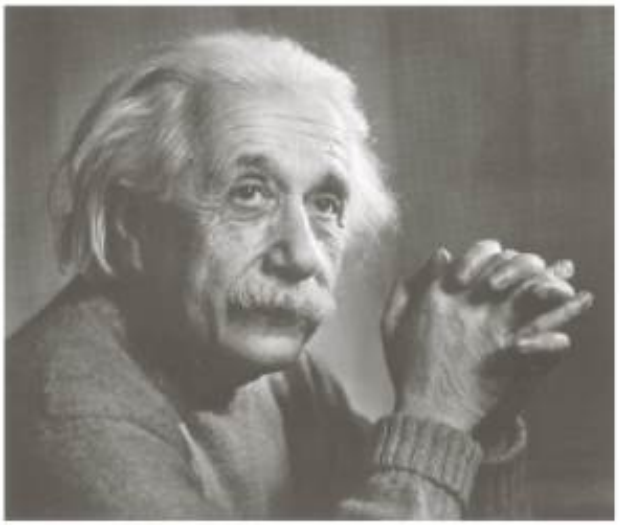
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Gamma Correction

- Many of you might be familiar with gamma correction of computer monitors
- Problem is that display devices do not respond linearly to different intensities
- Can be corrected using a log transform



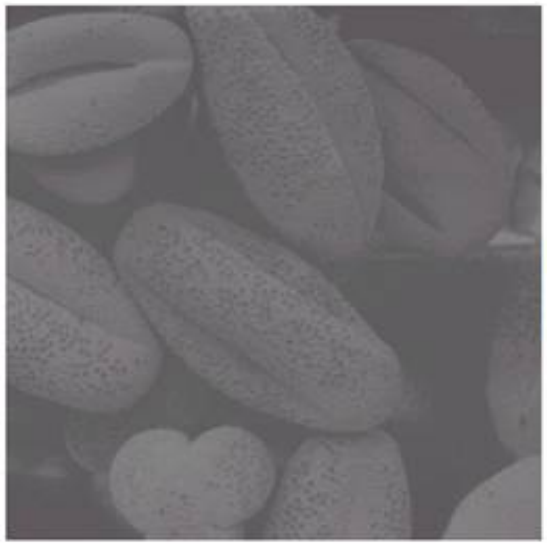
More Contrast Issues

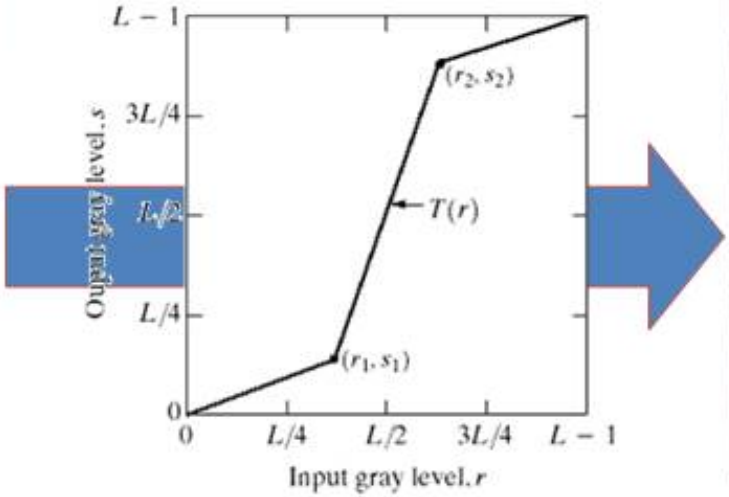




Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Piecewise Linear Transformation Functions

- Rather than using a well defined mathematical function we can use arbitrary user-defined transforms
- The images below show a contrast stretching linear transform to add contrast to a poor quality image

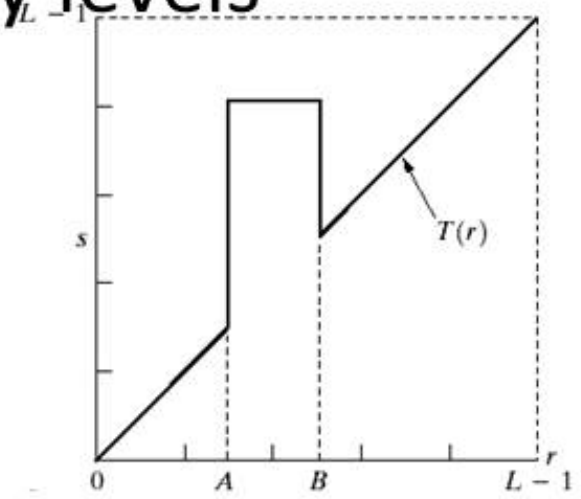


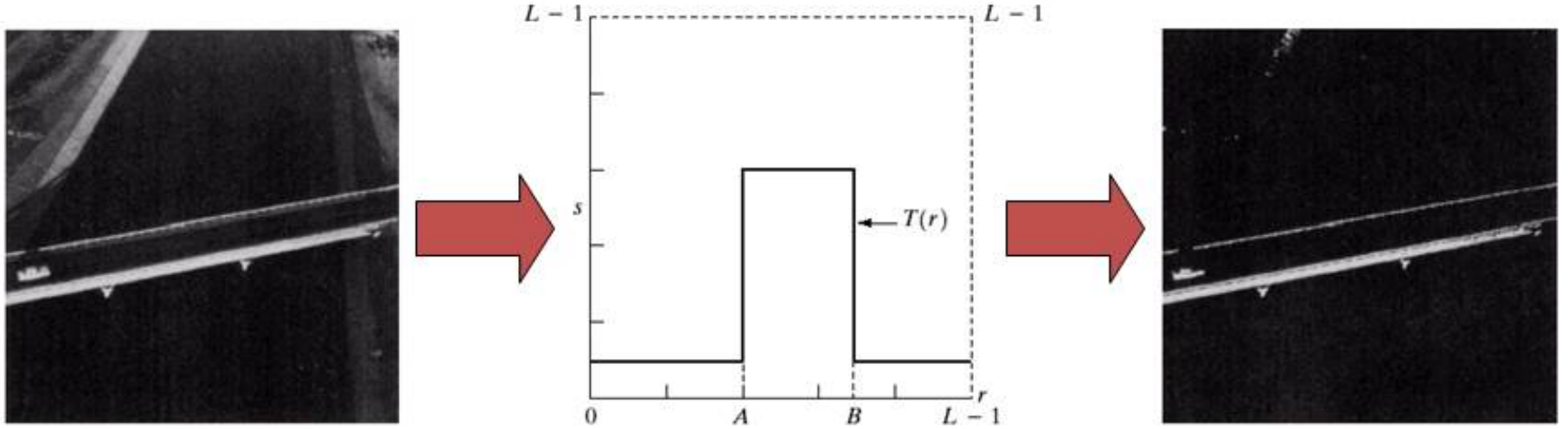


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Gray Level Slicing

- Highlights a specific range of grey levels
  - Similar to thresholding
  - Other levels can be suppressed or maintained
  - Useful for highlighting features in an image

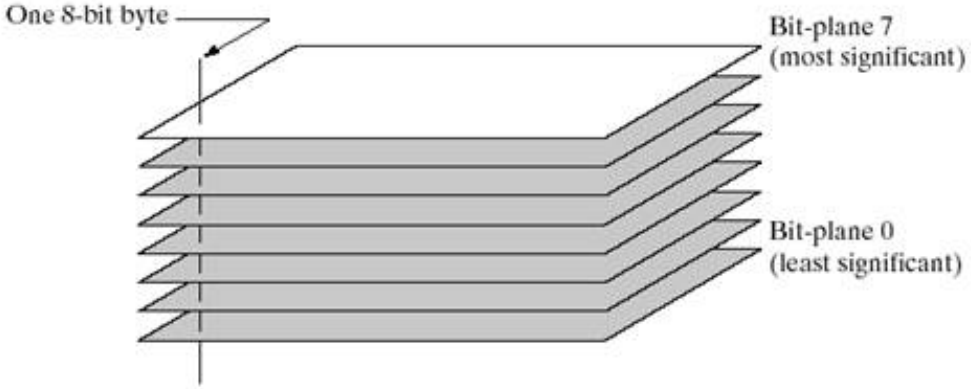


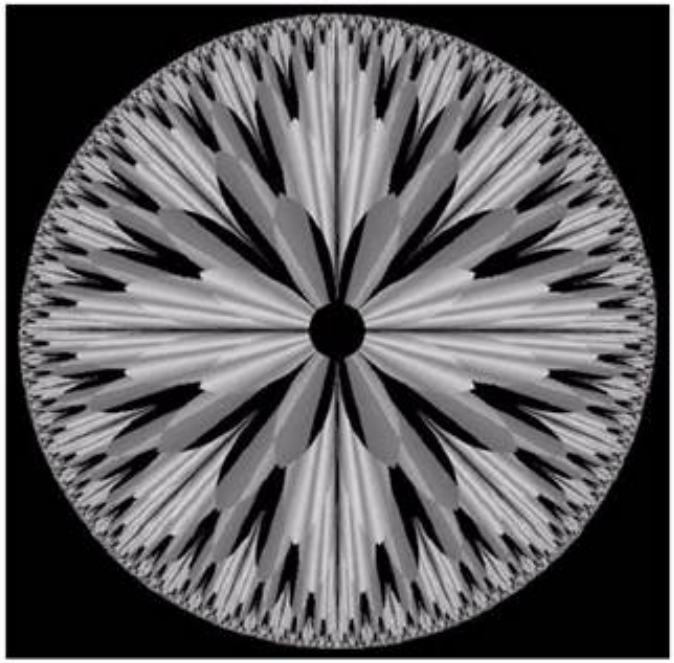


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

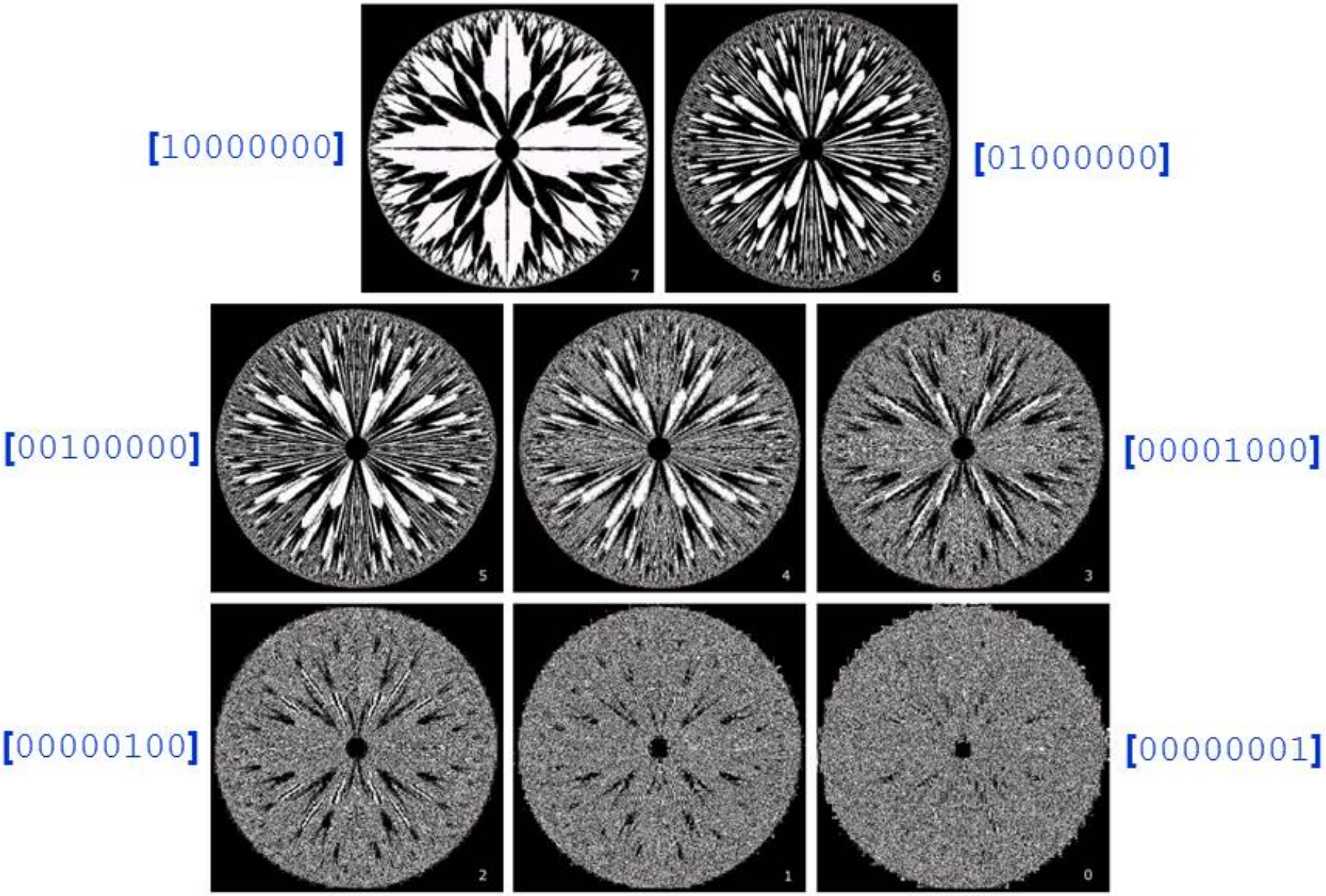
Bit Plane Slicing

- Often by isolating particular bits of the pixel values in an image we can highlight interesting aspects of that image
  - Higher-order bits usually contain most of the significant visual information
  - Lower-order bits contain subtle details



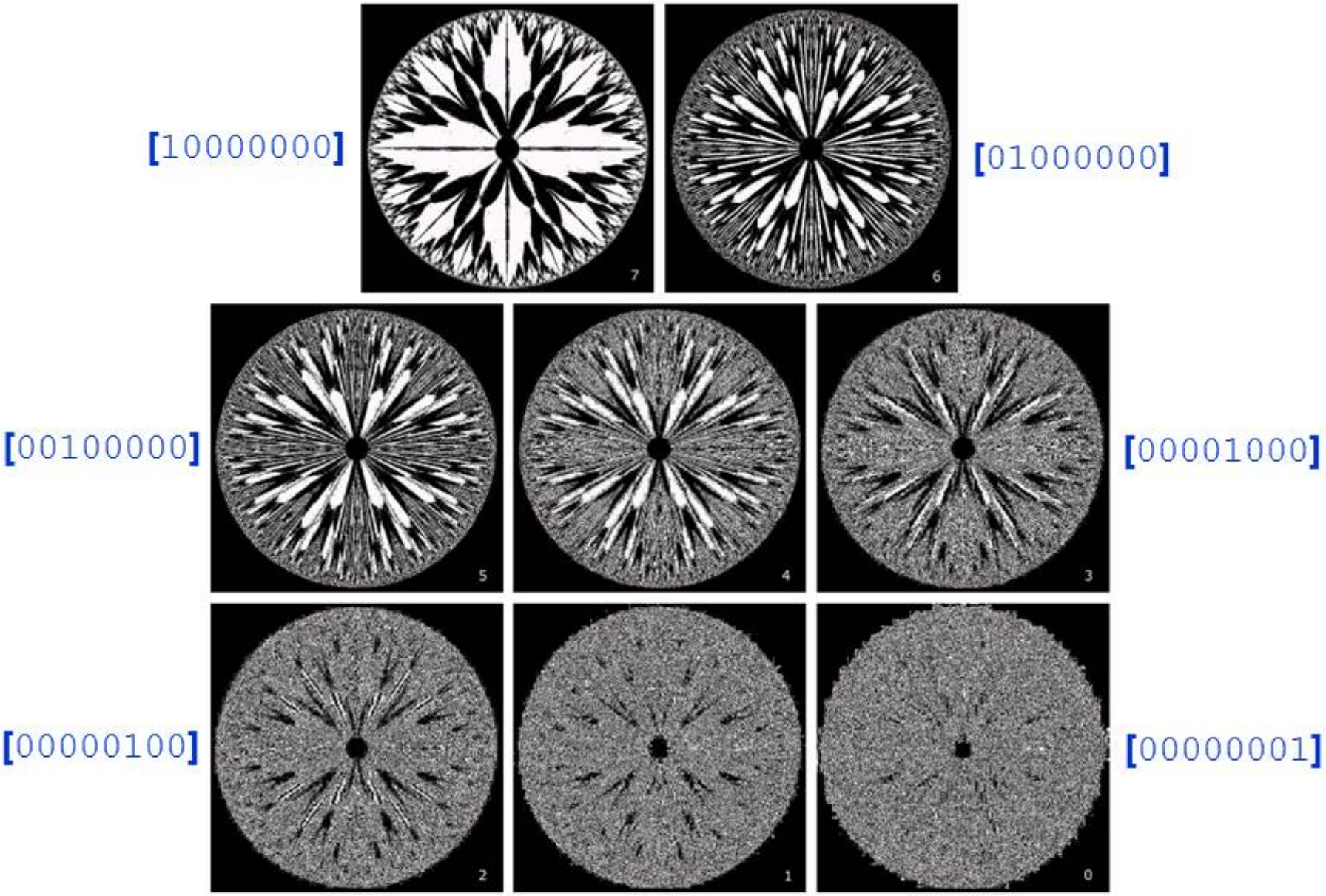


Bit Plane Slicing (cont…)



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

Bit Plane Slicing (cont…)



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

Bit Plane Slicing (cont…)

a b c
d e f
g h i

FIGURE 3.14 (a) An 8-bit gray-scale image of size 500 × 1192 pixels. (b) through (i) Bit planes 1 through 8, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image.

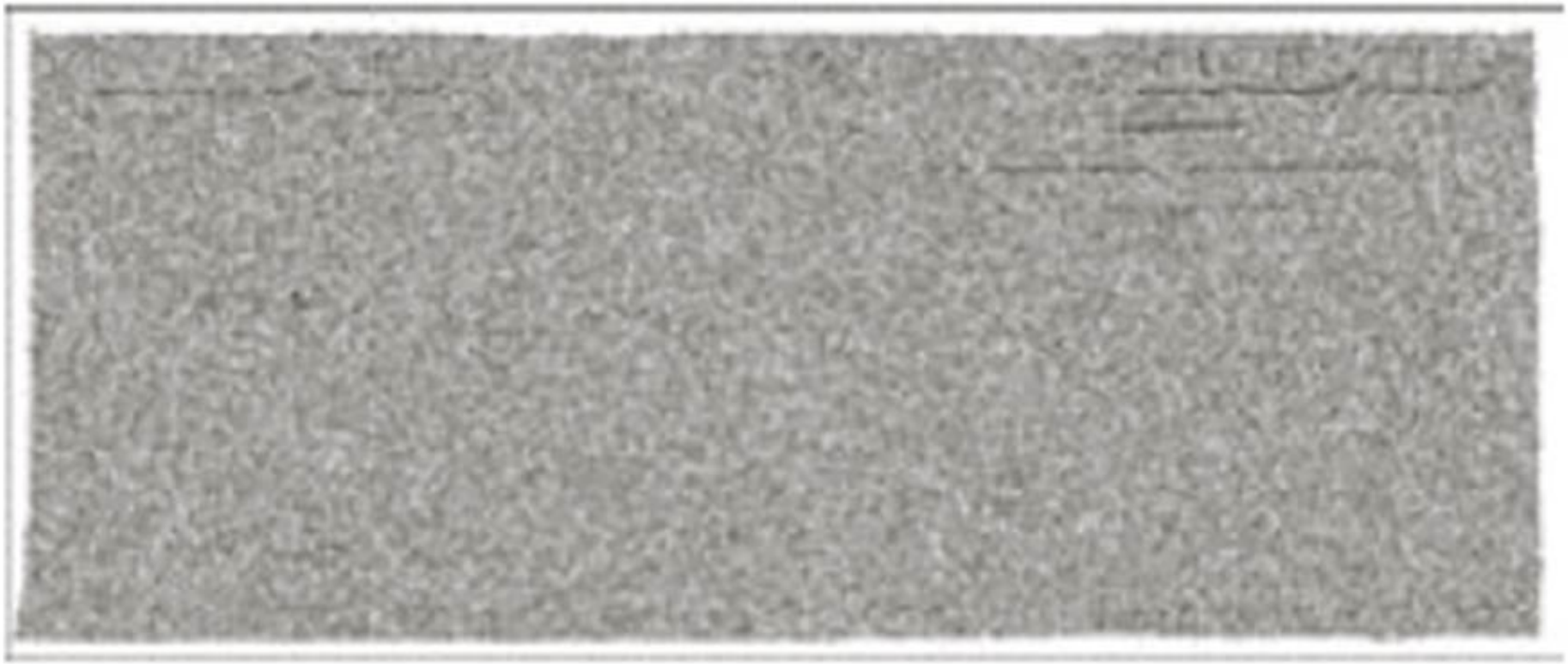
Bit Plane Slicing (cont…)

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

Bit Plane Slicing (cont…)

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

Bit Plane Slicing (cont…)

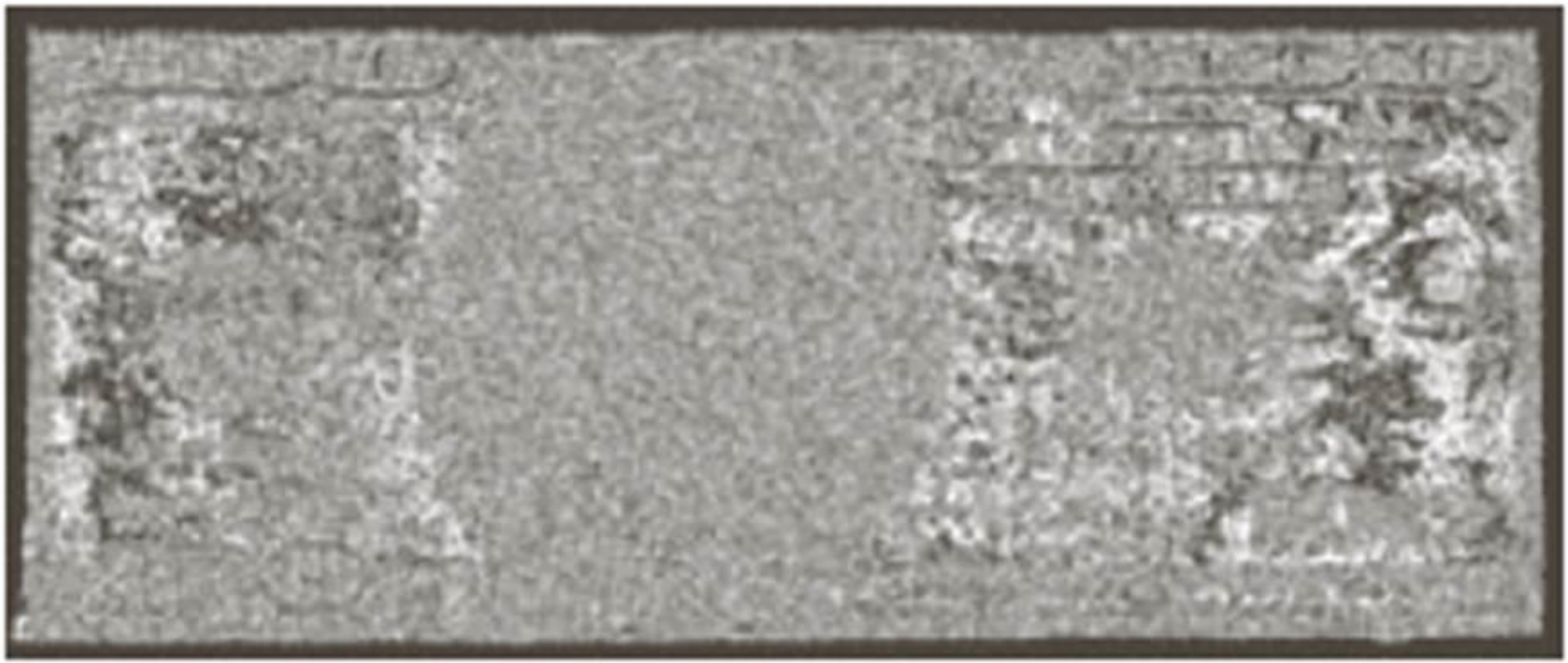


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)

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

Bit Plane Slicing (cont…)



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

Bit Plane Slicing (cont…)

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

Bit Plane Slicing (cont…)

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

Bit Plane Slicing (cont…)

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

Bit Plane Slicing (cont…)

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

Bit Plane Slicing (cont…)

Reconstructed image using only bit planes 8 and 7

Reconstructed image using only bit planes 8, 7 and 6

Reconstructed image using only bit planes 7, 6 and 5

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

A Note About Grey Levels

So far when we have spoken about image grey level values we have said they are in the range [0, 255]

- Where 0 is black and 255 is white

There is no reason why we have to use this range

- The range [0,255] stems from display technologes

For many of the image processing operations in this lecture grey levels are assumed to be given in the range [0.0, 1.0]

What Is Image Enhancement?

Image enhancement is the process of making images more useful

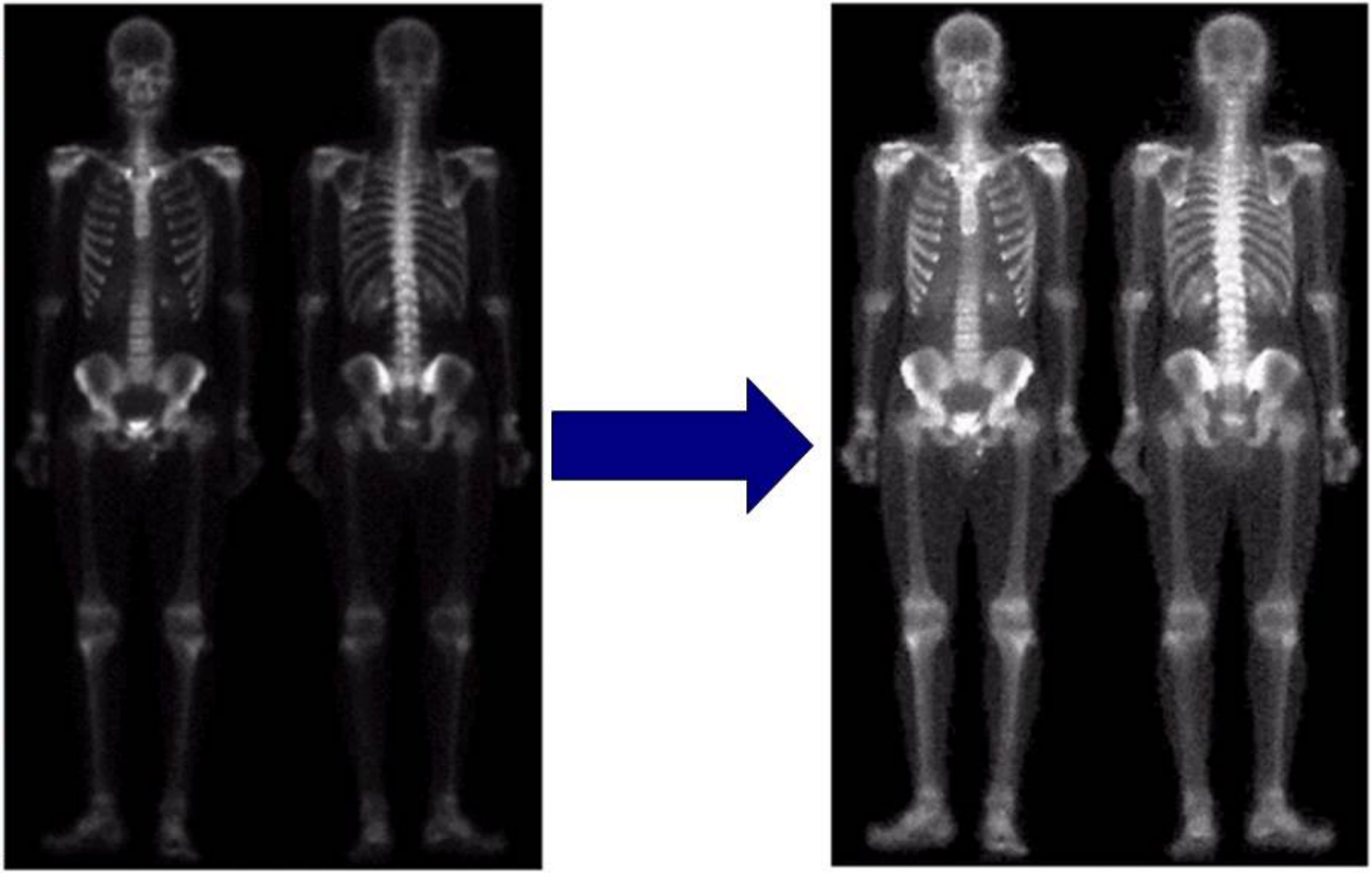
The reasons for doing this include:

- Highlighting interesting detail in images
- Removing noise from images
- Making images more visually appealing

Image Enhancement Examples

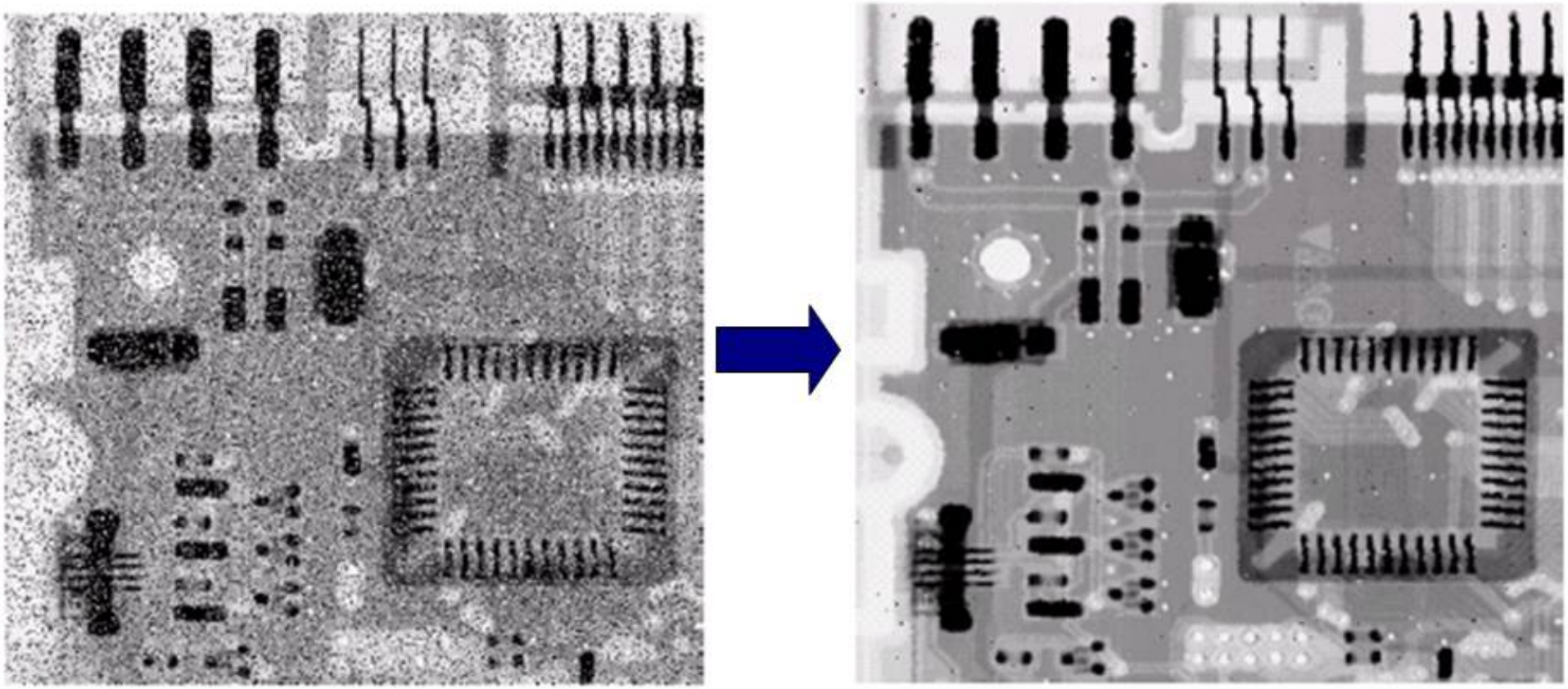
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Image Enhancement Examples (cont…)



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

Image Enhancement Examples (cont…)



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

Image Enhancement Examples (cont…)

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

Spatial & Frequency Domains

There are two broad categories of image enhancement techniques

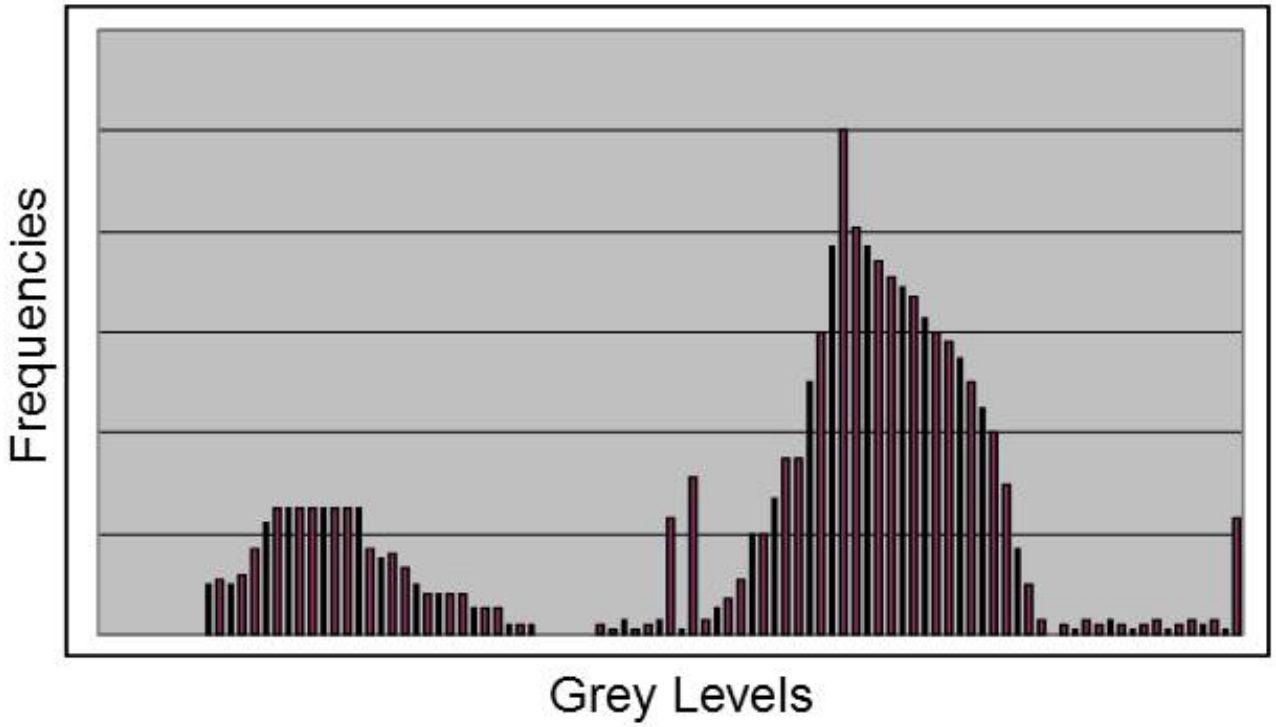
- Spatial domain techniques
  - Direct manipulation of image pixels
- Frequency domain techniques
  - Manipulation of Fourier transform or wavelet transform of an image

For the moment we will concentrate on techniques that operate in the spatial domain

Image Histograms

The histogram of an image shows us the distribution of grey levels in the image

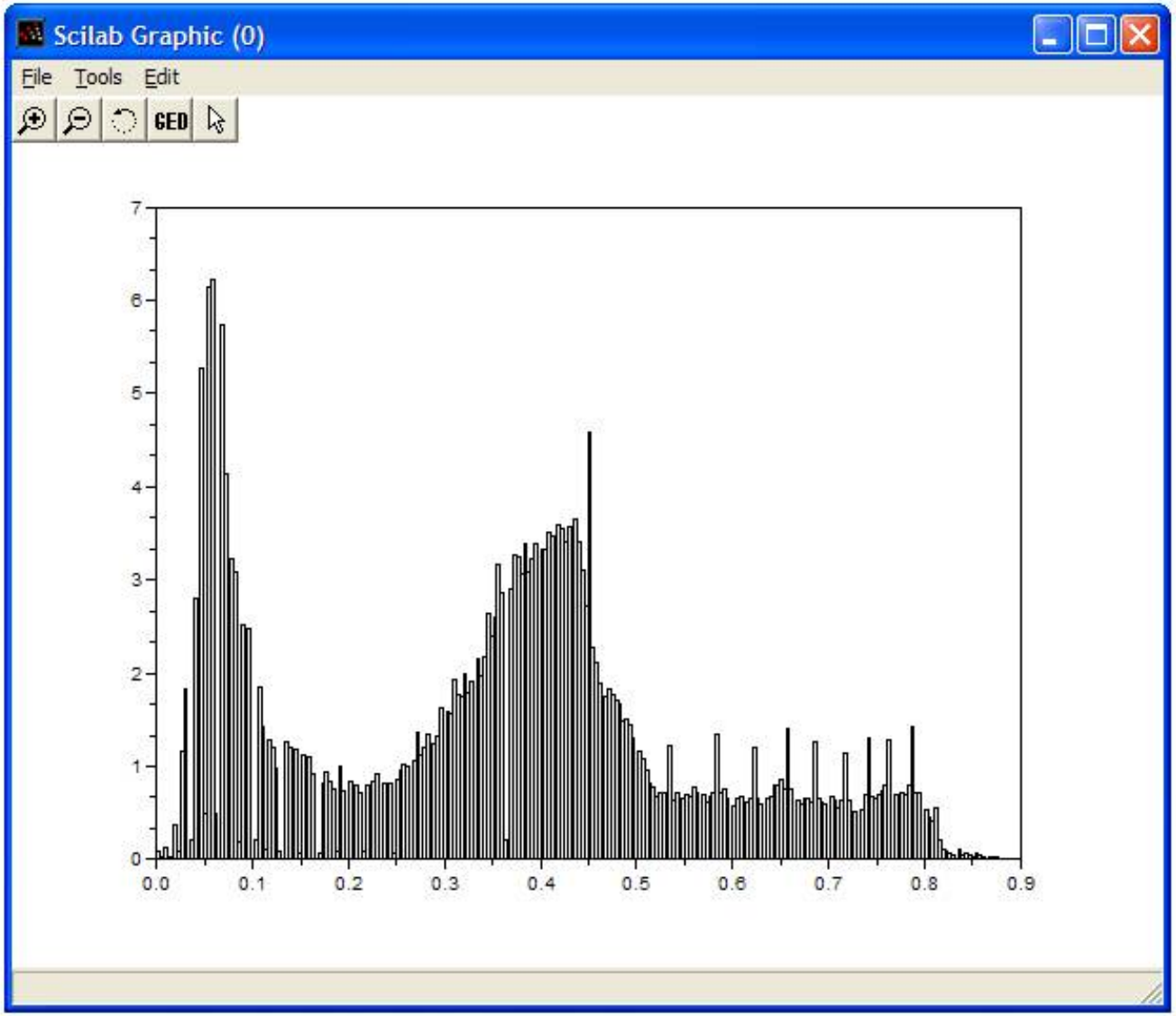
Massively useful in image processing, especially in segmentation



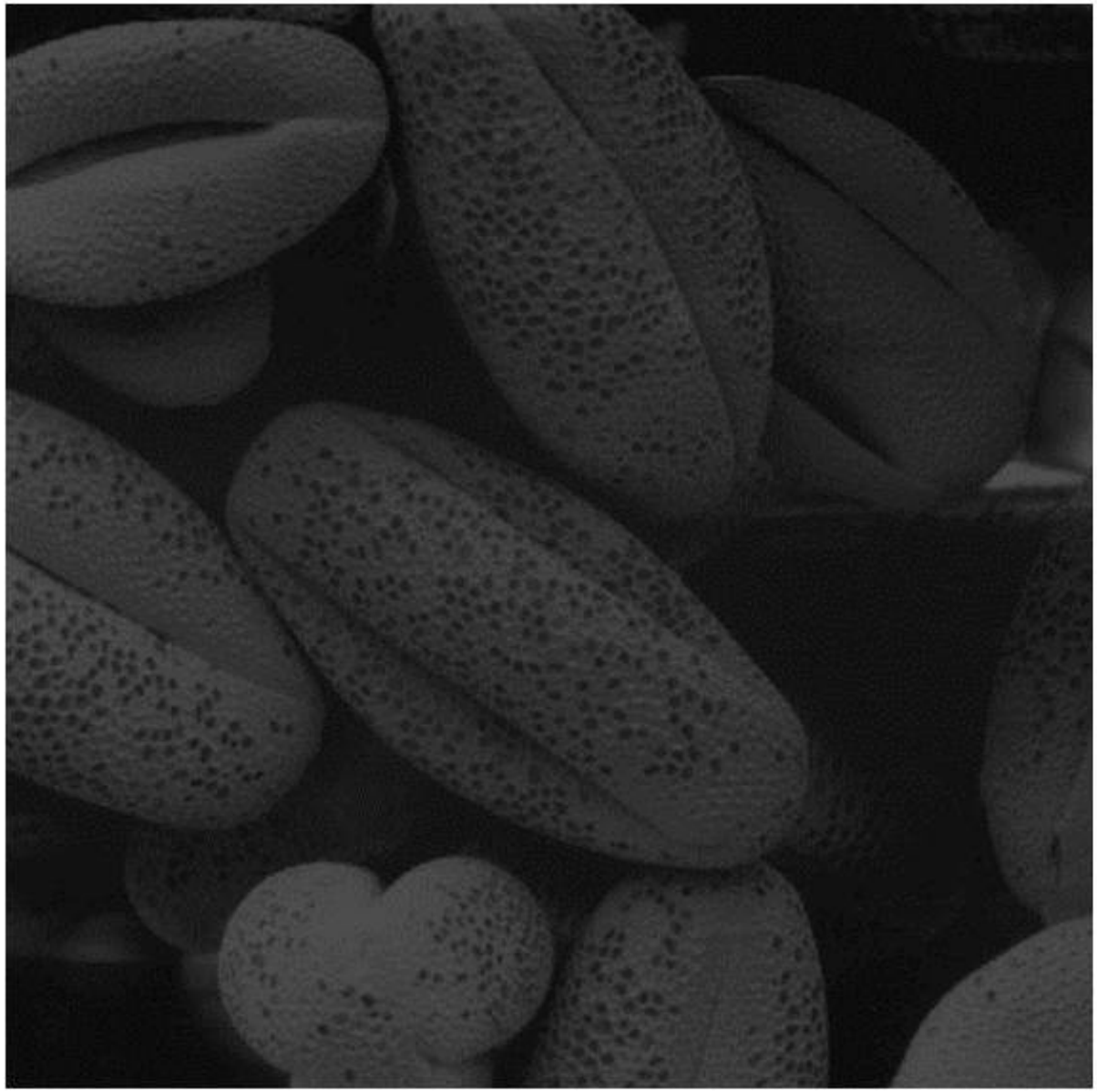
Histogram Examples

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

Histogram Examples (cont…)



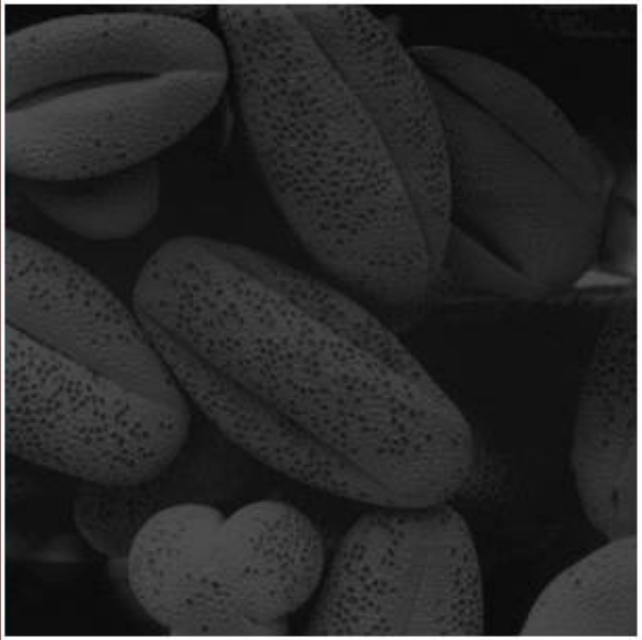
Histogram Examples (cont…)

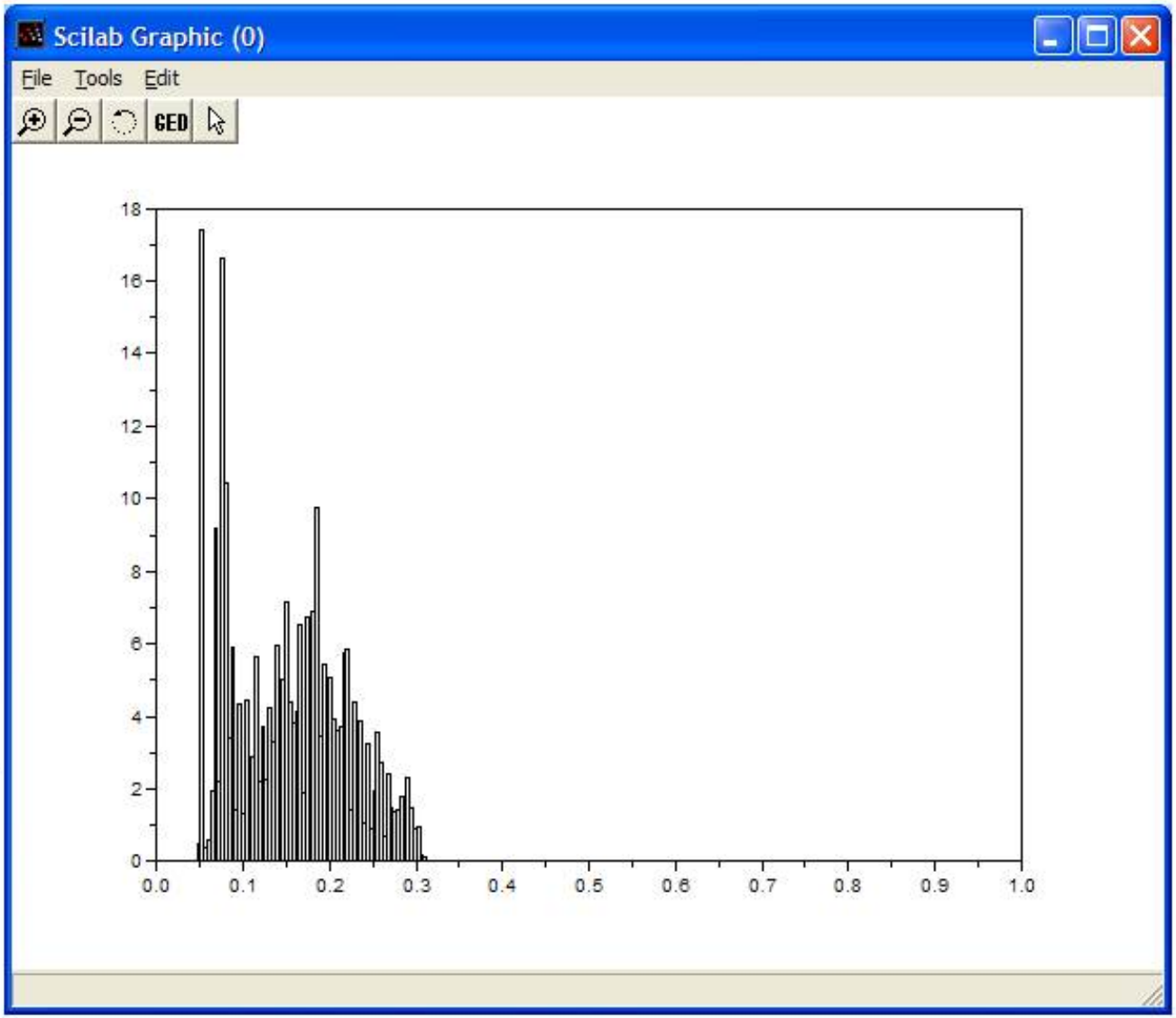


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Histogram Examples (cont…)

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

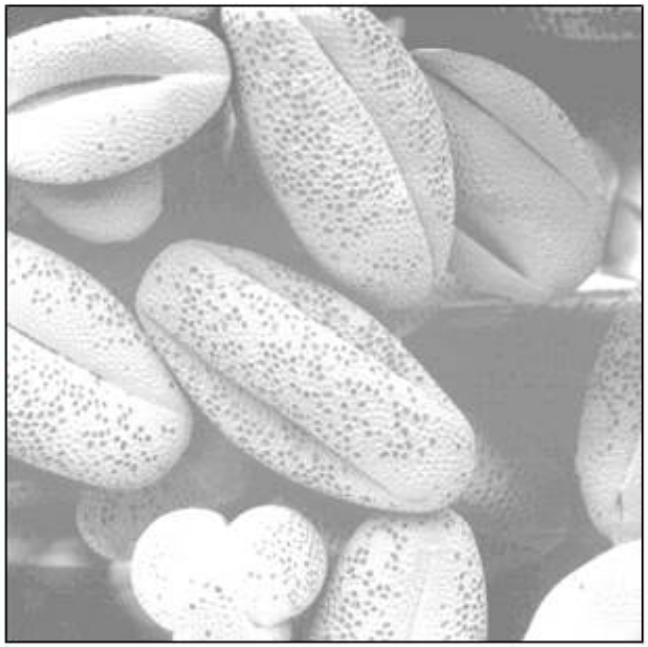


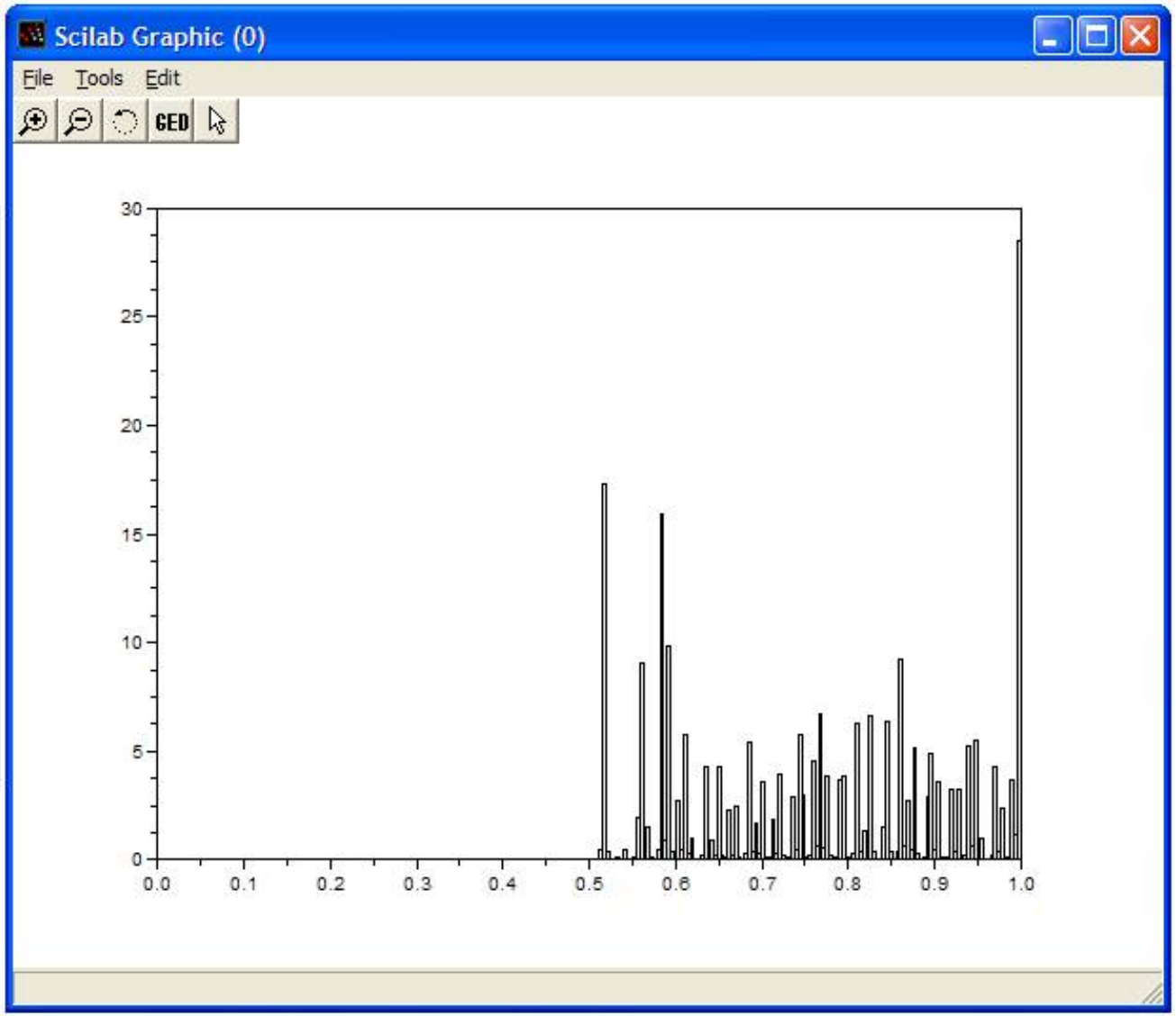


Histogram Examples (cont…)

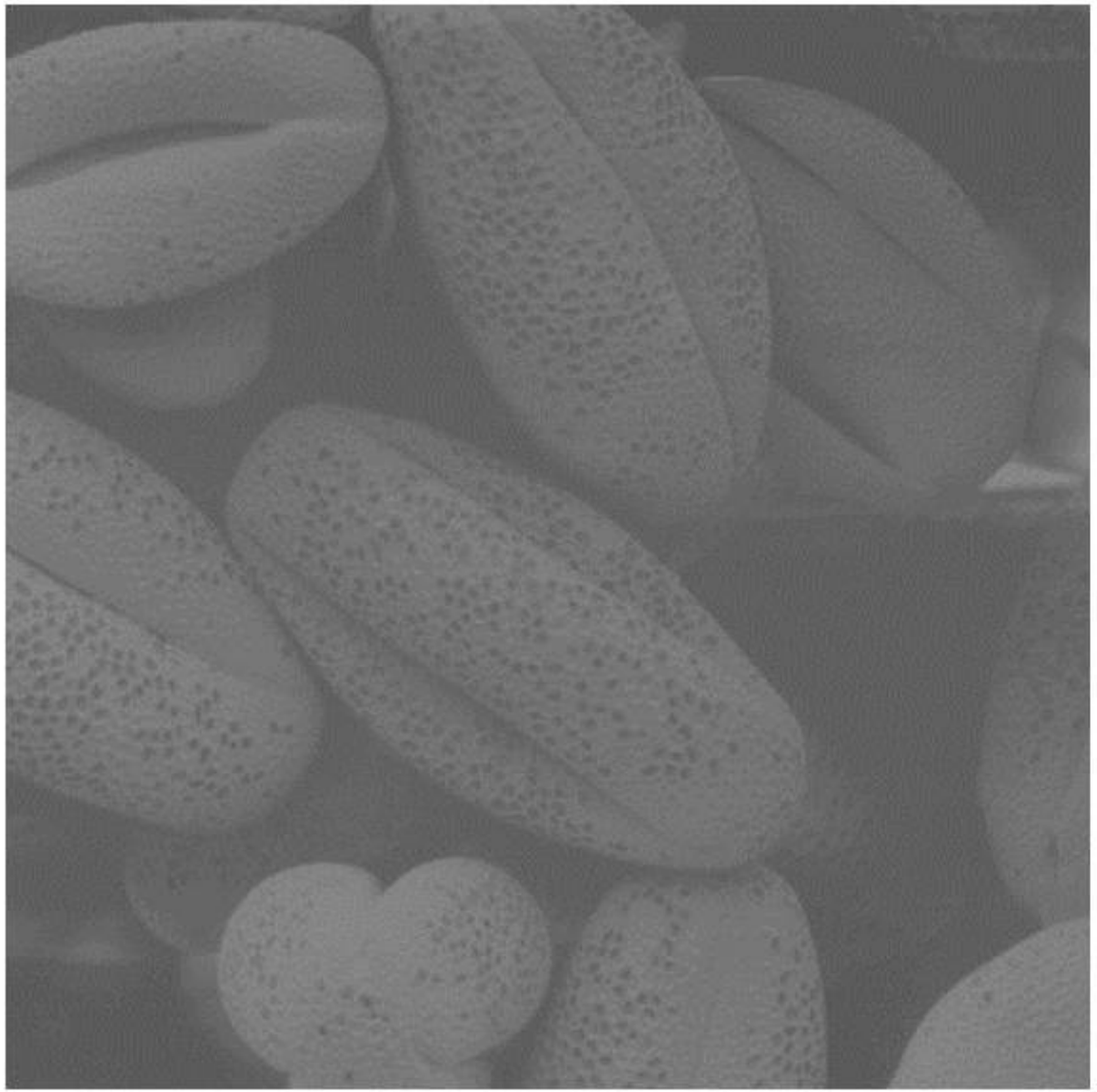
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Histogram Examples (cont…)

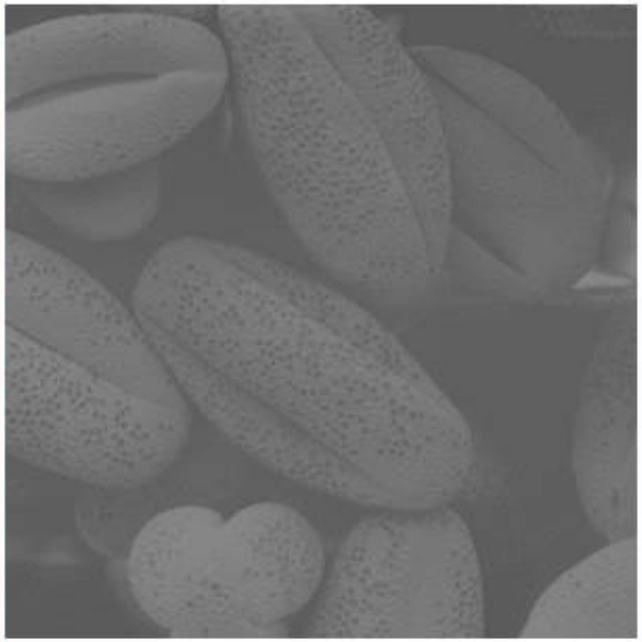


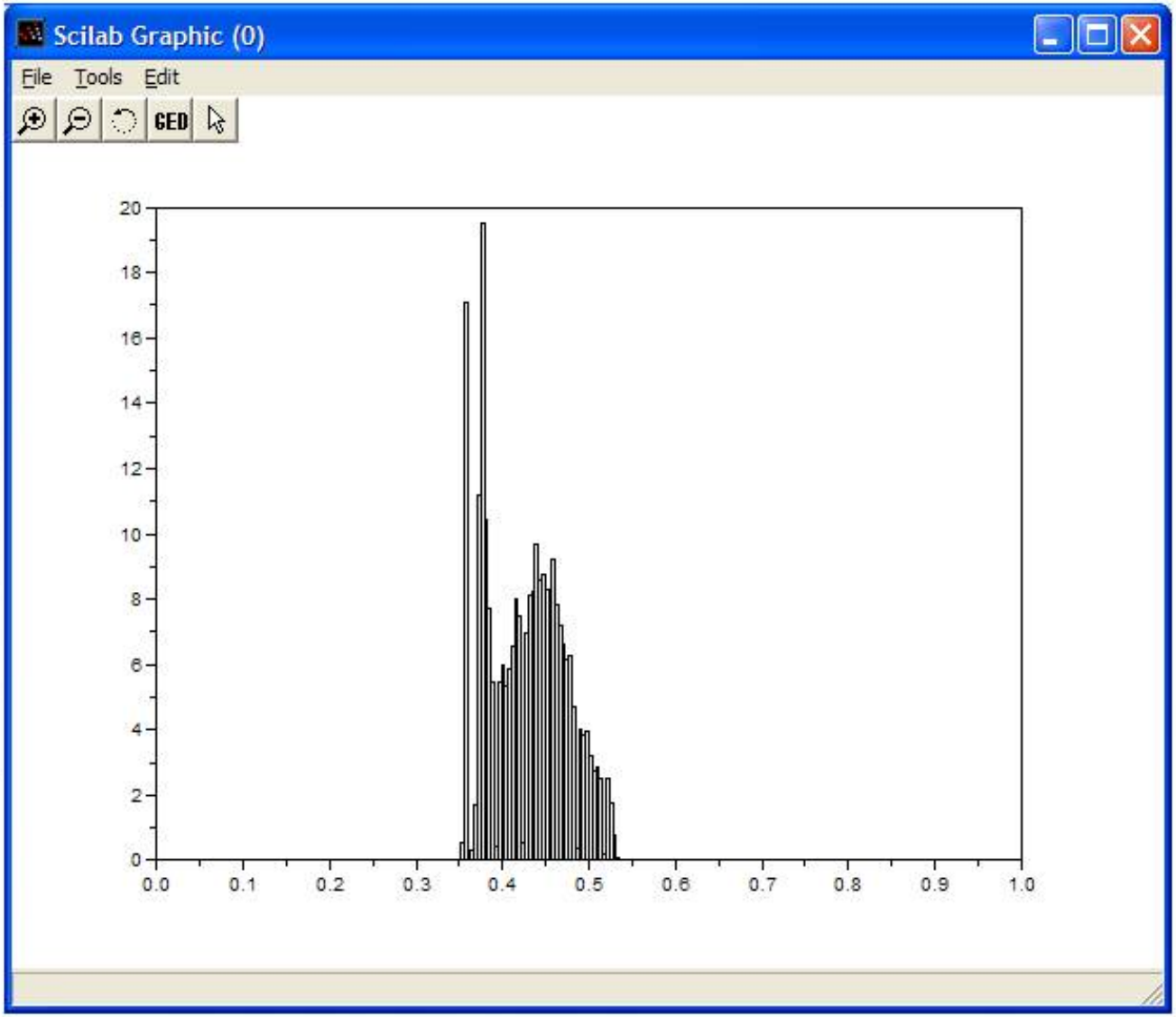


Histogram Examples (cont…)



Histogram Examples (cont…)



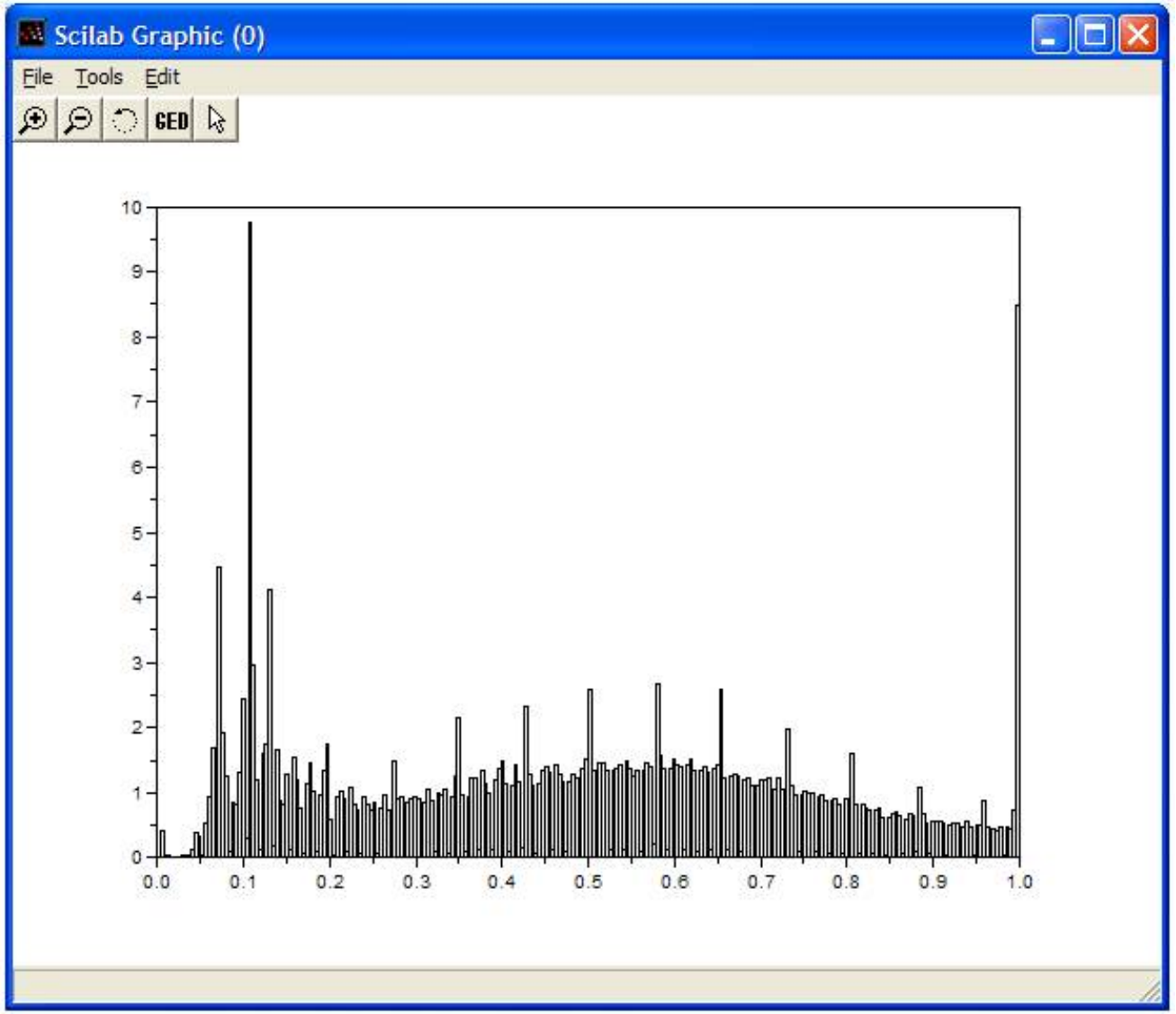


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Histogram Examples (cont…)

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

Histogram Examples (cont…)



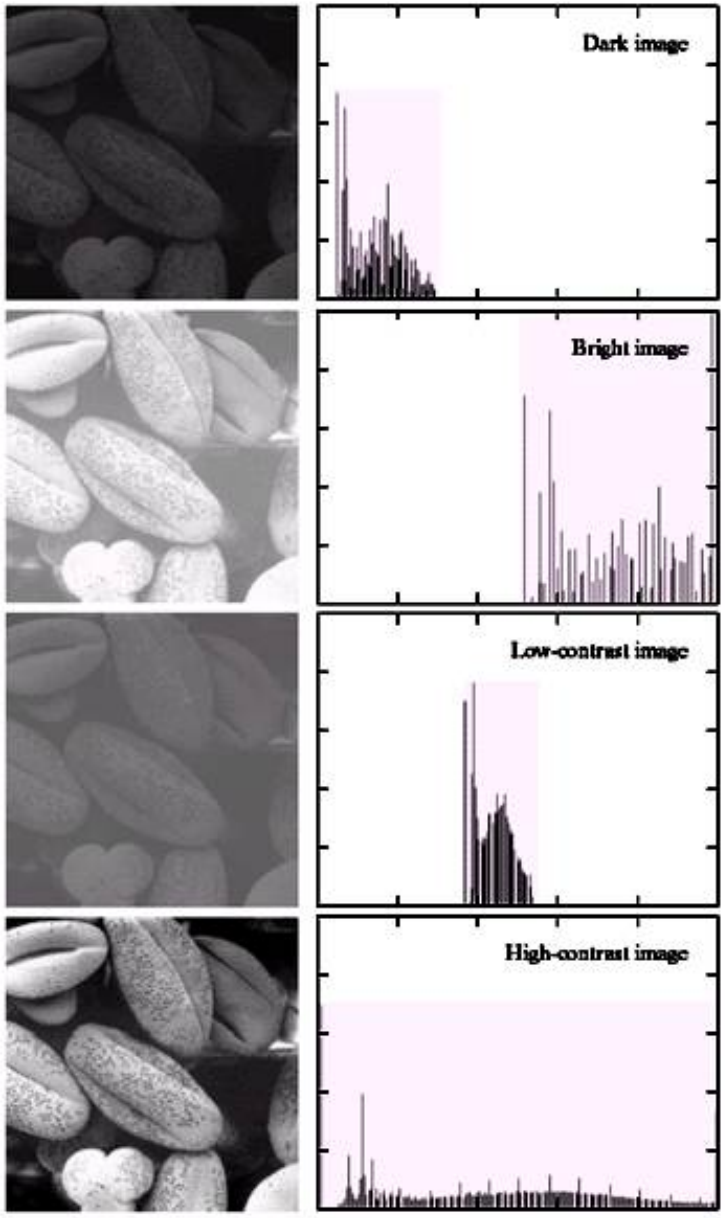
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Histogram Examples (cont…)

A selection of images and their histograms

Notice the relationships between the images and their histograms

Note that the high contrast image has the most evenly spaced histogram



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

Contrast Stretching

We can fix images that have poor contrast by applying a pretty simple contrast specification

The interesting part is how do we decide on this transformation function?

Histogram Equalisation

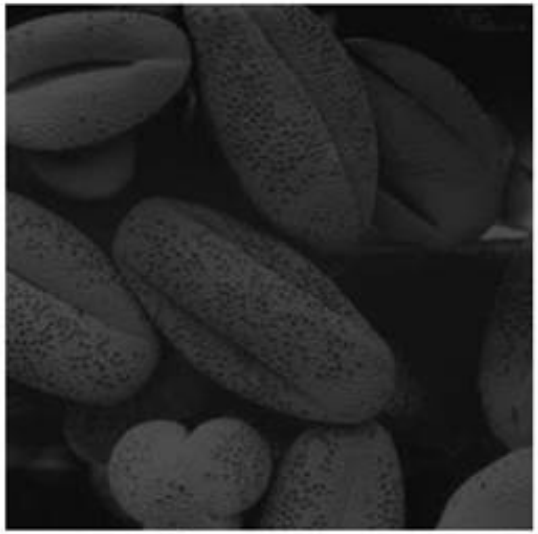
Spreading out the frequencies in an image (or equalising the image) is a simple way to improve dark or washed out images

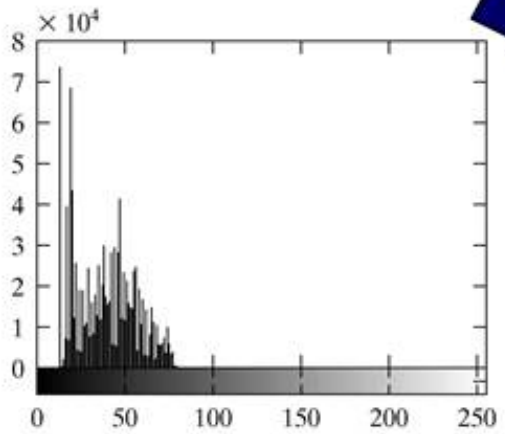
The formula for histogram equalisation is given where

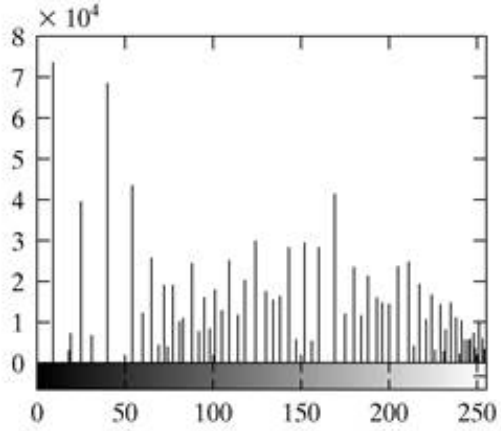
- r_k: input intensity
- s_k: processed intensity
- k: the intensity range (e.g 0.0 – 1.0)
- n_j: the frequency of intensity j
- n: the sum of all frequencies

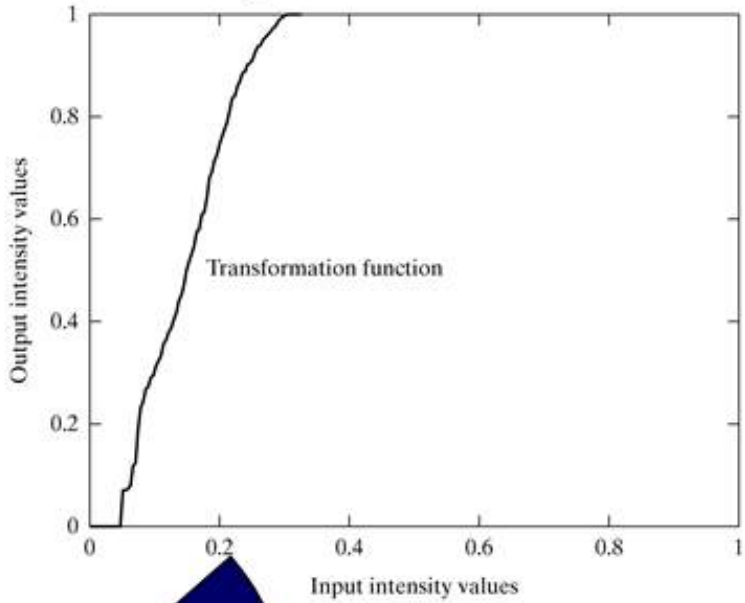
$$
\begin{aligned}
s_k &= T(r_k) \\
&= \sum_{j=1}^{k} p_r(r_j) \\
&= \sum_{j=1}^{k} \frac{n_j}{n}
\end{aligned}
$$

Equalisation Transformation Function



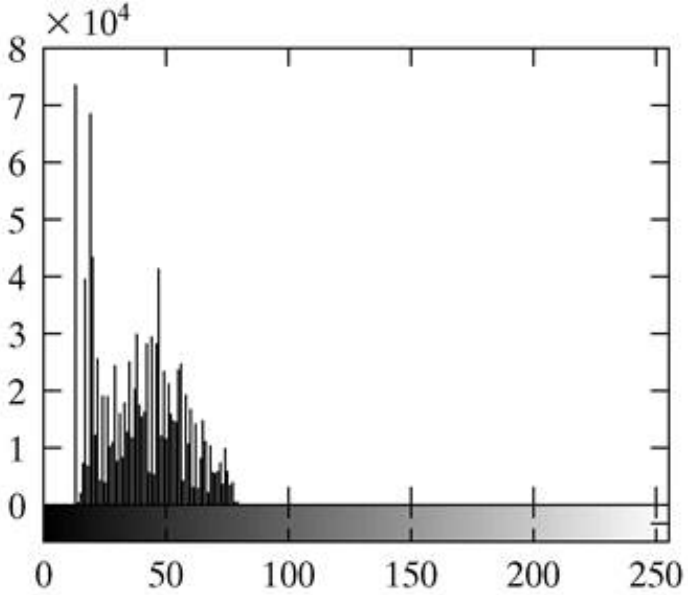


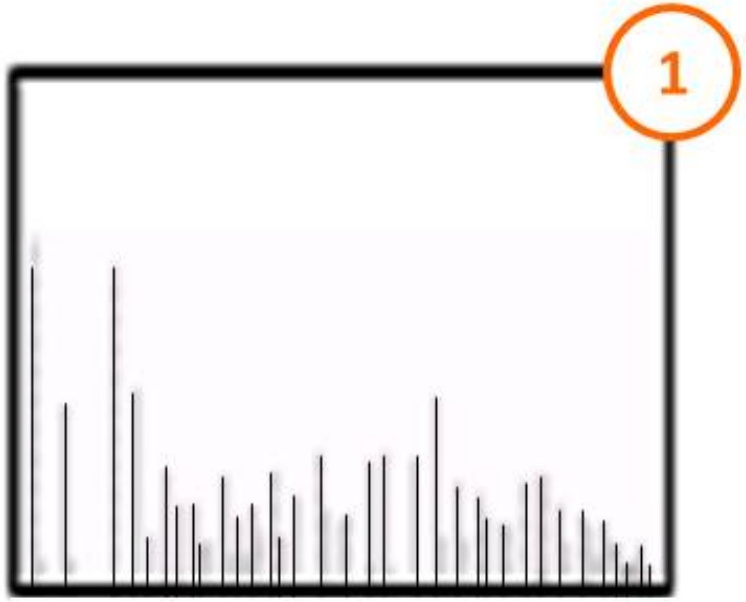


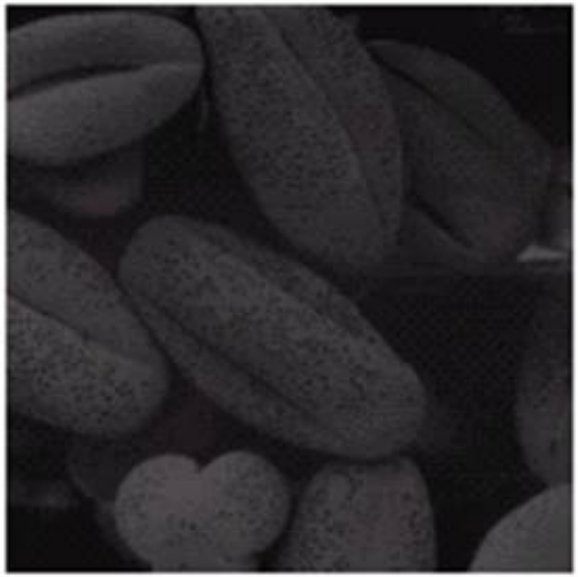


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Equalisation Examples



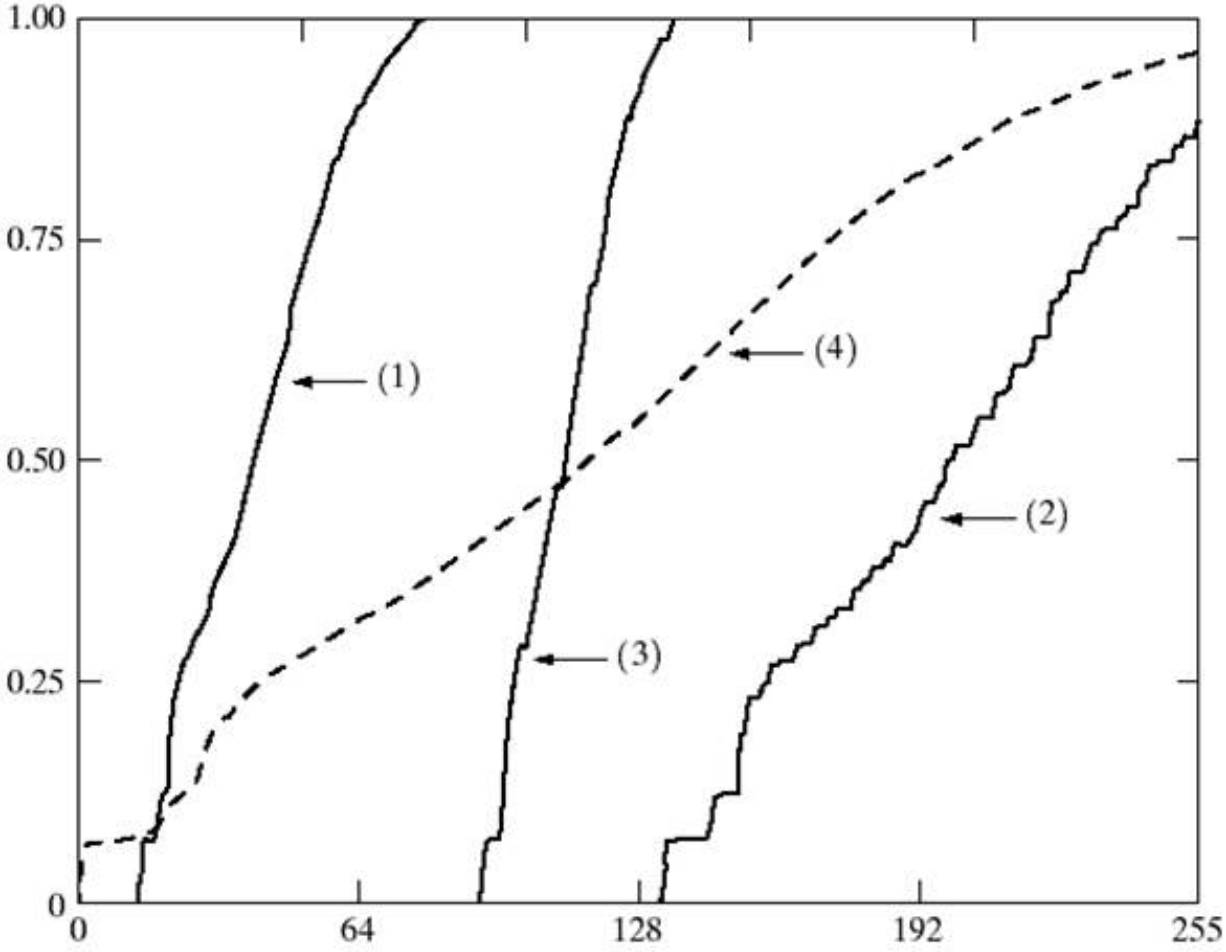




Images taken from Gonzalez & Woods, Digital Image Processing (2002)

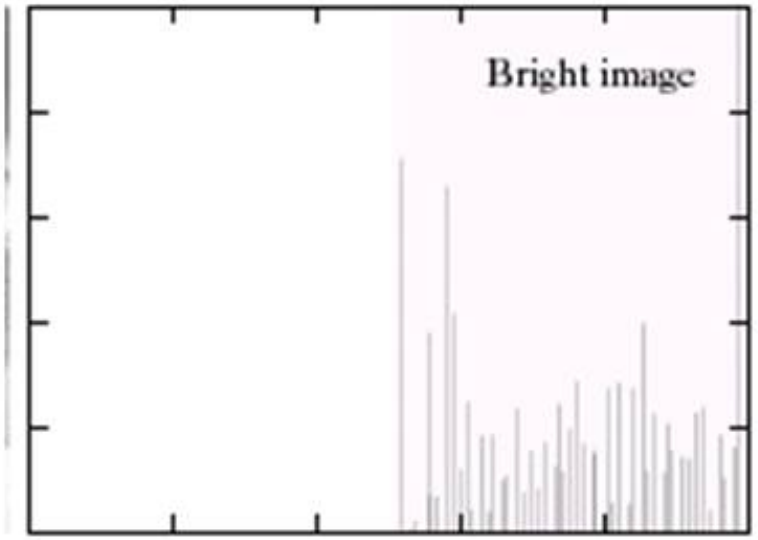
Equalisation Transformation Functions

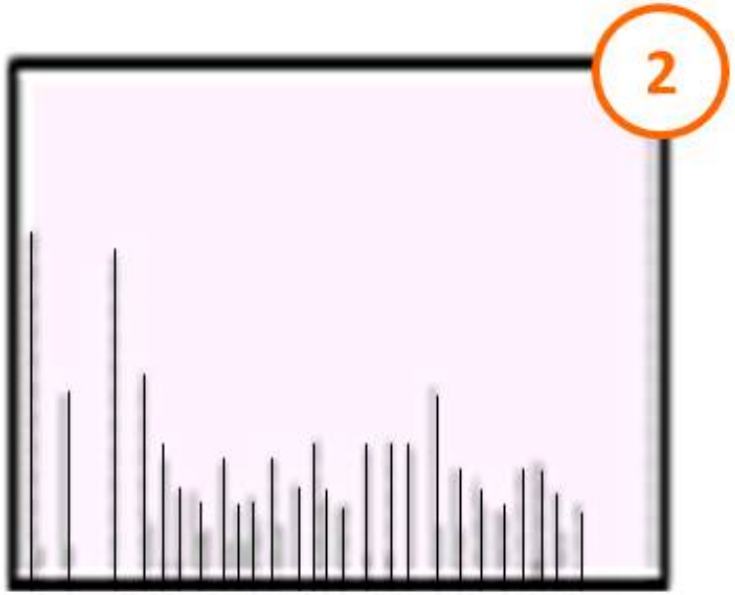
The functions used to equalise the images in the previous example

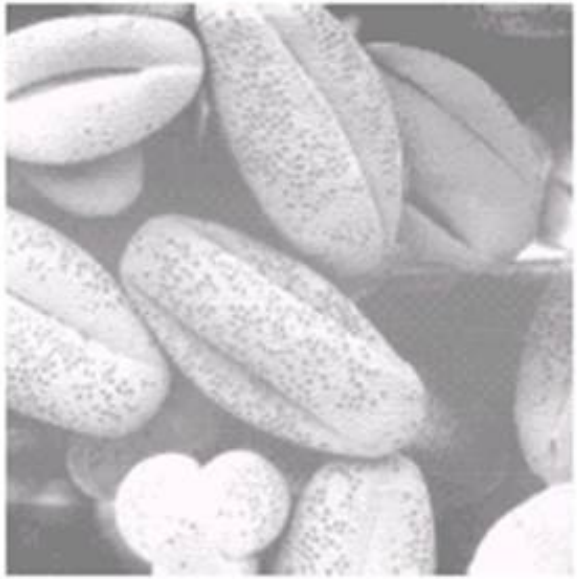


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Equalisation Examples



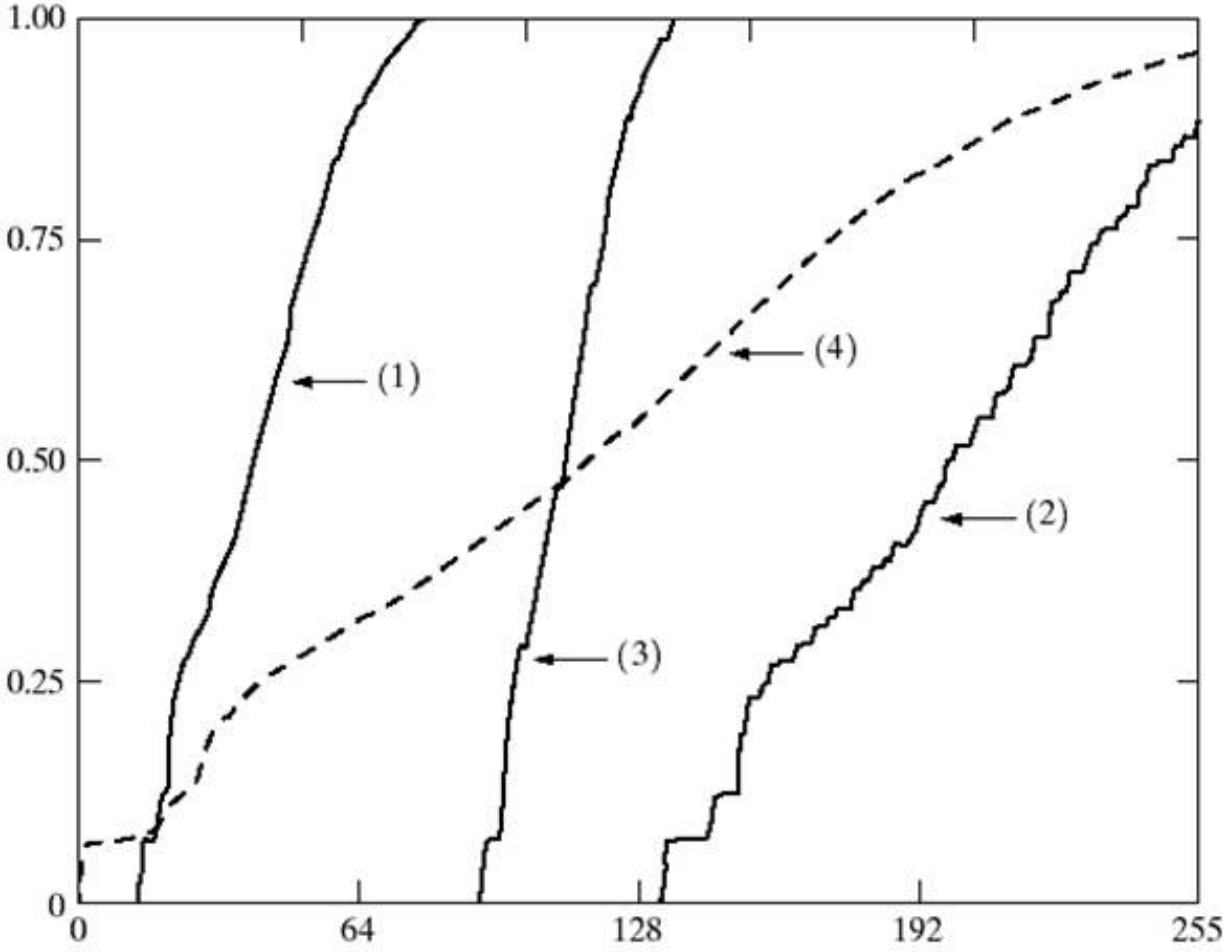




Images taken from Gonzalez & Woods, Digital Image Processing (2002)

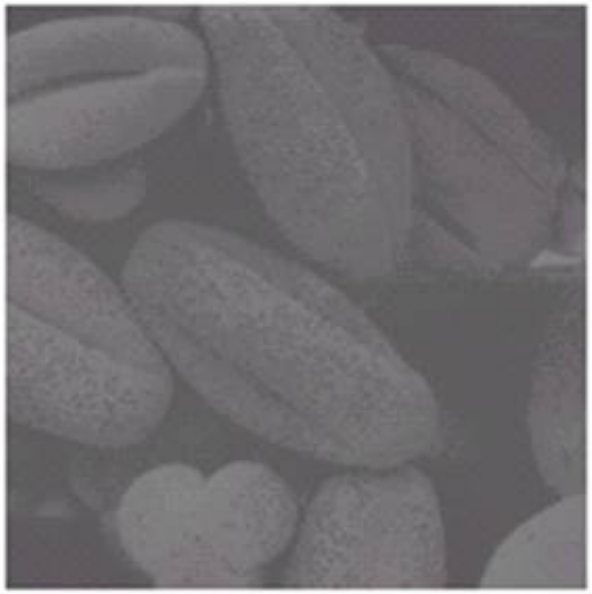
Equalisation Transformation Functions

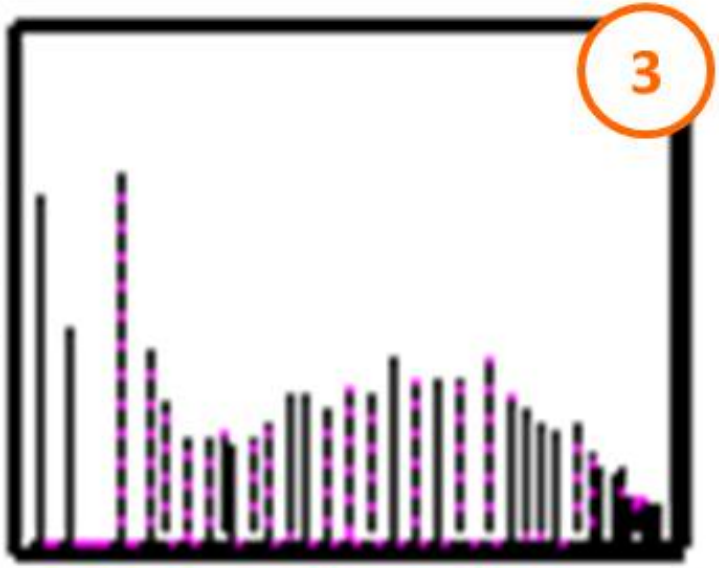
The functions used to equalise the images in the previous example

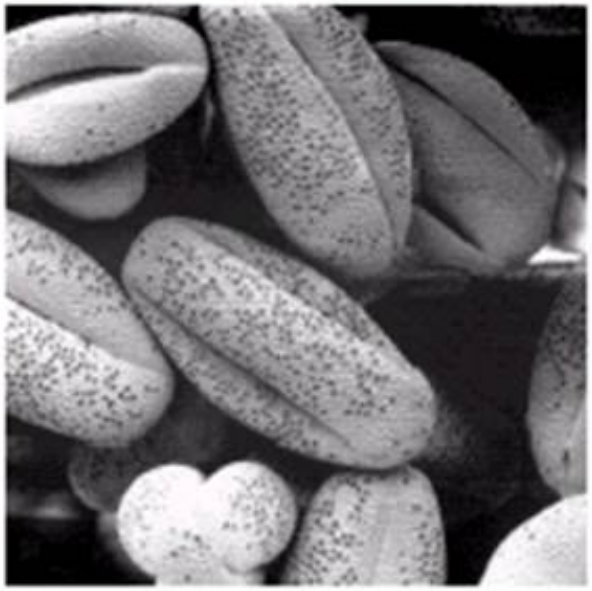


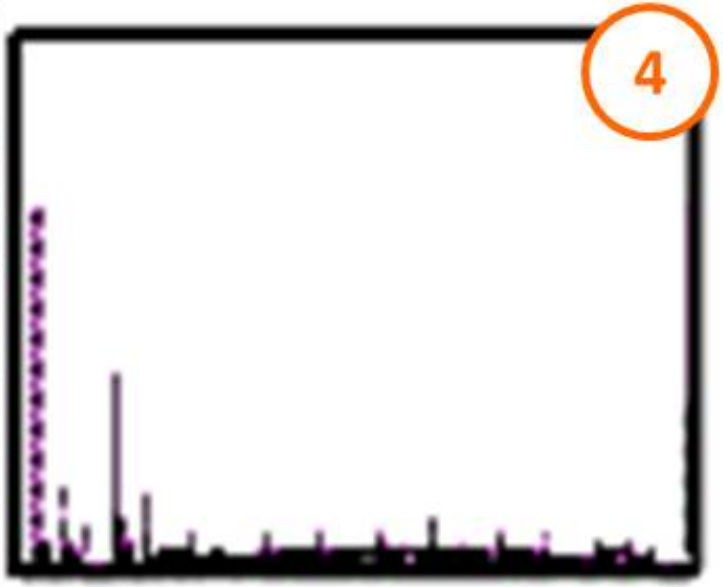
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Equalisation Examples (cont...)



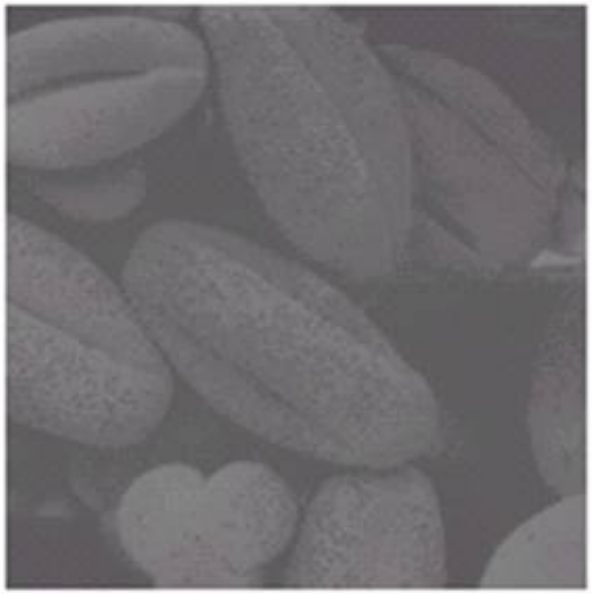


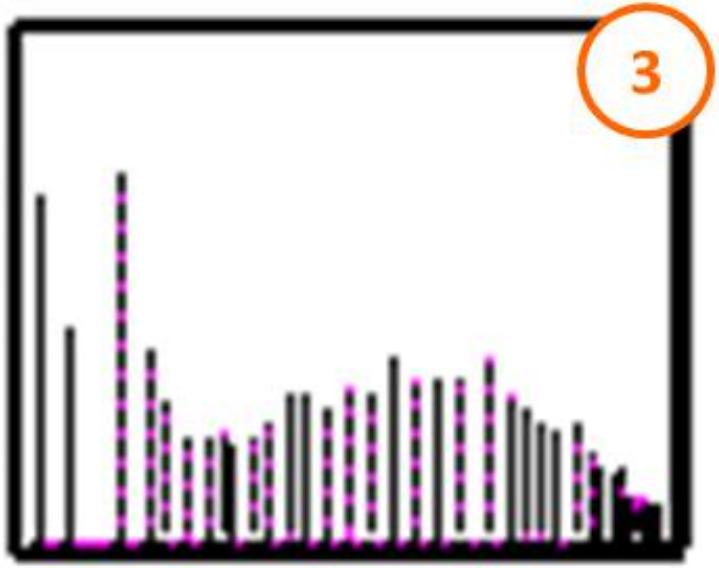


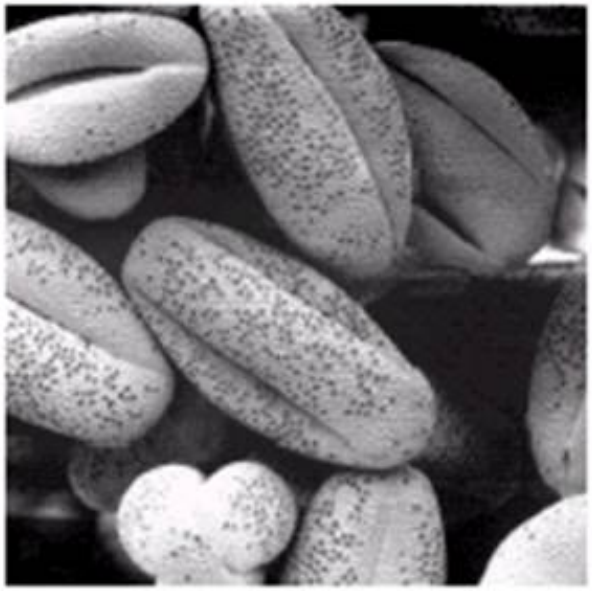


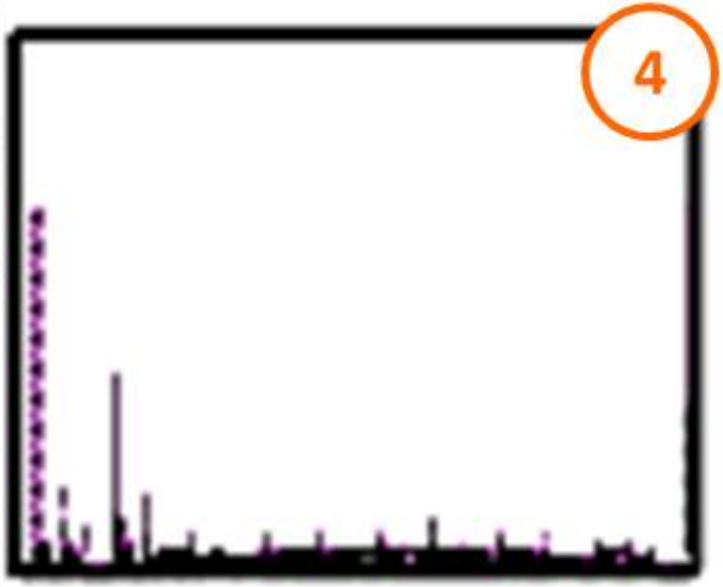
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Equalisation Examples (cont...)





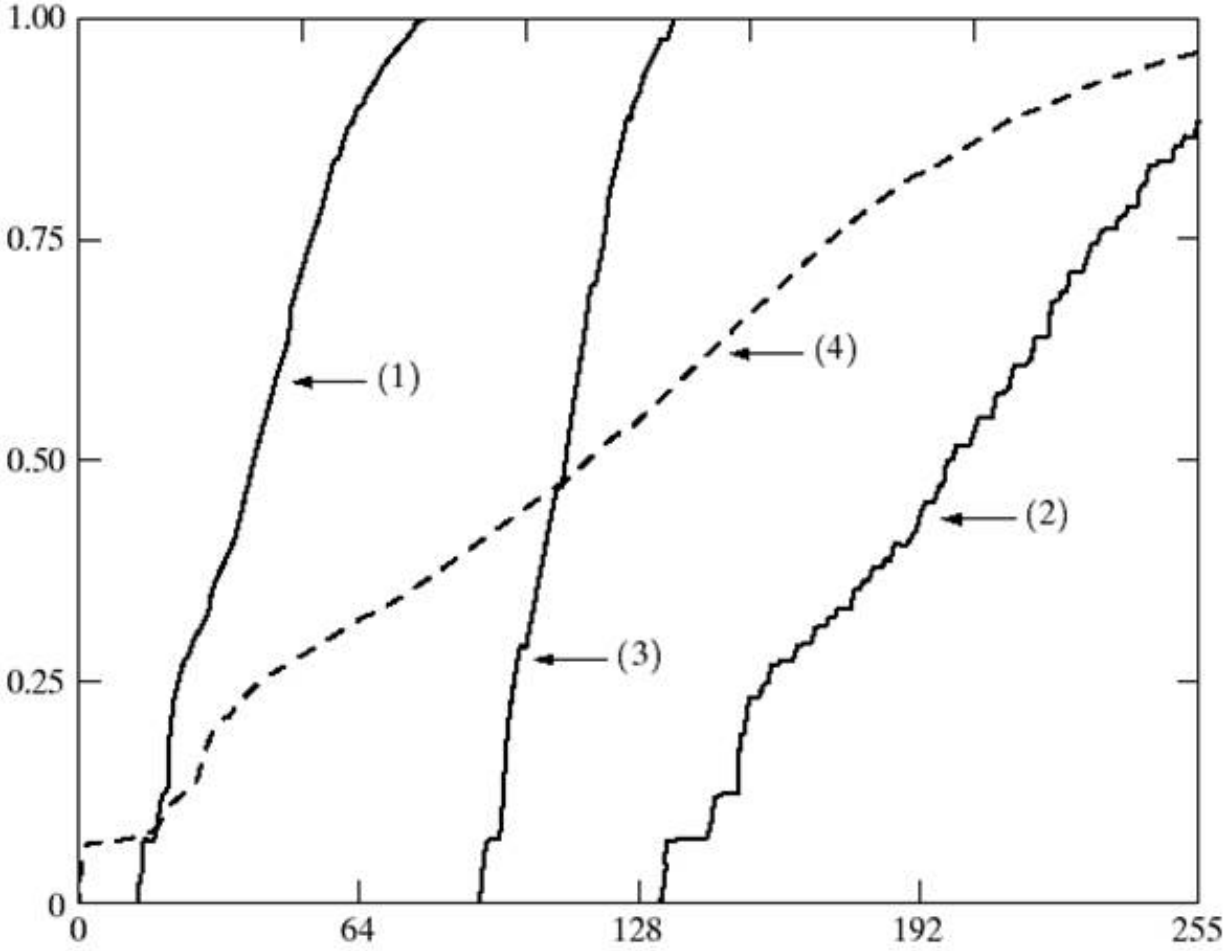




Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Equalisation Transformation Functions

The functions used to equalise the images in the previous examples



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

Summary

We have looked at:

- Different kinds of image enhancement
- Histograms
- Histogram equalisation

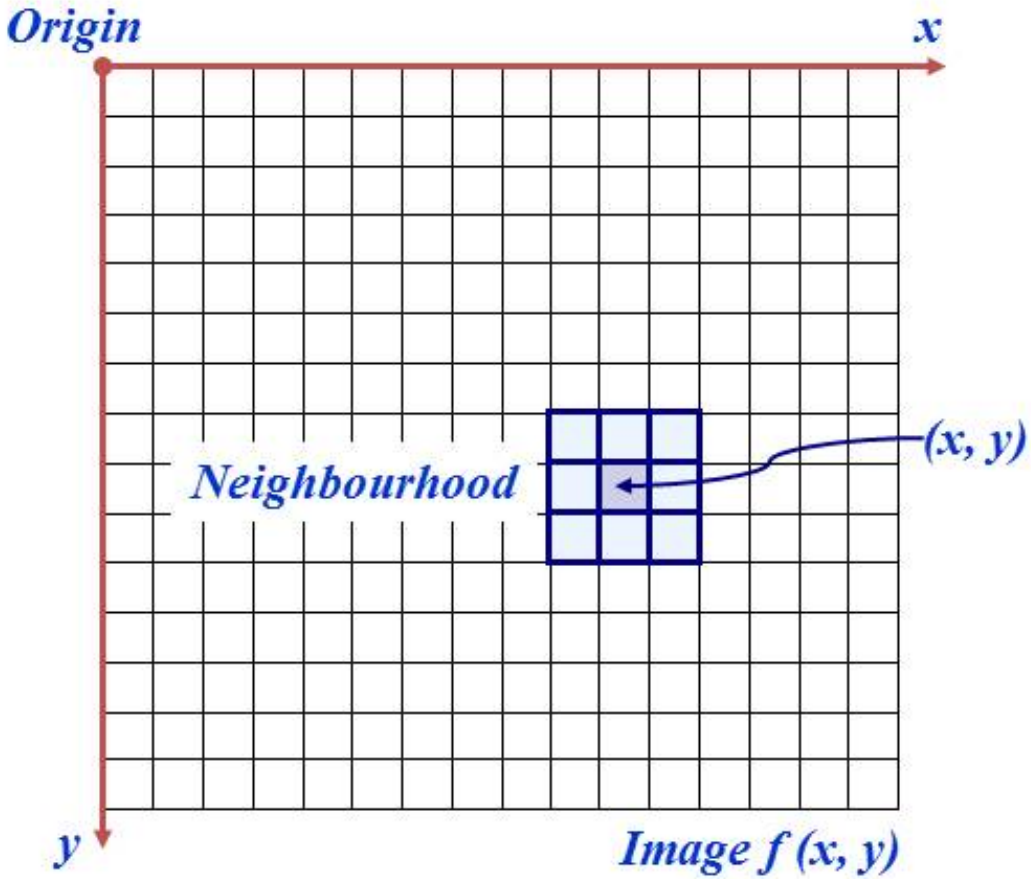
Next time we will start to look at point processing and some neighbourhood operations

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

Neighbourhood Operations

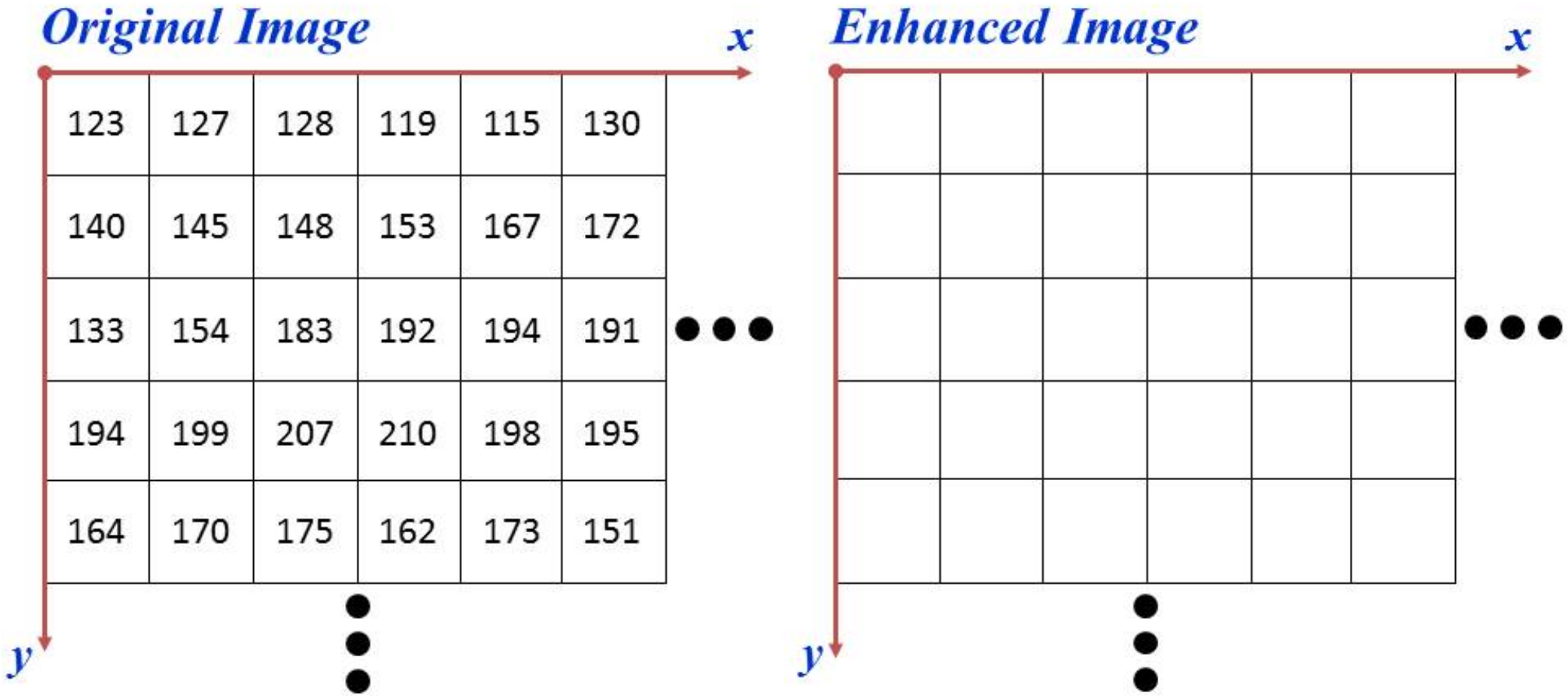
- Neighbourhood operations simply operate on a larger neighbourhood of pixels than point operations
- Neighbourhoods are mostly a rectangle around a central pixel
- Any size rectangle and any shape filter are possible



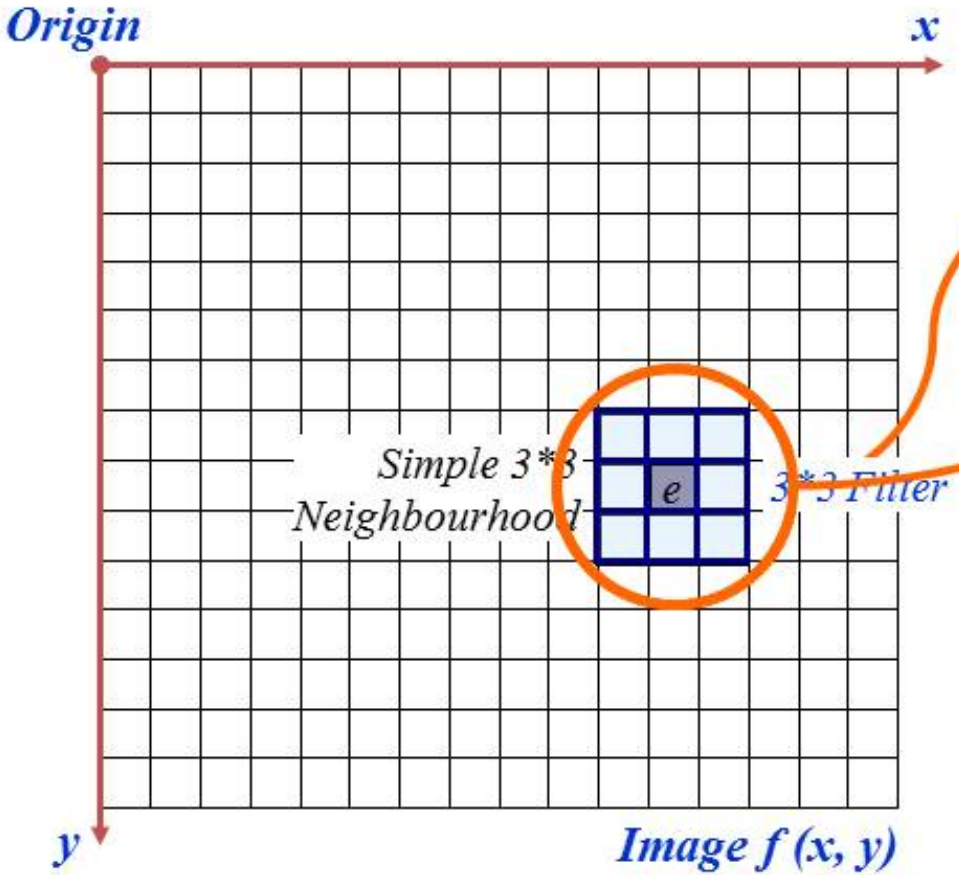
Simple Neighbourhood Operations

- 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] 15 is the median). Sometimes the median works better than the average

Simple Neighbourhood Operations Example



The Spatial Filtering Process



a	b	c
d	e	f
g	h	i

Original Image Pixels

$*$

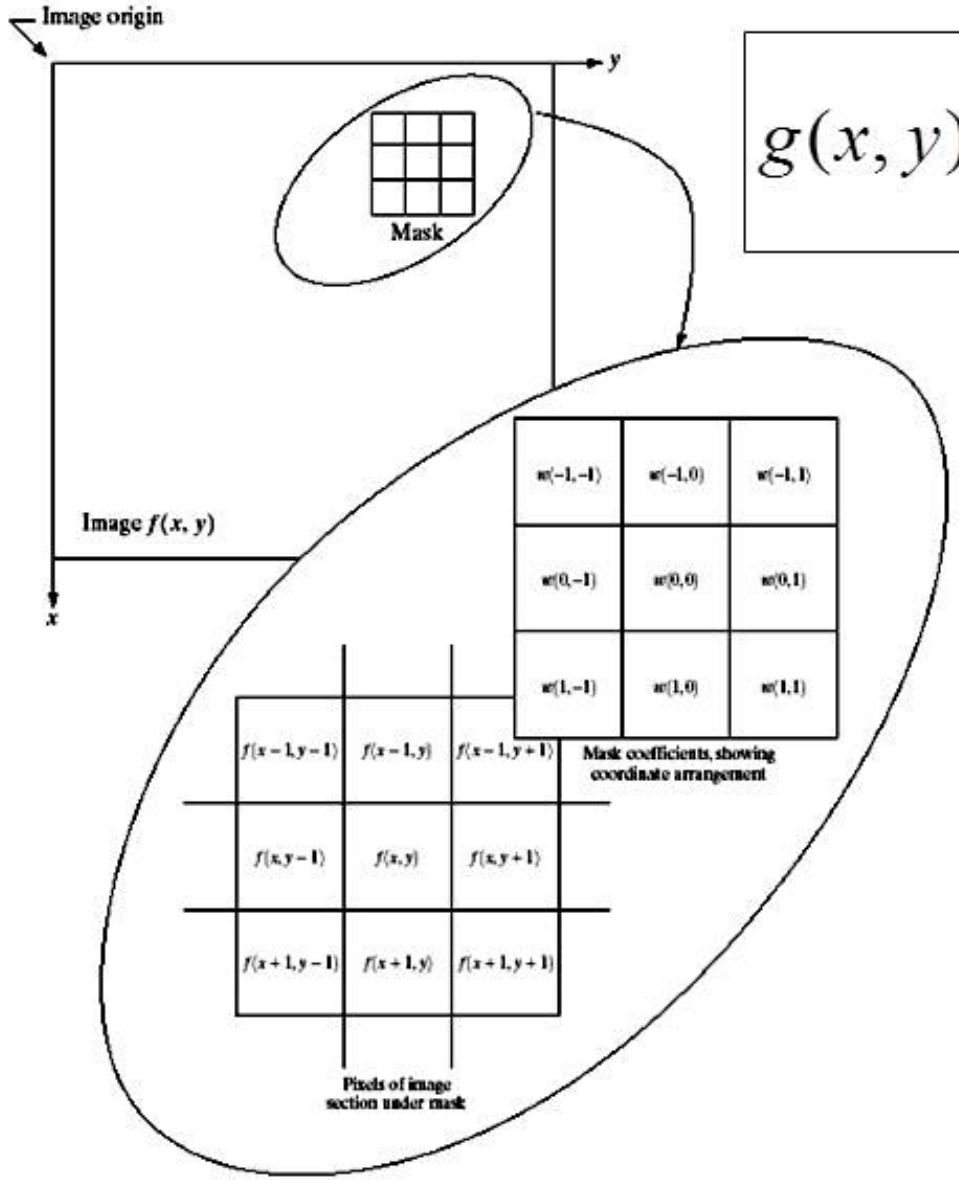
r	s	t
u	v	w
x	y	z

Filter

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

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

Spatial Filtering: Equation Form



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

Filtering can be given in equation form as shown above

Notations are based on the image shown to the left

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

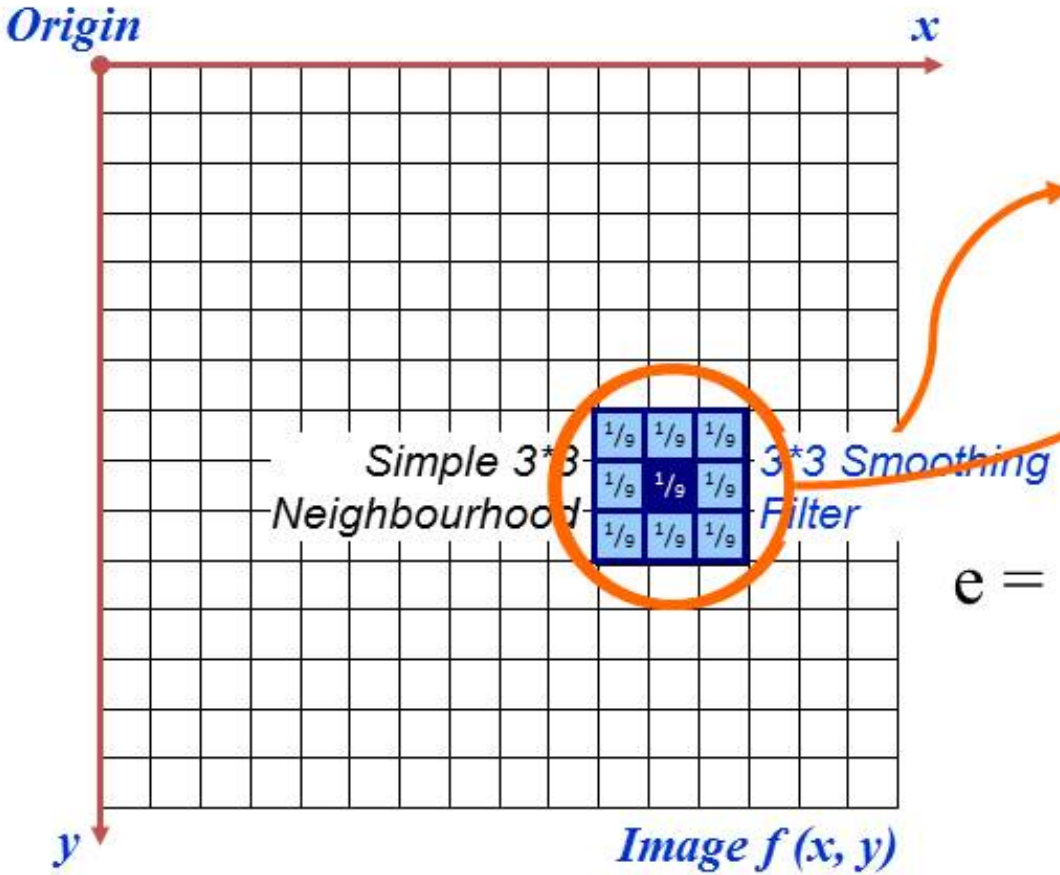
Smoothing Spatial Filters

- 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

$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$

Simple averaging filter

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

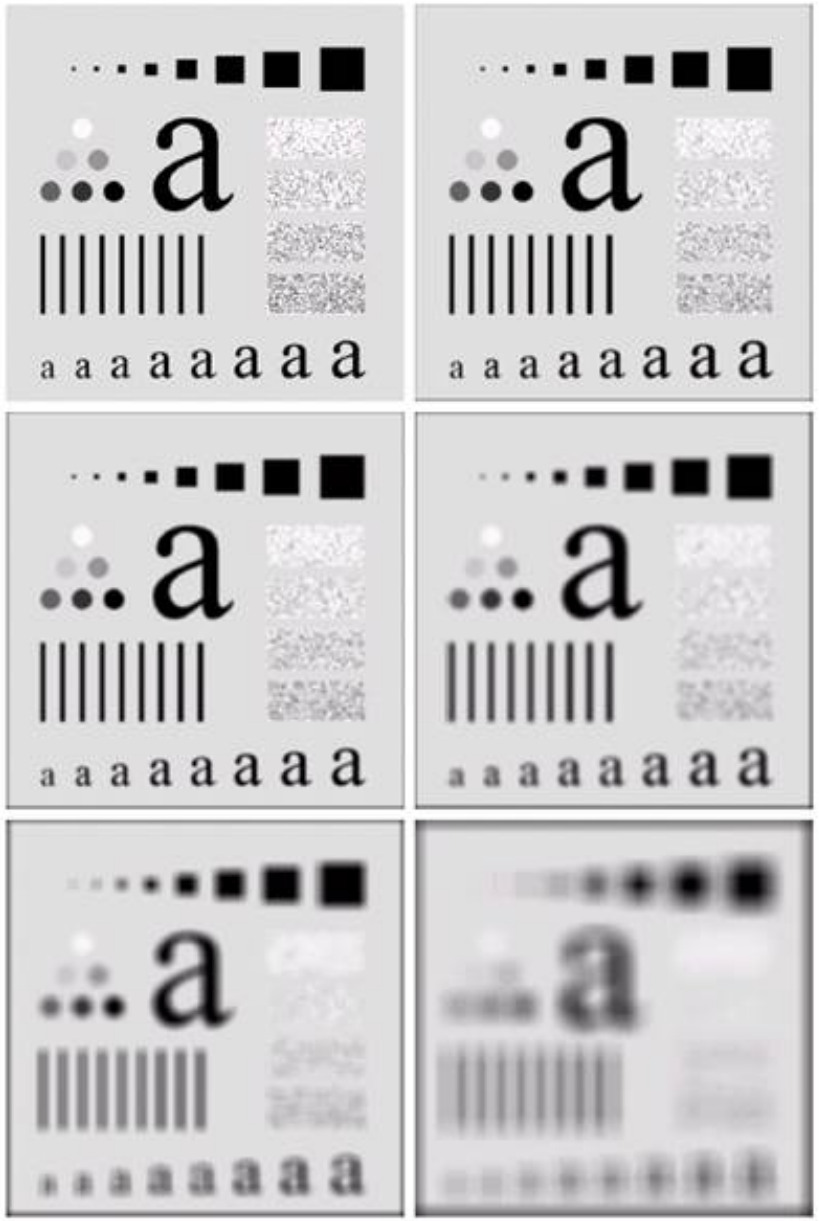


Image Smoothing Example

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

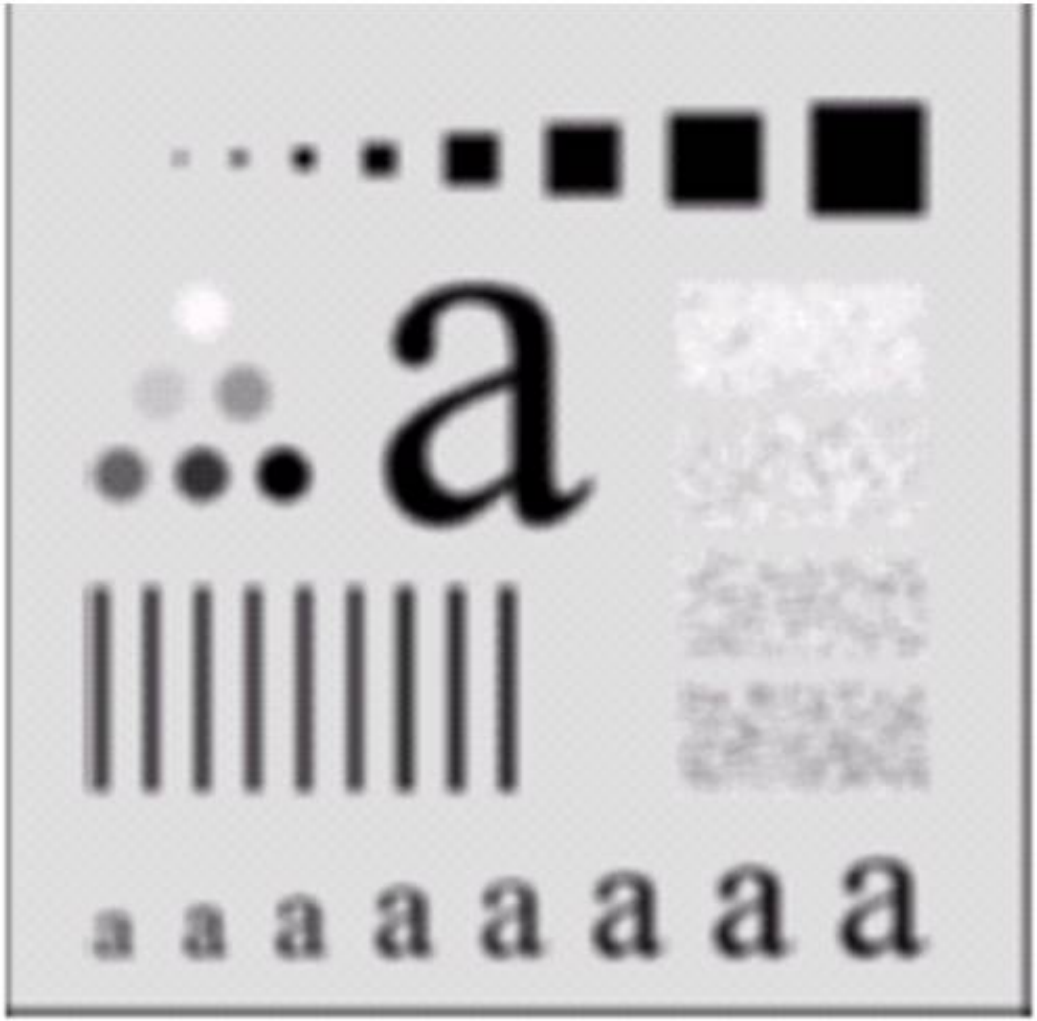
Image Smoothing Example

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

Image Smoothing Example

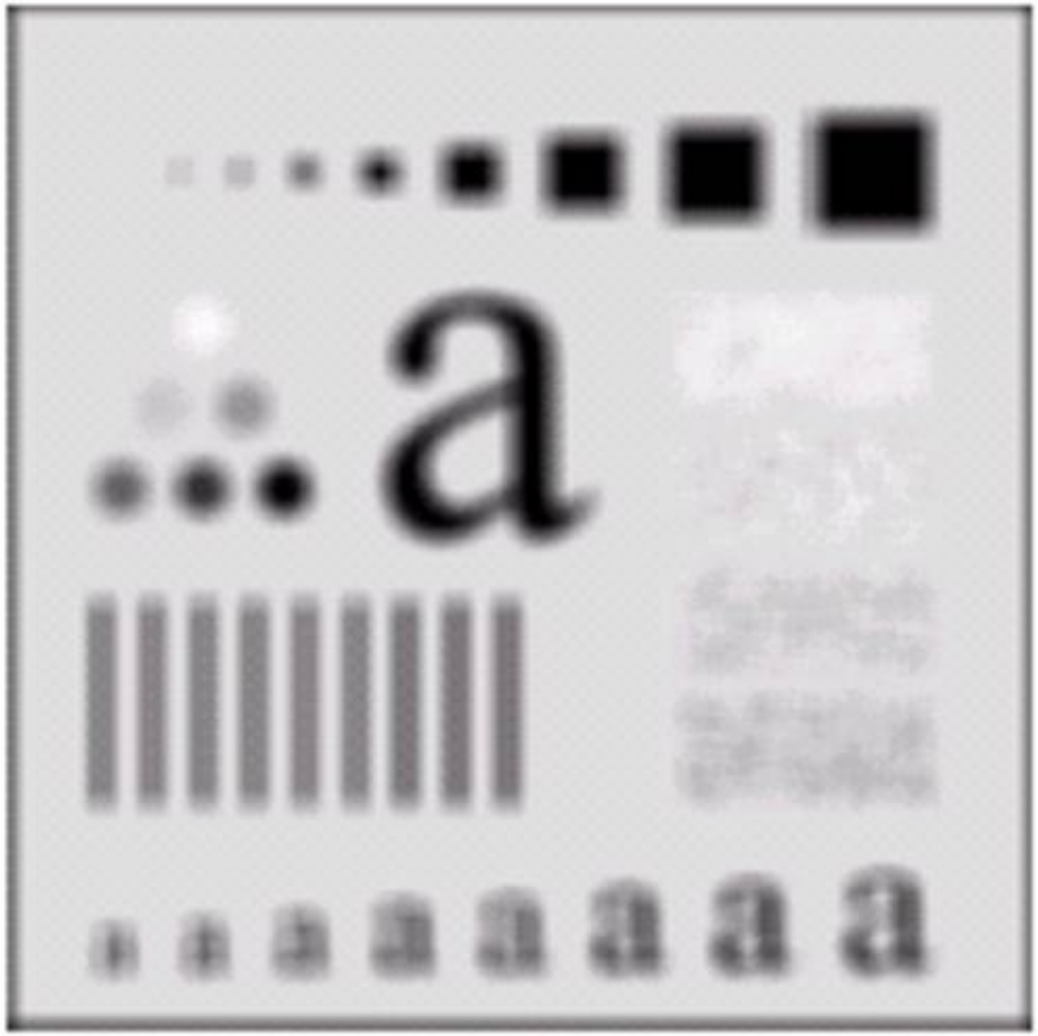
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Image Smoothing Example



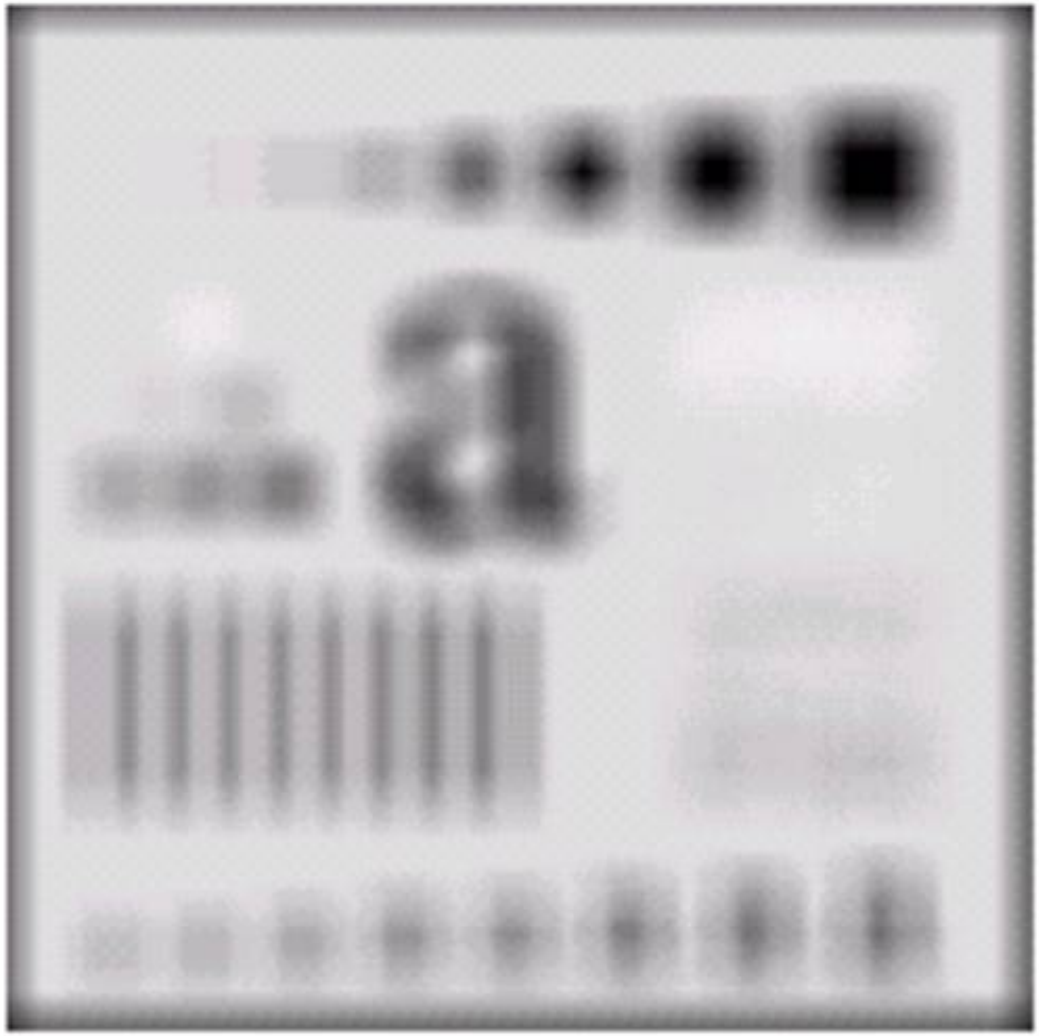
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Image Smoothing Example



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

Image Smoothing Example



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

Weighted Smoothing Filters

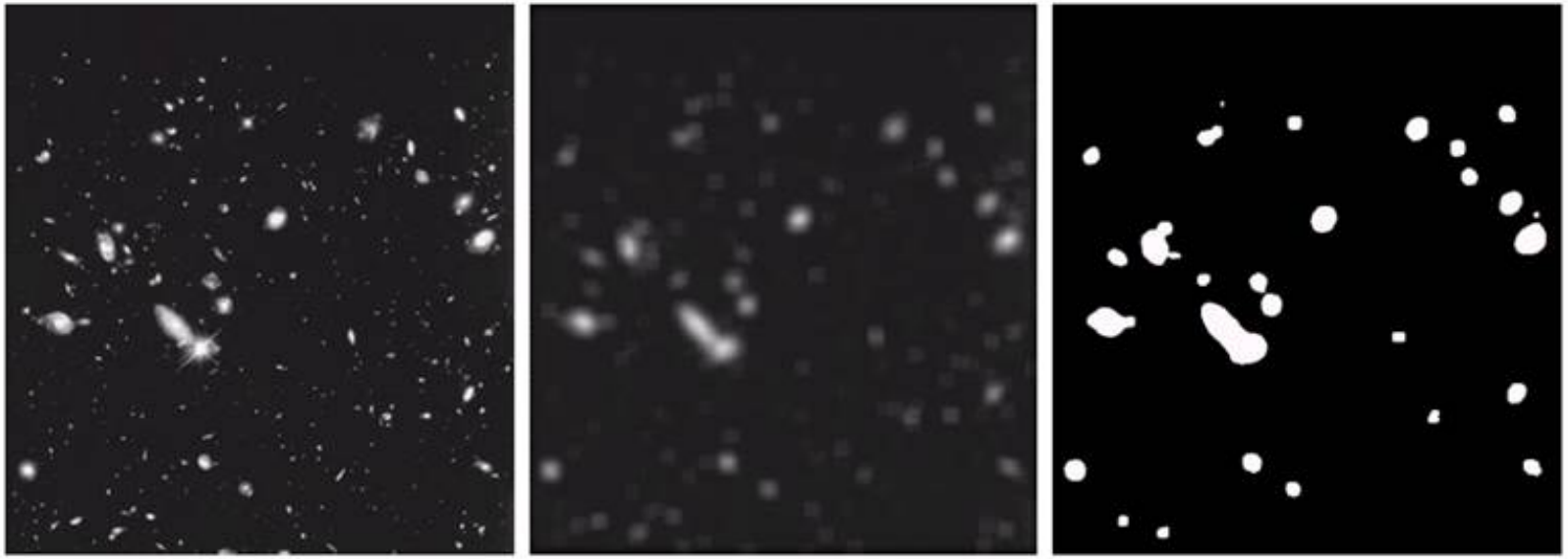
- More effective smoothing filters can be generated by allowing different pixels in the neighbourhood different weights in the averaging function
  - Pixels closer to the central pixel are more important
  - Often referred to as a *weighted averaging*

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

Weighted averaging filter

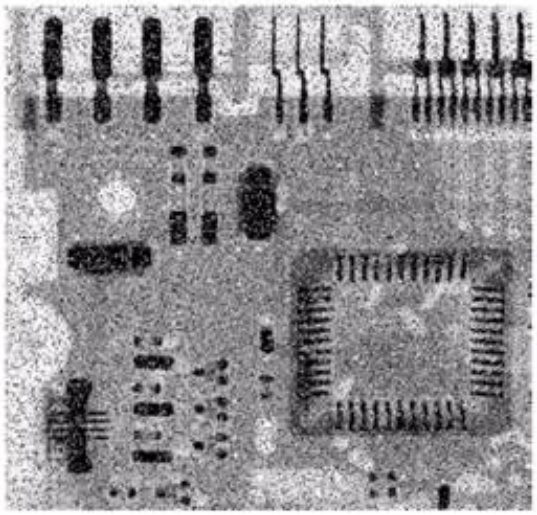
Another Smoothing Example

- 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

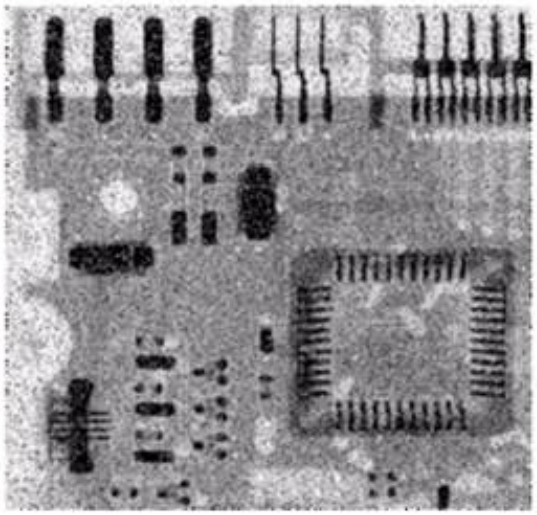


Image After Averaging Filter

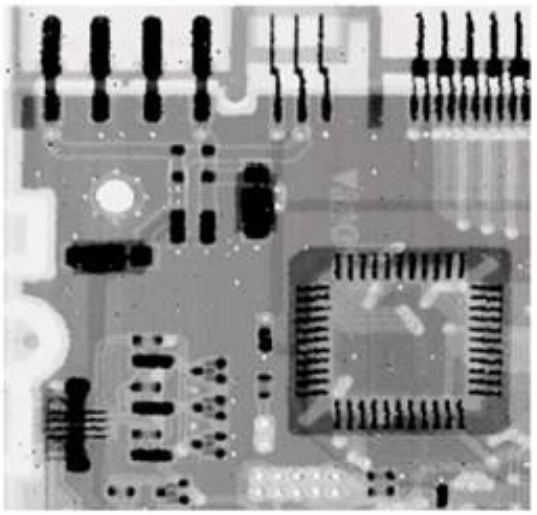
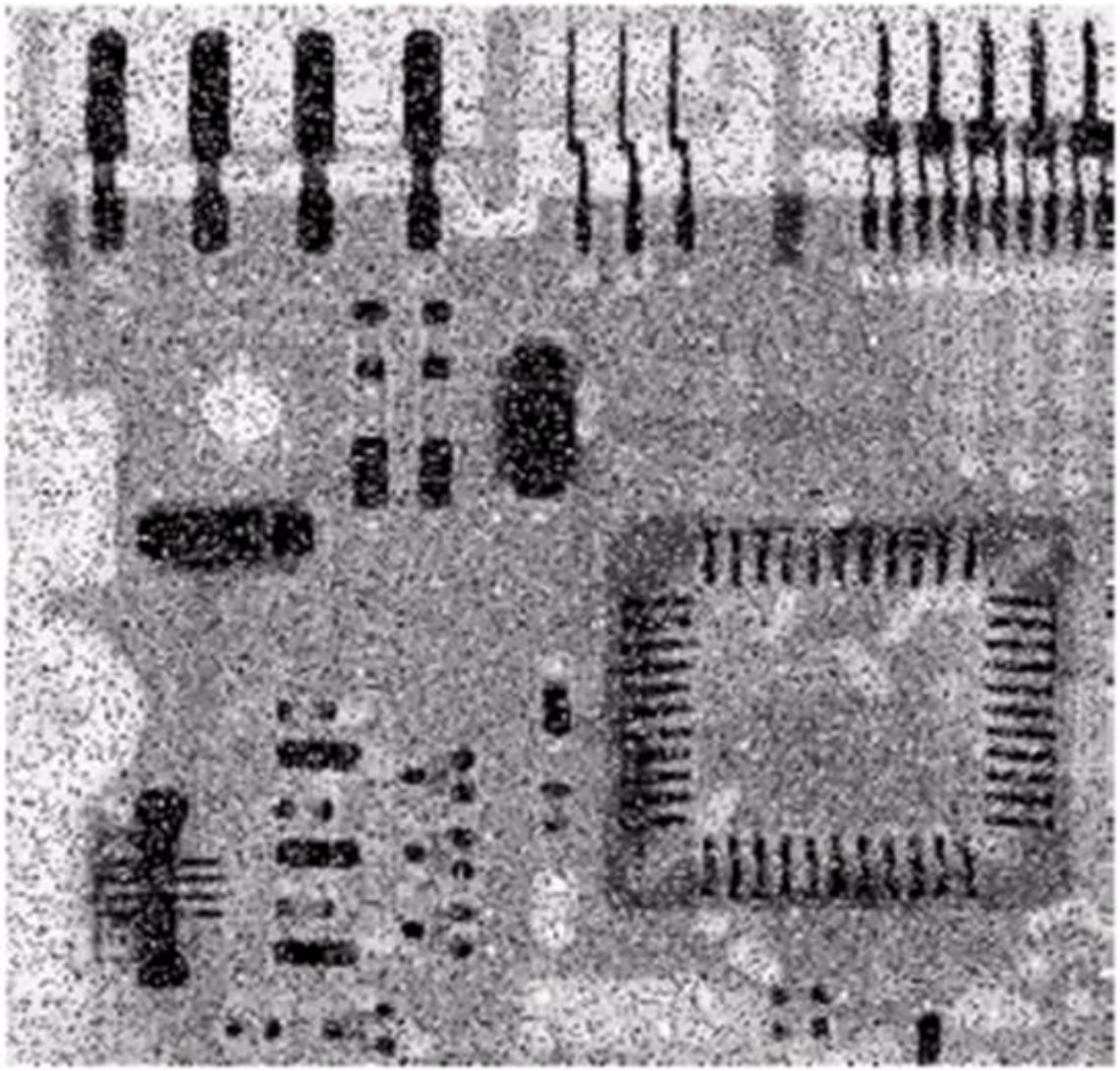


Image After Median Filter

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

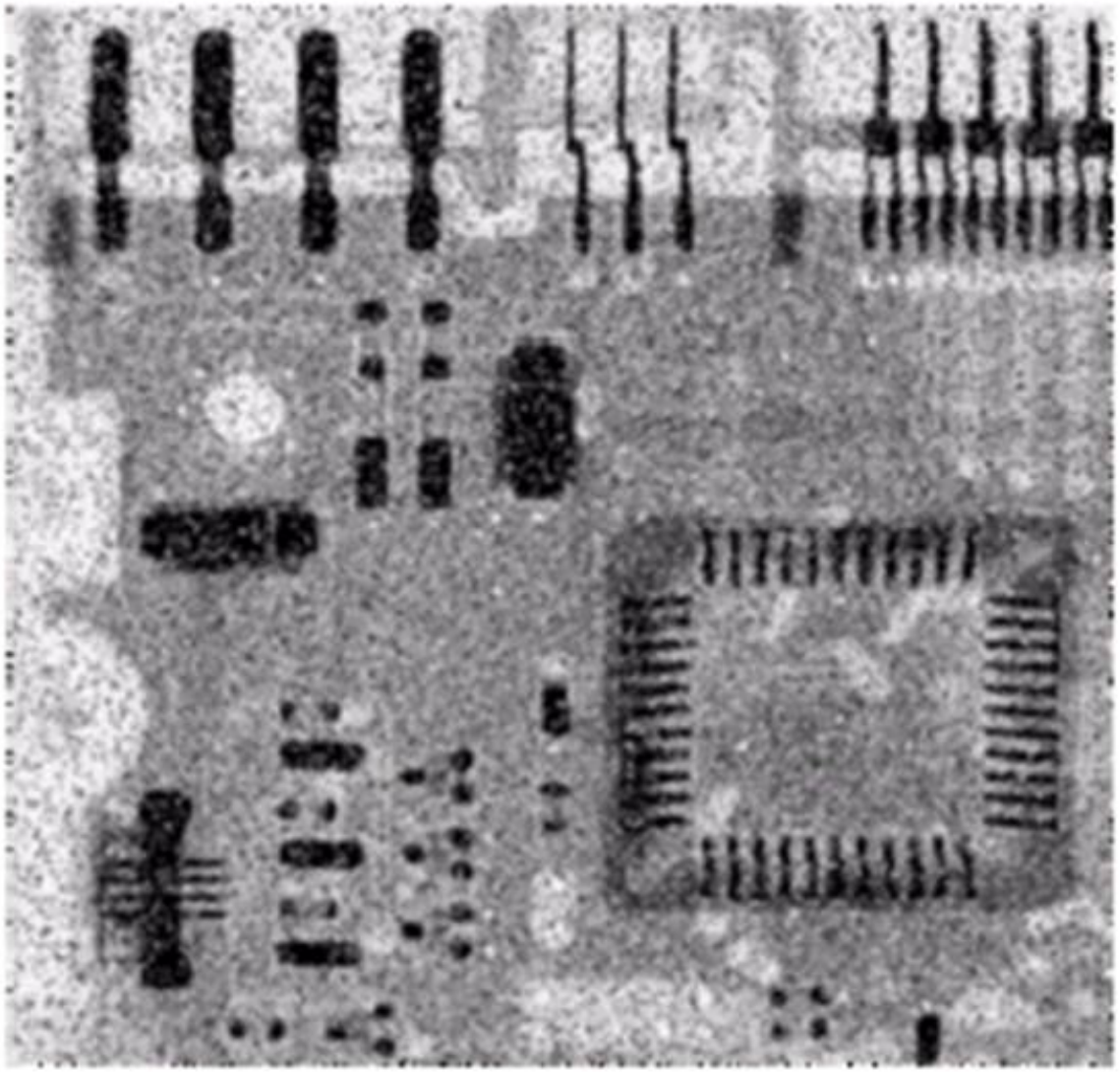
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Averaging Filter Vs. Median Filter Example



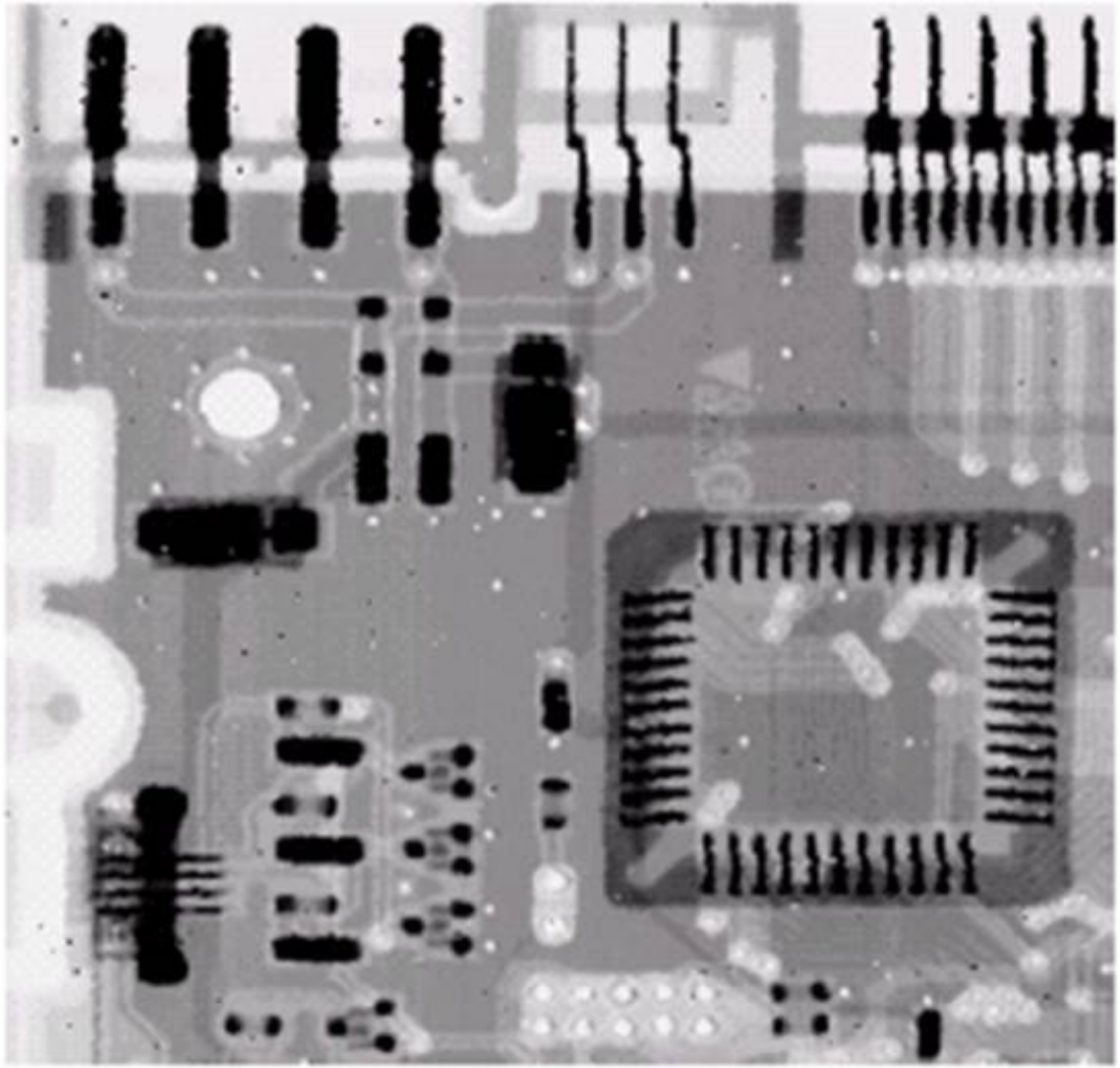
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Averaging Filter Vs. Median Filter Example



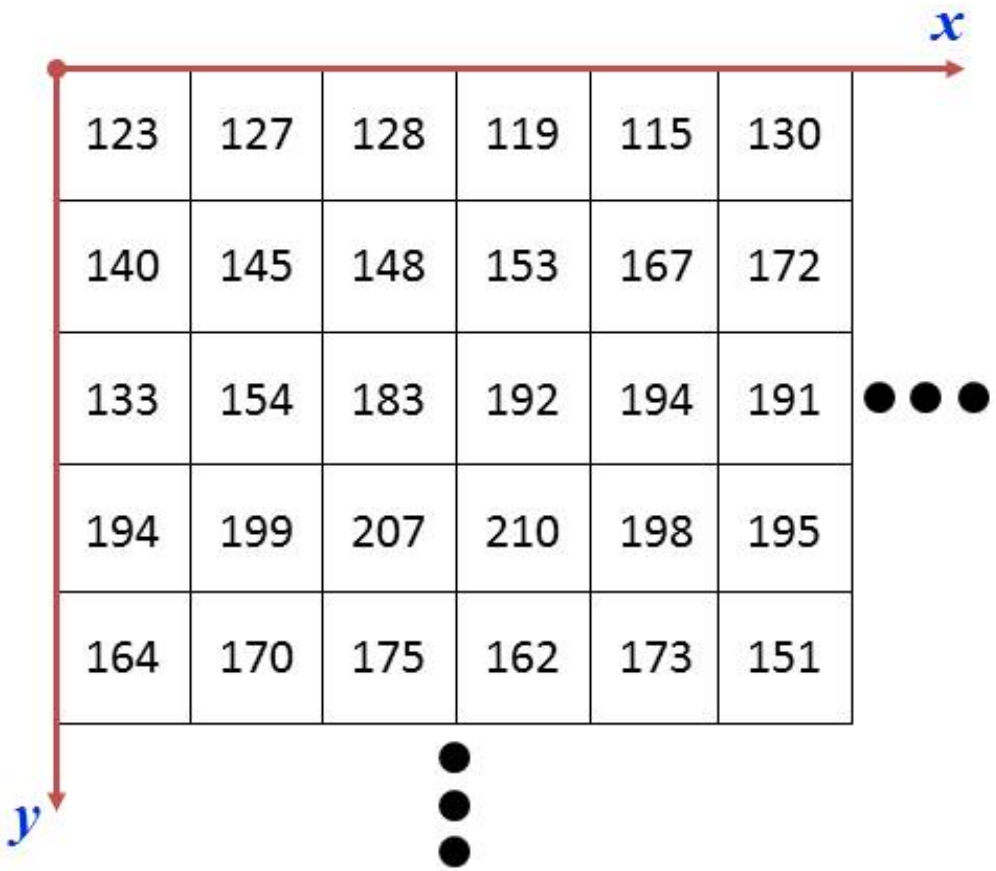
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Averaging Filter Vs. Median Filter Example



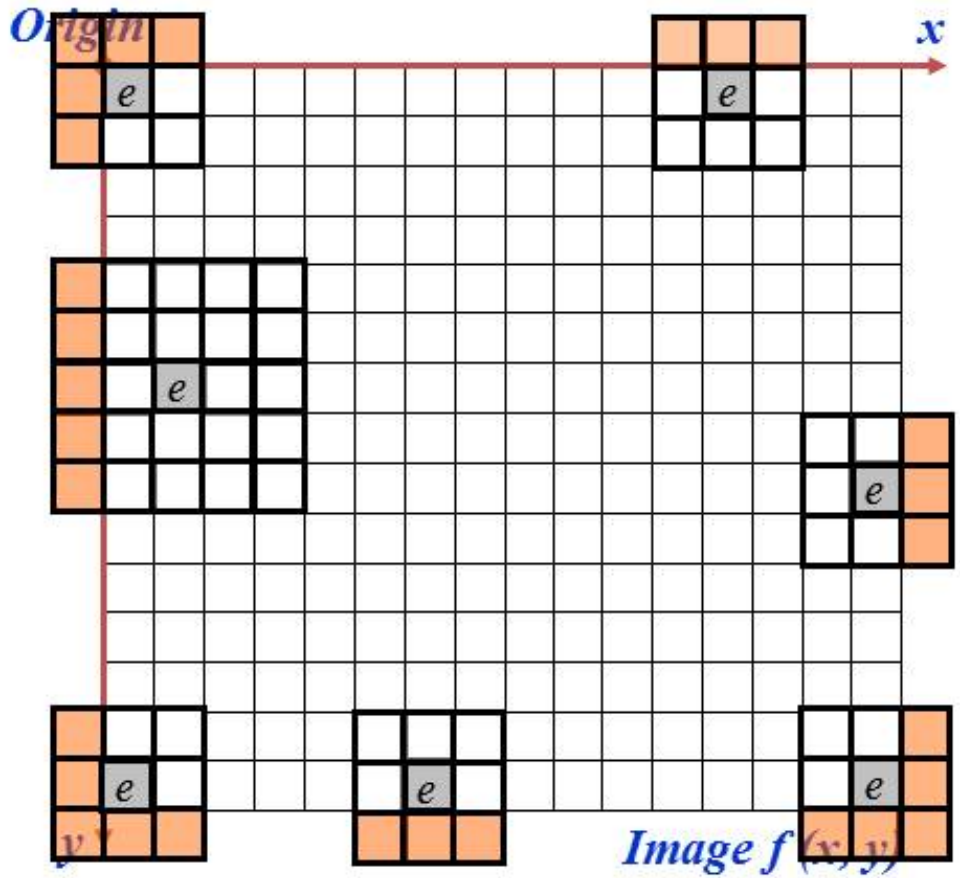
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Simple Neighbourhood Operations Example



Strange Things Happen At The Edges!

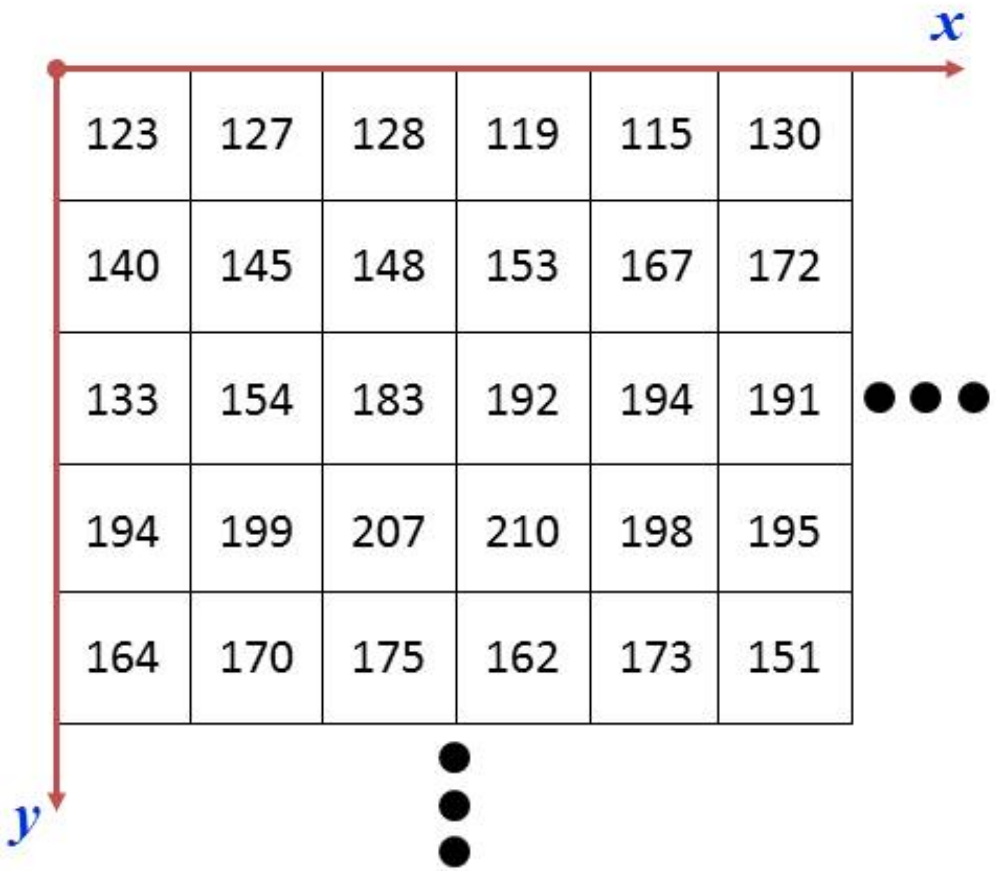
At the edges of an image we are missing pixels to form a neighbourhood



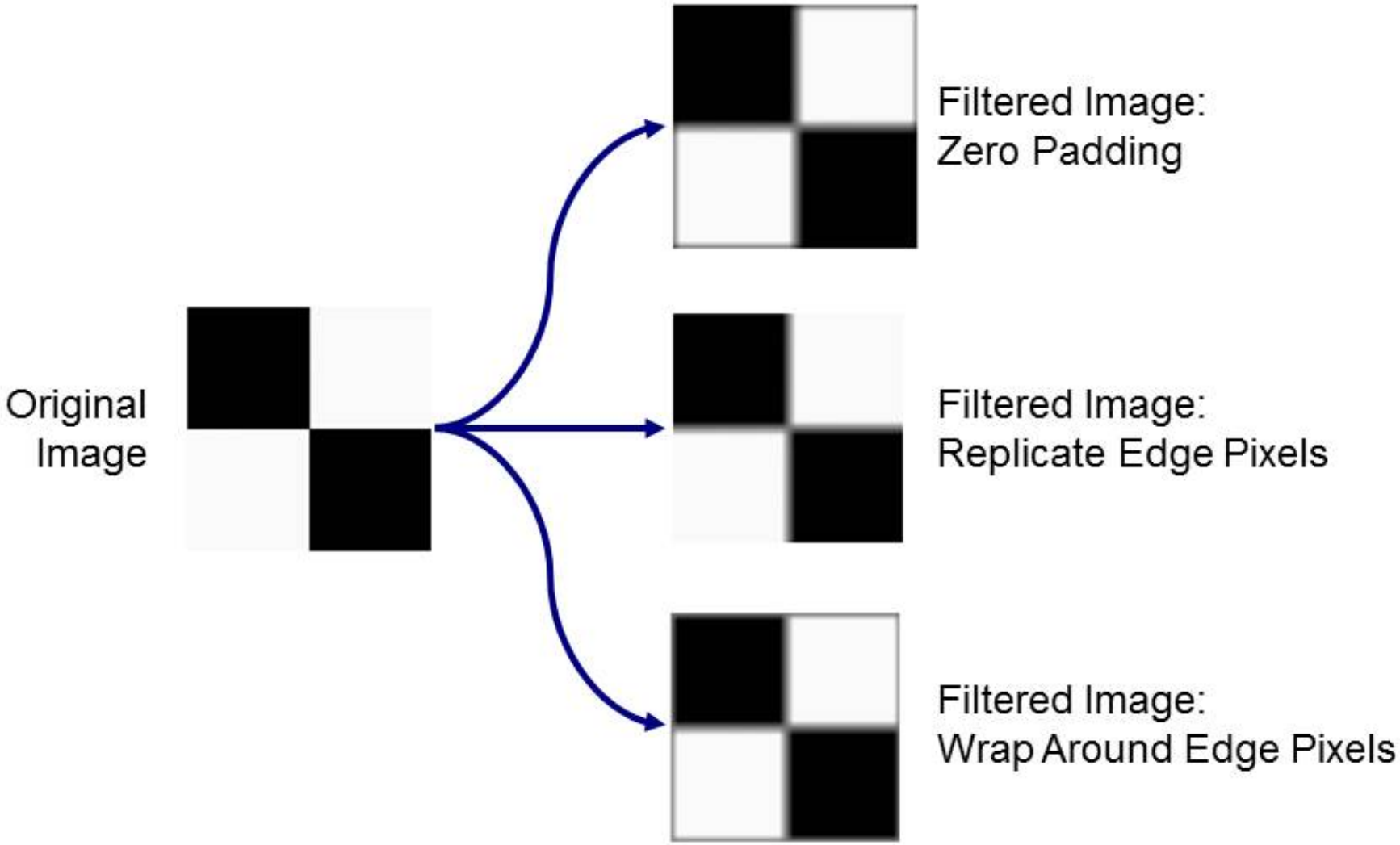
Strange Things Happen At The Edges! (cont…)

- 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

Simple Neighbourhood Operations Example

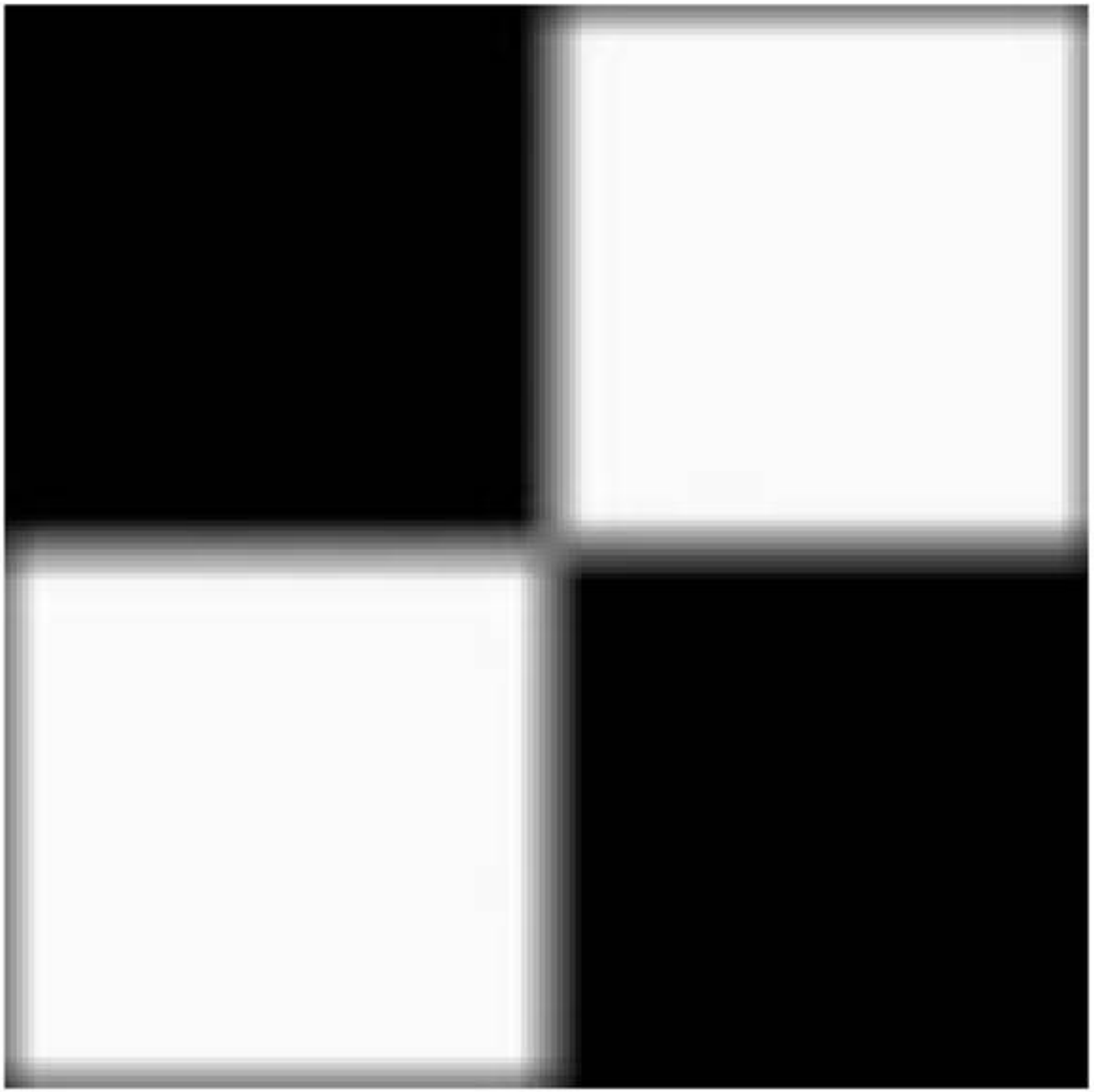


Strange Things Happen At The Edges! (cont…)



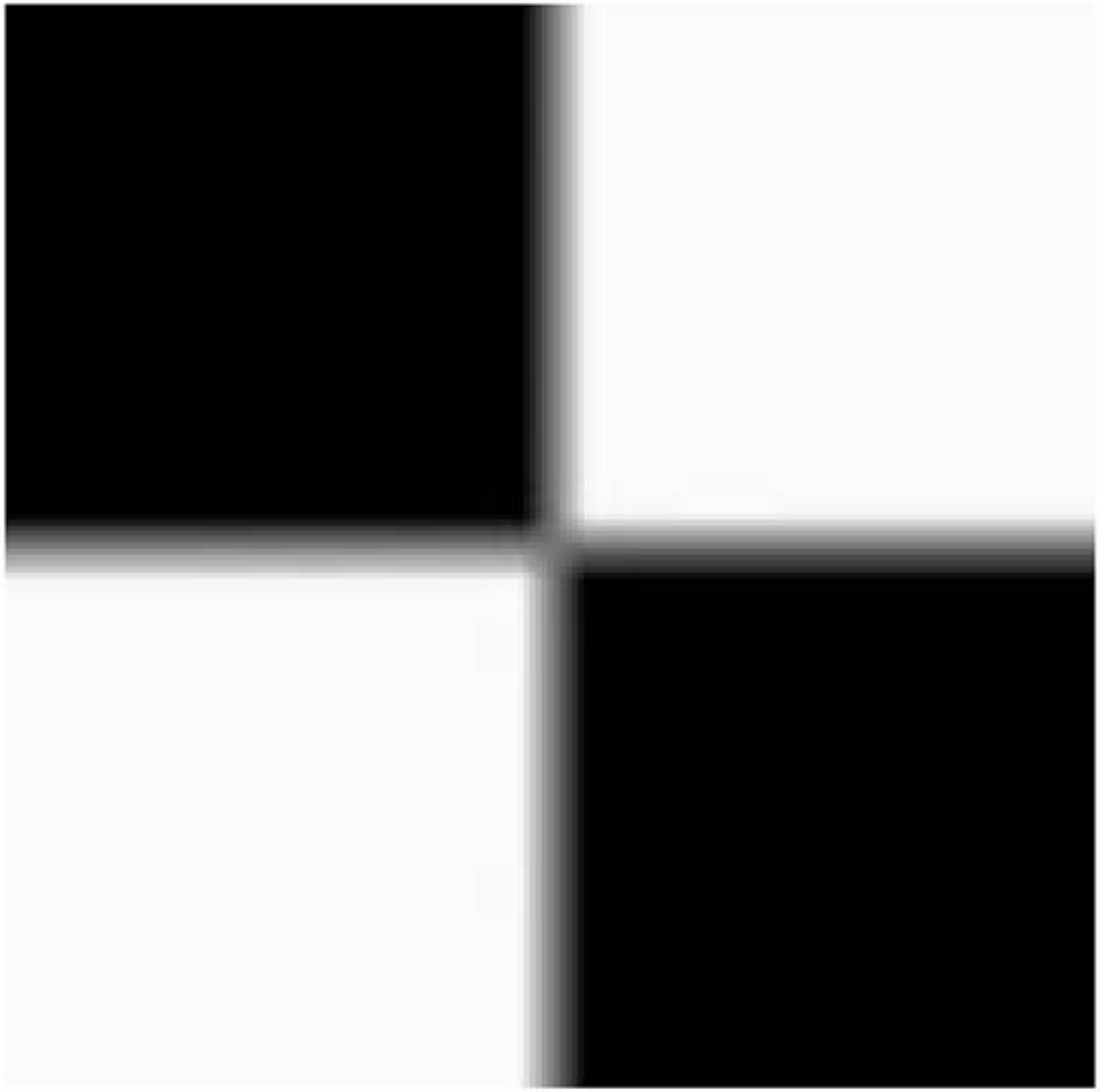
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Strange Things Happen At The Edges! (cont…)



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

Strange Things Happen At The Edges! (cont…)



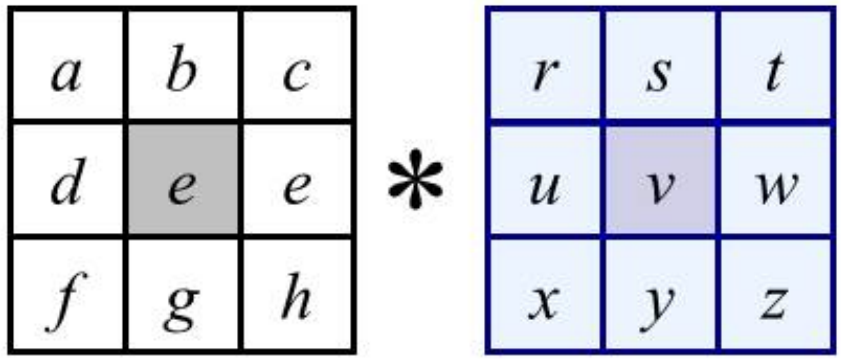
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Strange Things Happen At The Edges! (cont…)

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

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



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

- For symmetric filters it makes no difference

Summary

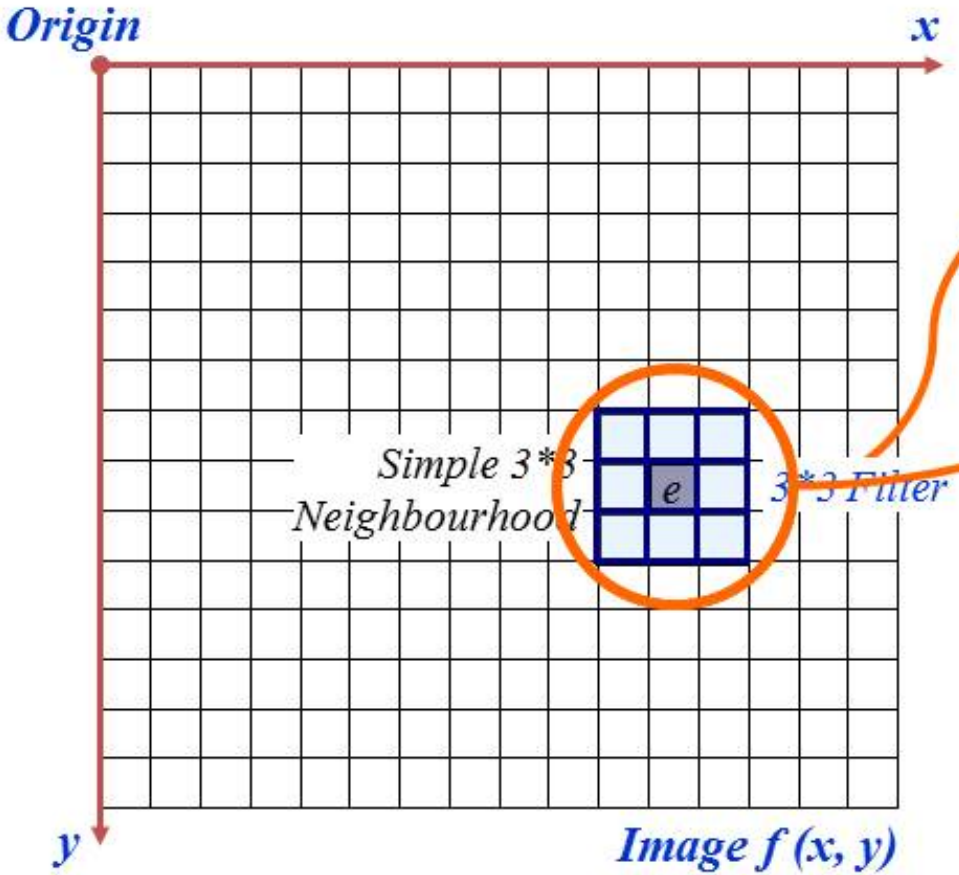
- In this lecture we have looked at the idea of spatial filtering and in particular:
  - Neighbourhood operations
  - The filtering process
  - Smoothing filters
  - Dealing with problems at image edges when using filtering
  - Correlation and convolution
- Next time we will looking at sharpening filters and more on filtering and image enhancement

Contents

In this lecture we will look at more spatial filtering techniques

- Spatial filtering refresher
- Sharpening filters
  - 1st derivative filters
  - 2nd derivative filters
- Combining filtering techniques

Spatial Filtering Refresher



a	*b*	*c*
d	*e*	*f*
g	*h*	*i*

Original Image Pixels

*

r	*s*	*t*
u	*v*	*w*
x	*y*	*z*

Filter

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

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

Sharpening Spatial Filters

Previously we have looked at smoothing filters which remove fine detail

Sharpening spatial filters seek to highlight fine detail

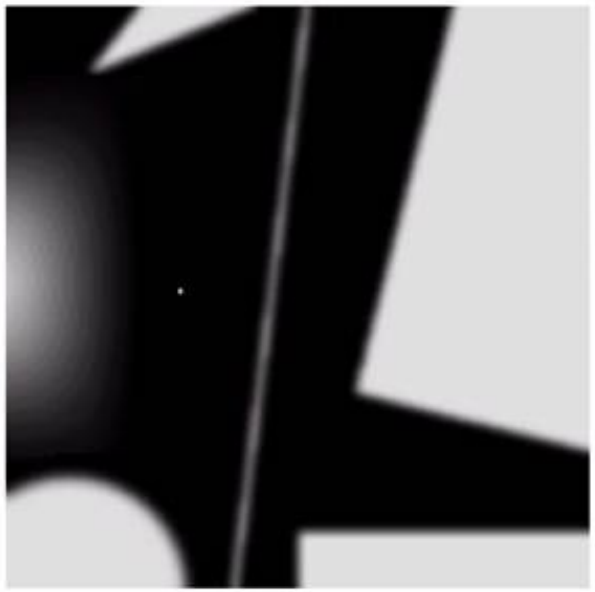
- Remove blurring from images
- Highlight edges

Sharpening filters are based on *spatial differentiation*

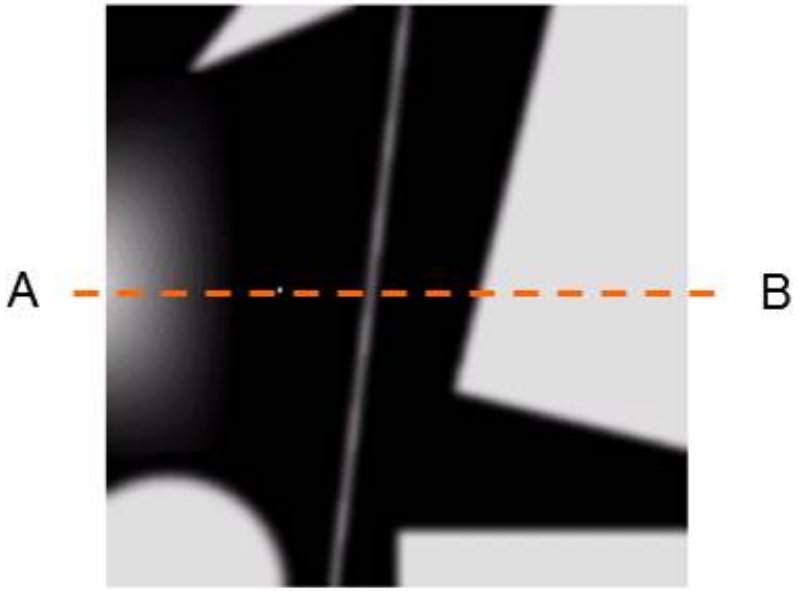
Spatial Differentiation

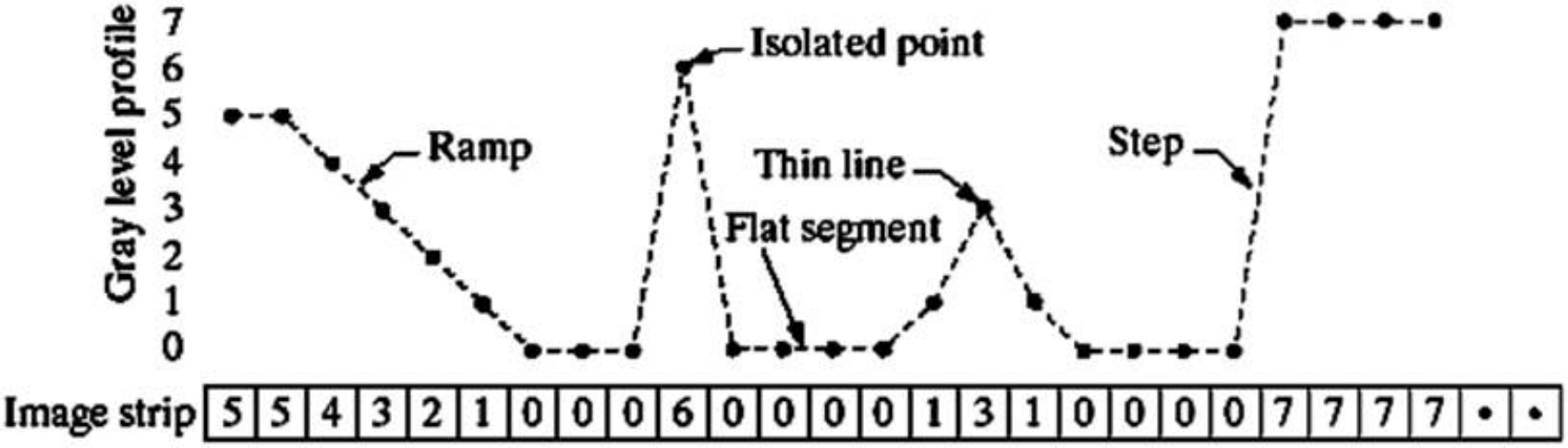
Differentiation measures the *rate of change* of a function

Let's consider a simple 1 dimensional example



Spatial Differentiation





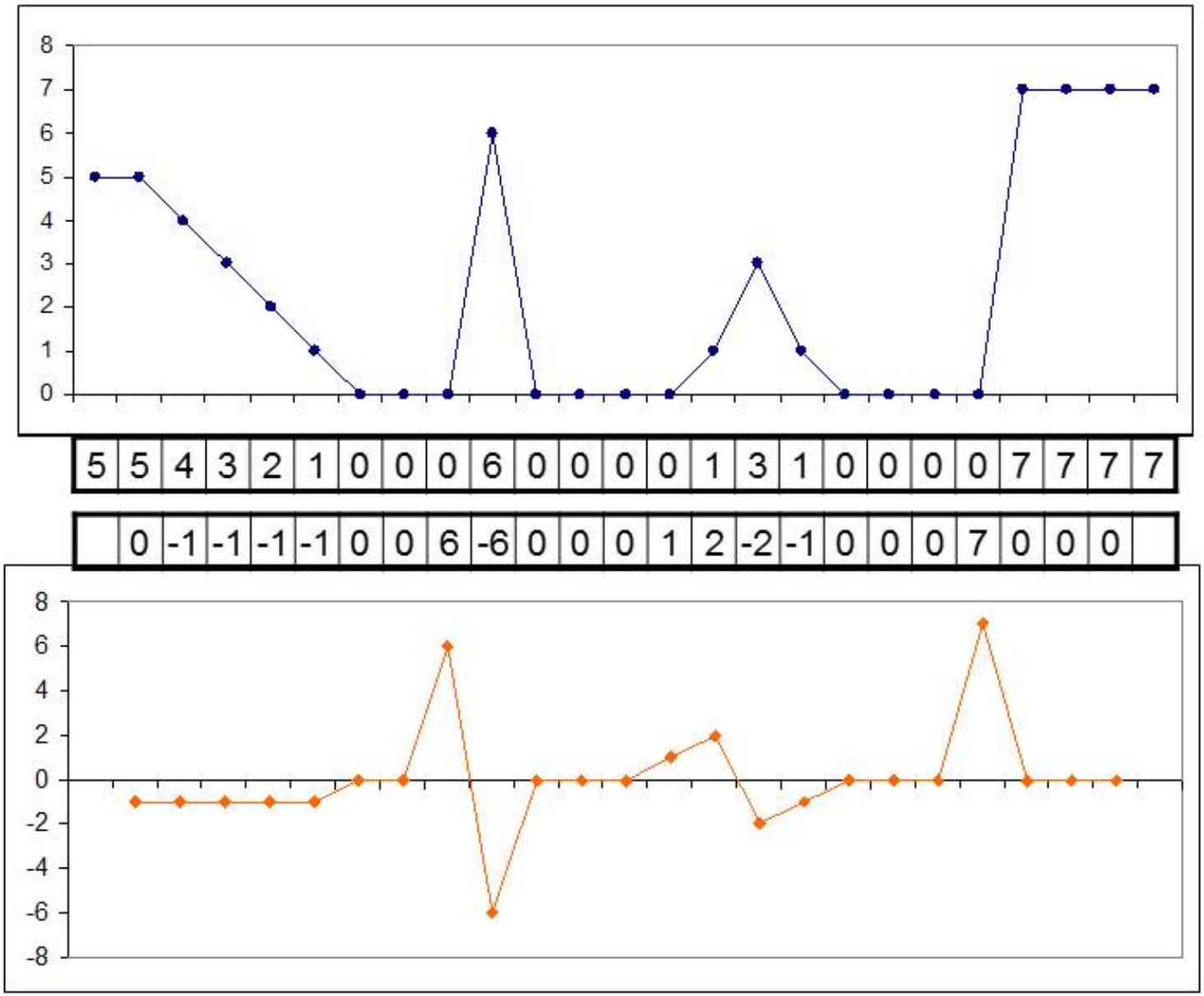
1st Derivative

The formula for the 1st derivative of a function is as follows:

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

It's just the difference between subsequent values and measures the rate of change of the function

1st Derivative (cont…)



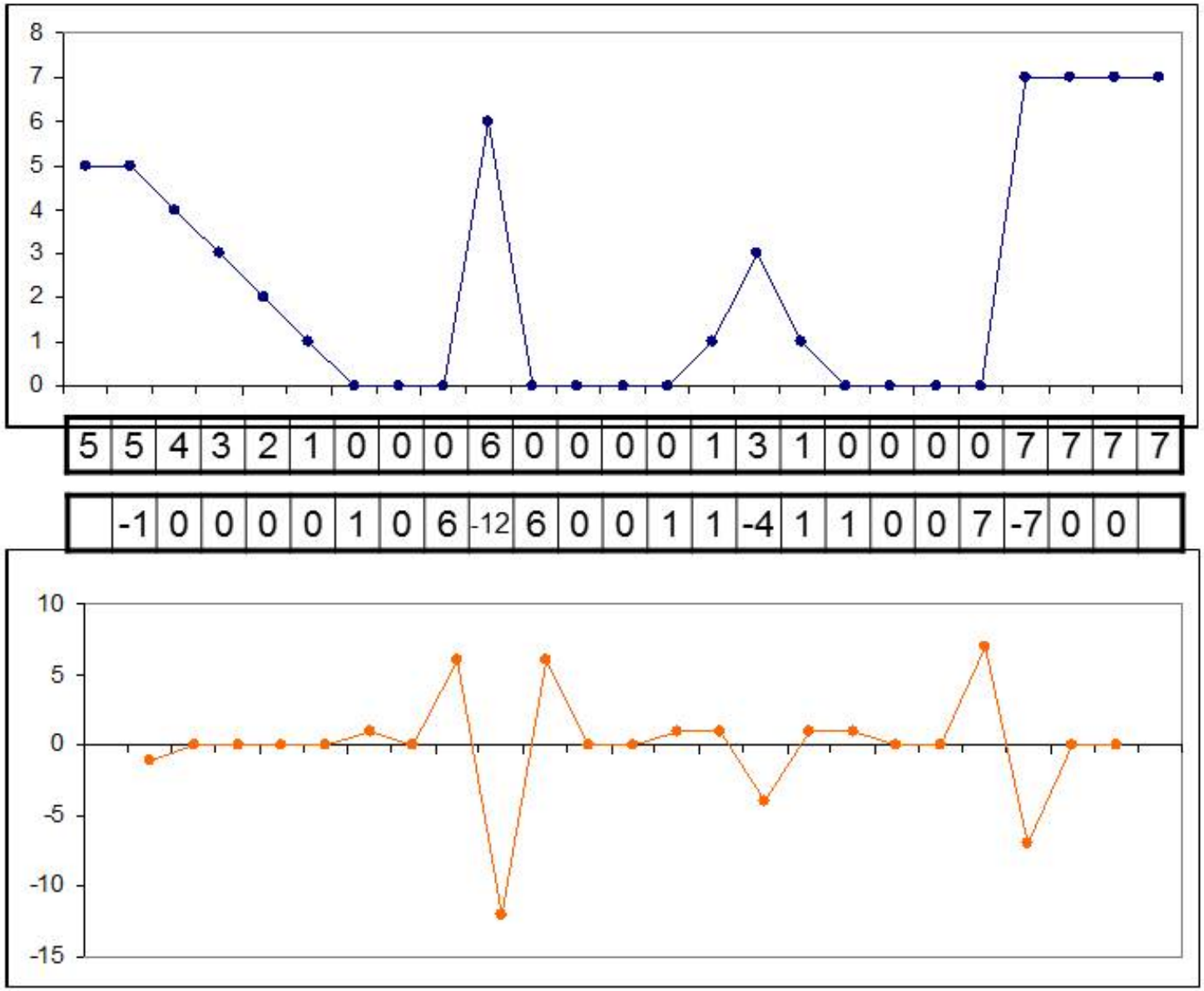
2nd Derivative

The formula for the 2nd derivative of a function is as follows:

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

Simply takes into account the values both before and after the current value

2nd Derivative (cont…)



Using Second Derivatives For Image Enhancement

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

- Stronger response to fine detail
- Simpler implementation
- We will come back to the 1st order derivative later on

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

- Isotropic
- One of the simplest sharpening filters
- We will look at a digital implementation

The Laplacian

The Laplacian is defined as follows:

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

where the partial 1st order derivative in the x direction is defined as follows:

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

and in the y direction as follows:

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

The Laplacian (cont…)

So, the Laplacian can be given as follows:

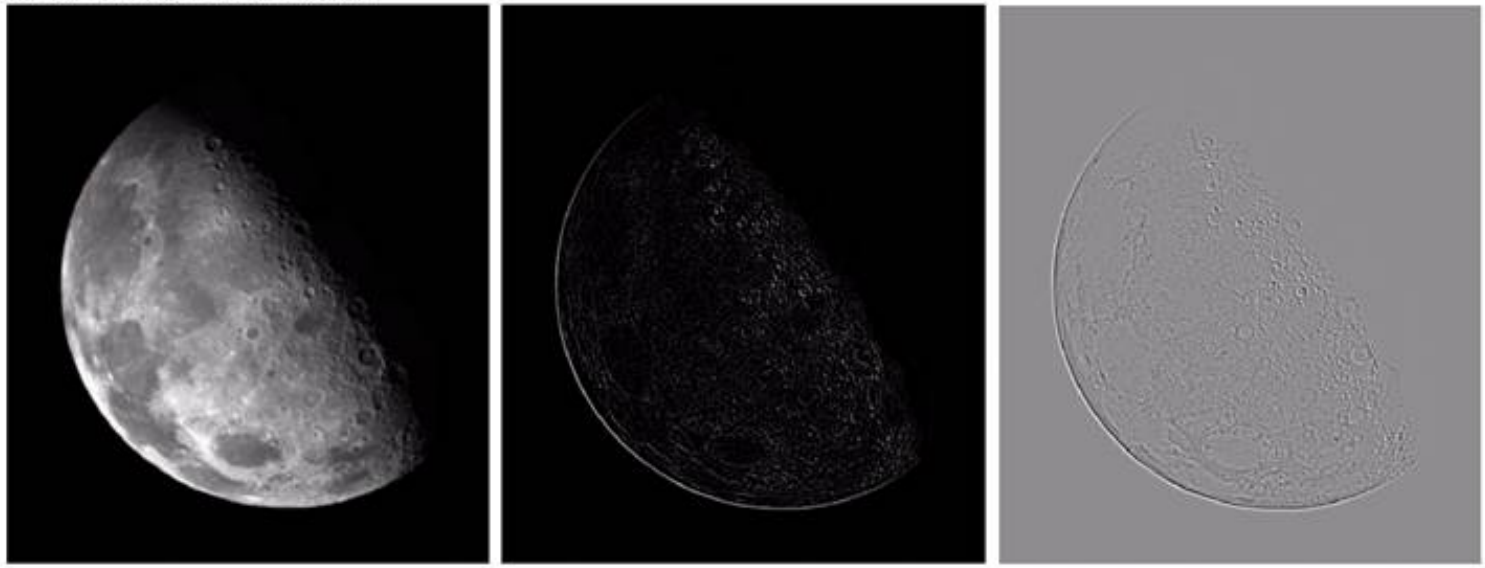
$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x, y)$$

We can easily build a filter based on this

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

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

But That Is Not Very Enhanced!

The result of a Laplacian filtering is not an enhanced image

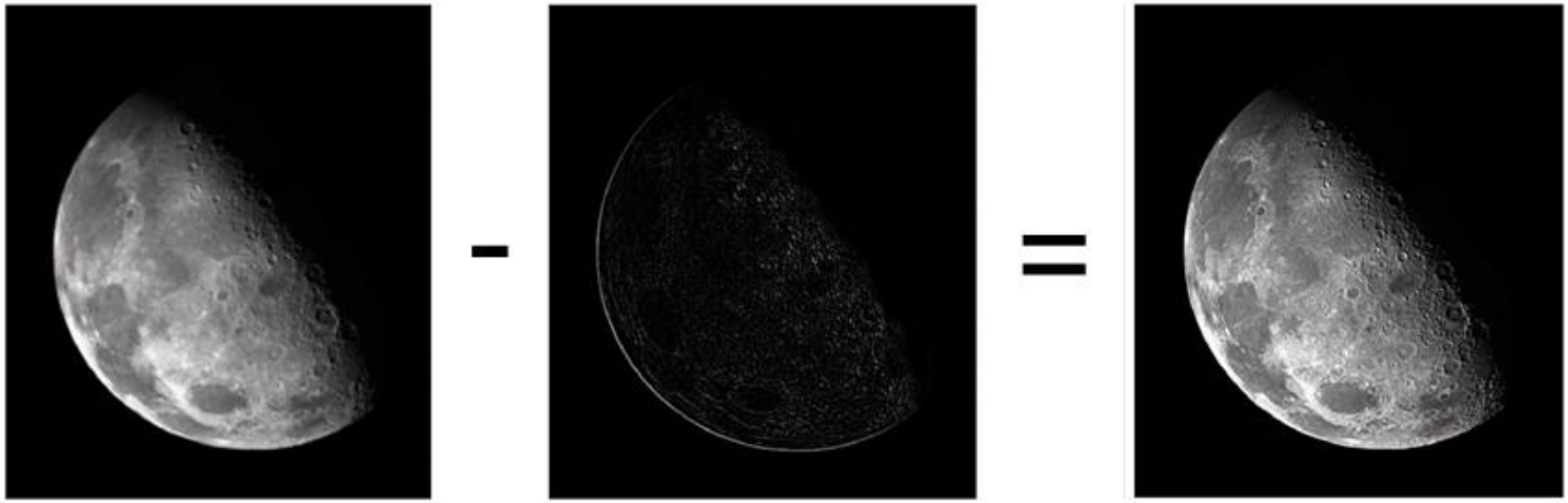
We have to do more work in order to get our final image

Subtract the Laplacian result from the original image to generate our final sharpened enhanced image

Laplacian
Filtered Image
Scaled for Display

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

Laplacian Image Enhancement



Original Image

Laplacian Filtered Image

Sharpened Image

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

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

Laplacian Image Enhancement

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

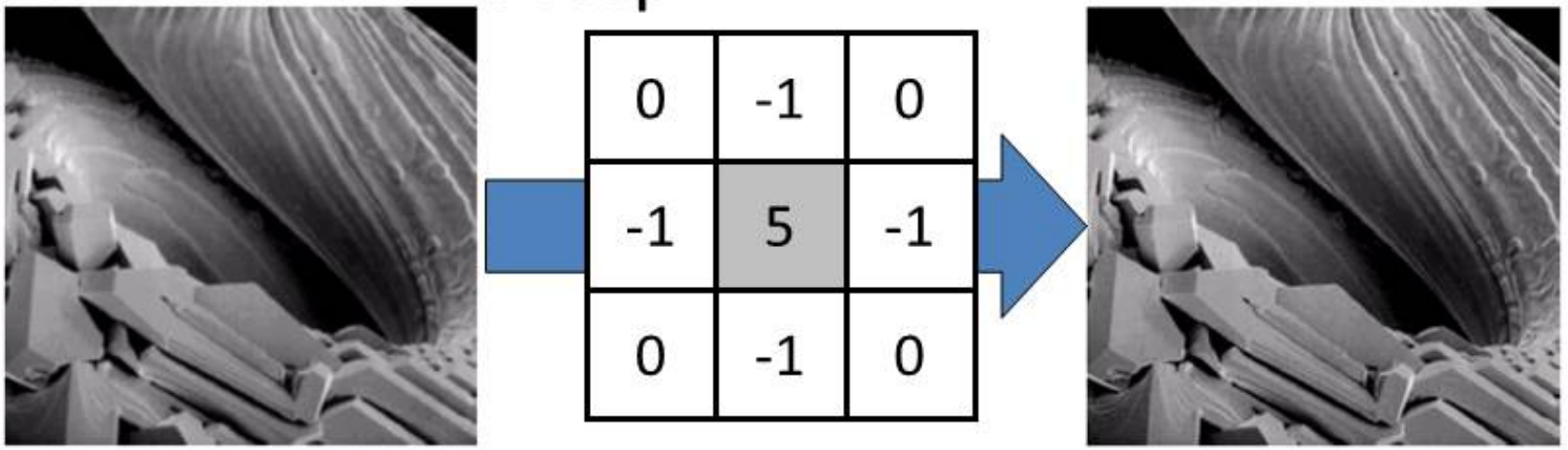
Simplified Image Enhancement

The entire enhancement can be combined into a single filtering operation

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

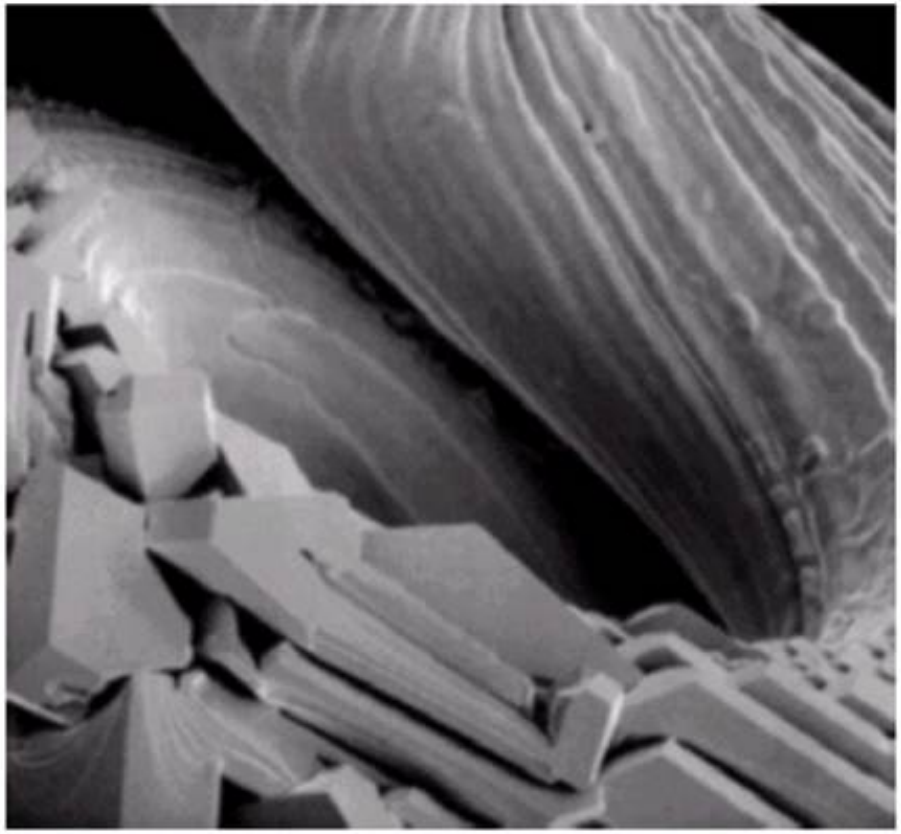
Simplified Image Enhancement (cont…)

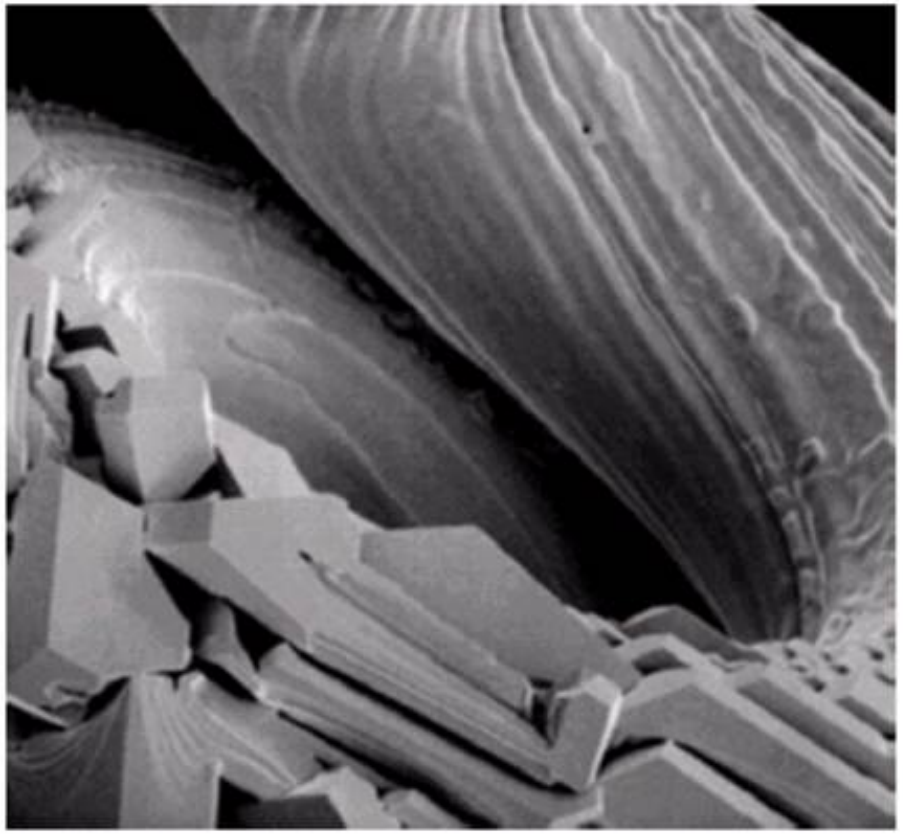
This gives us a new filter which does the whole job for us in one step



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

Simplified Image Enhancement (cont…)





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

Variants On The Simple Laplacian

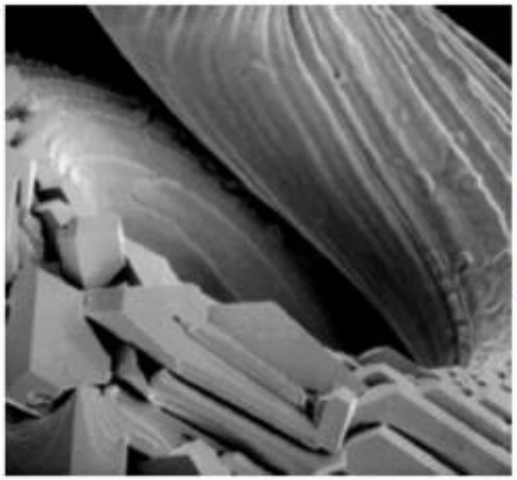
There are lots of slightly different versions of the Laplacian that can be used:

0	1	0
1	-4	1
0	1	0

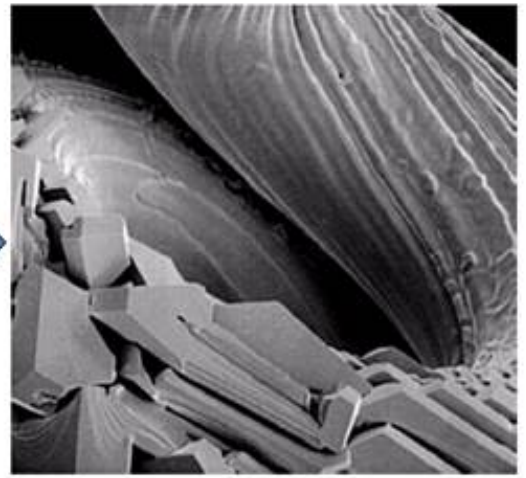
Simple Laplacian

1	1	1
1	-8	1
1	1	1

Variant of Laplacian



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

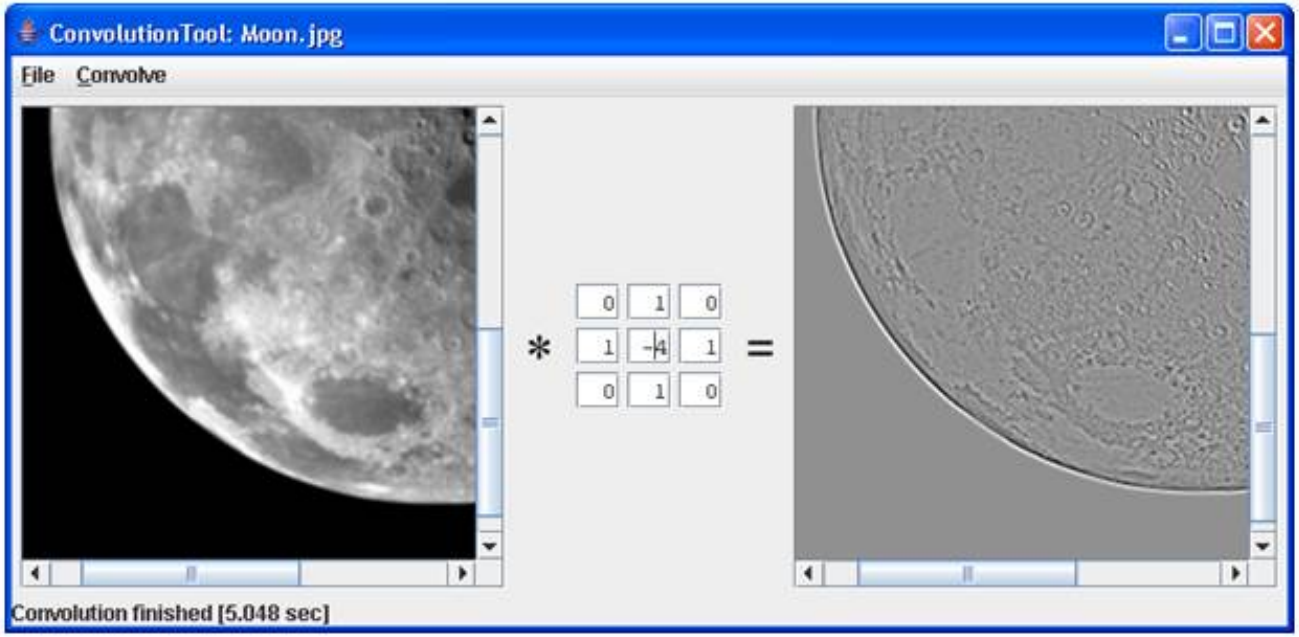


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Simple Convolution Tool In Java

A great tool for testing out different filters

- From the book "Image Processing tools in Java"
- Available from webCT later on today
- To launch: `java ConvolutionTool Moon.jpg`



1st Derivative Filtering

Implementing 1st derivative filters is difficult in practice

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

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

1st Derivative Filtering (cont…)

The magnitude of this vector is given by:

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

For practical reasons this can be simplified as:

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

1st Derivative Filtering (cont…)

There is some debate as to how best to calculate these gradients but we will use:

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

which is based on these coordinates

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Sobel Operators

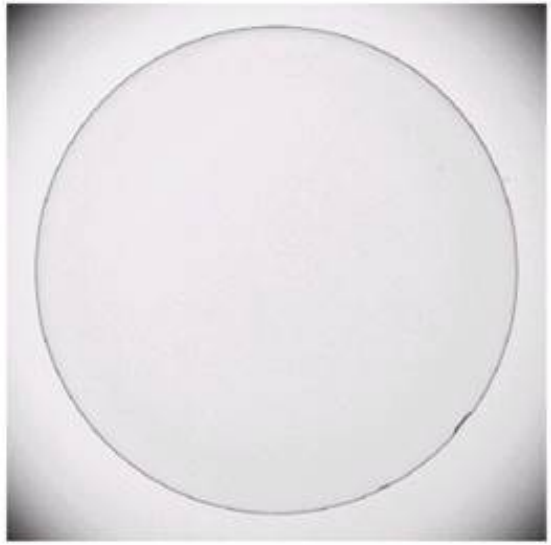
Based on the previous equations we can derive the *Sobel Operators*

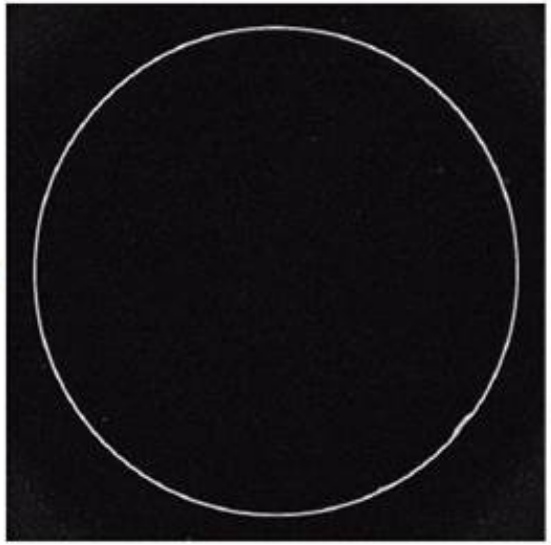
-1	-2	-1
0	0	0
1	2	1

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

To filter an image it is filtered using both operators the results of which are added together

Sobel 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 filters are typically used for edge detection

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

1^{st} & 2^{nd} Derivatives

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

- 1^{st} order derivatives generally produce thicker edges
- 2^{nd} order derivatives have a stronger response to fine detail e.g. thin lines
- 1^{st} order derivatives have stronger response to grey level step
- 2^{nd} order derivatives produce a double response at step changes in grey level

Summary

In this lecture we looked at:

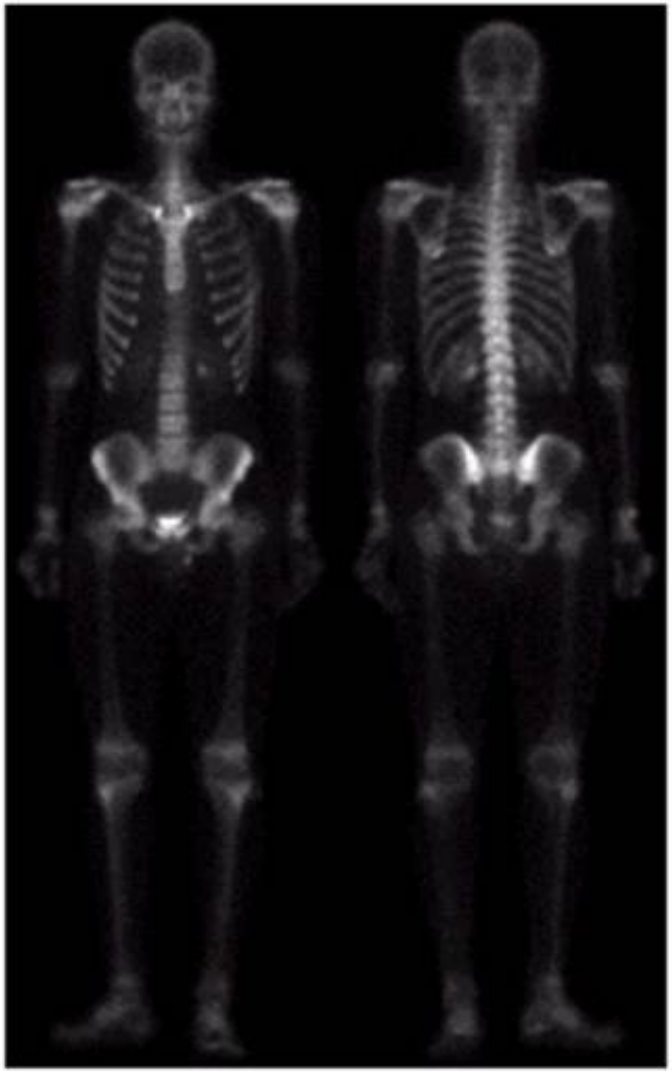
- Sharpening filters
  - 1st derivative filters
  - 2nd derivative filters
- Combining filtering techniques

Combining Spatial Enhancement Methods

Successful image enhancement is typically not achieved using a single operation

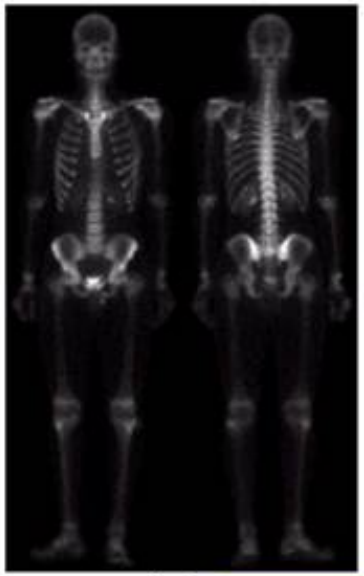
Rather we combine a range of techniques in order to achieve a final result

This example will focus on enhancing the bone scan to the right



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

Combining Spatial Enhancement Methods (cont…)



(a)

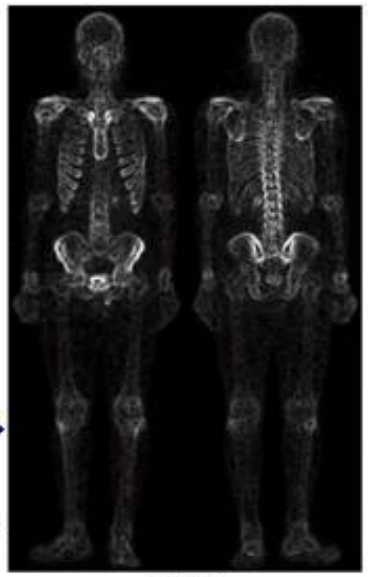
Laplacian filter of bone scan (a)

(b)

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

(c)

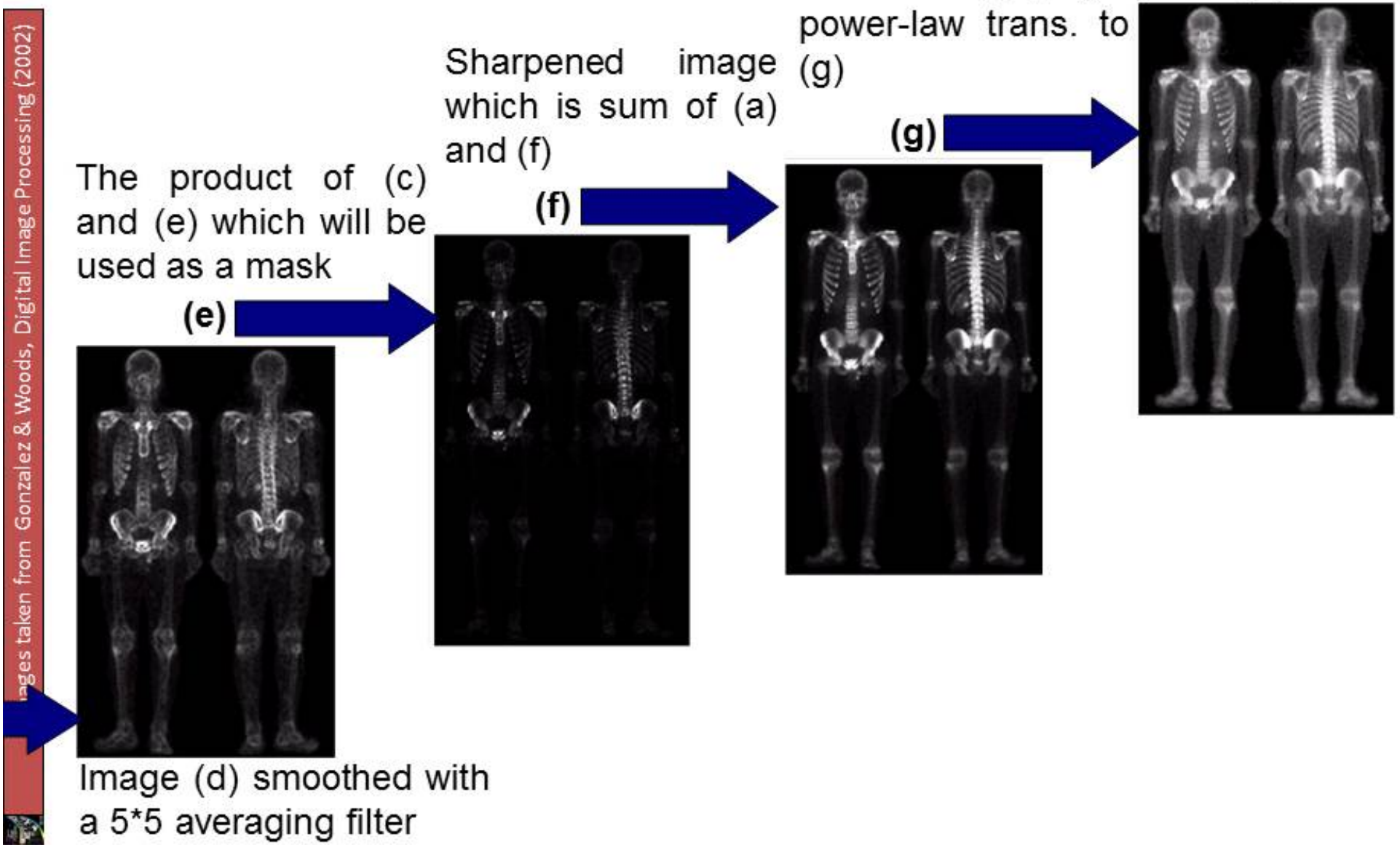
Sobel filter of bone scan (a)



(d)

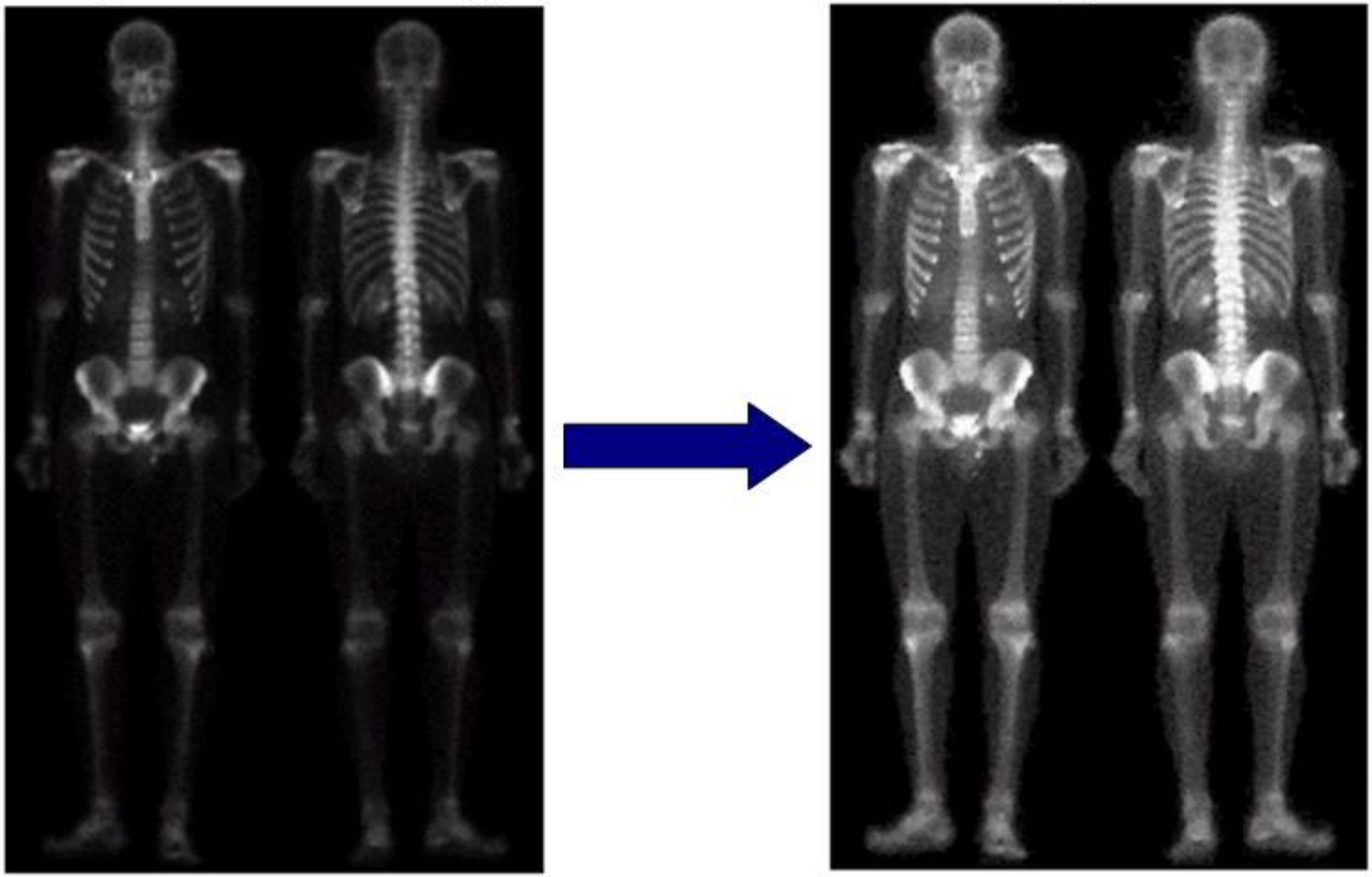
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Combining Spatial Enhancement Methods (cont...)



Combining Spatial Enhancement Methods (cont…)

Compare the original and final images



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