Image Enhancement

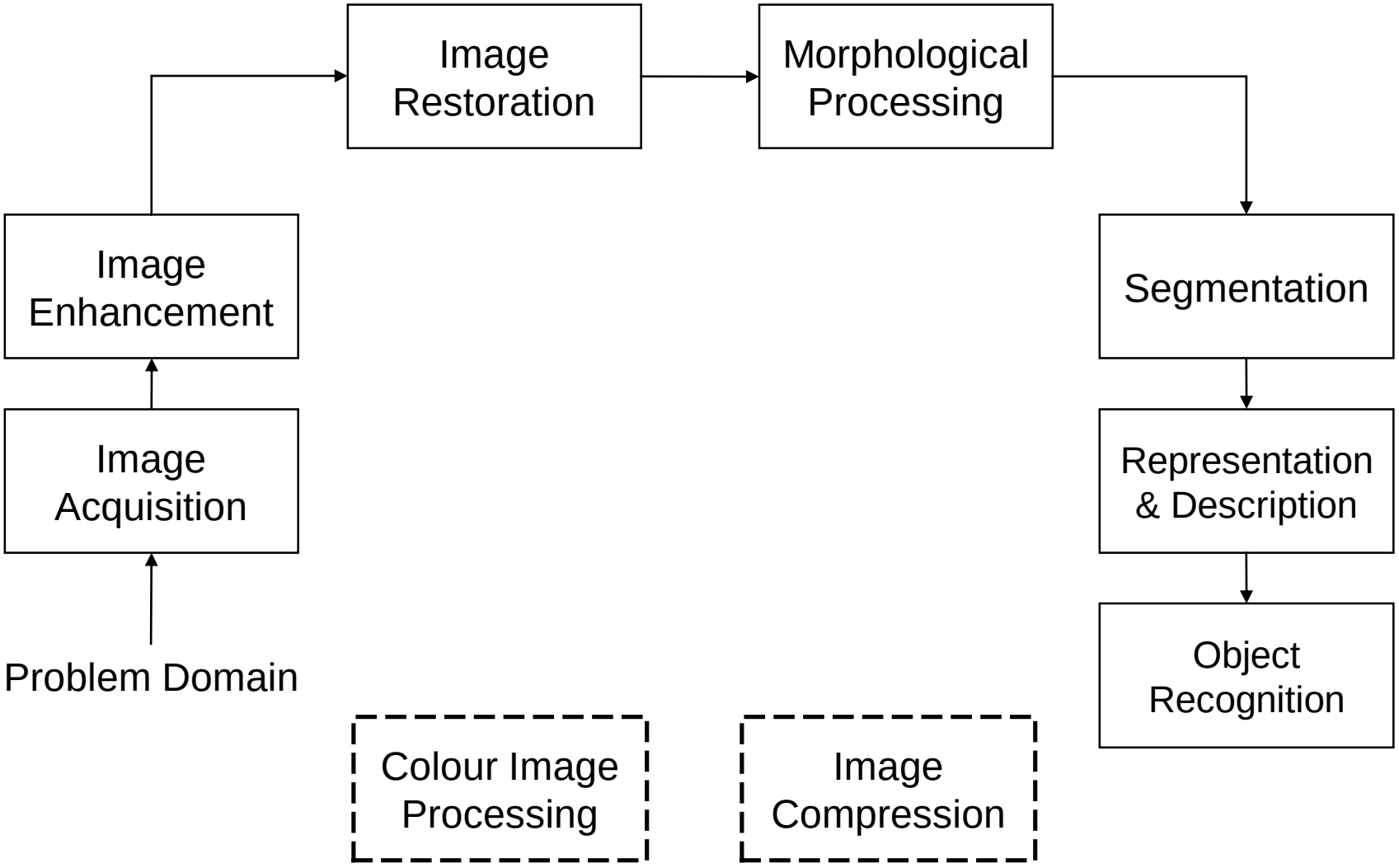
In Spatial Domain

Partially Adopted from Brian Mac Namee

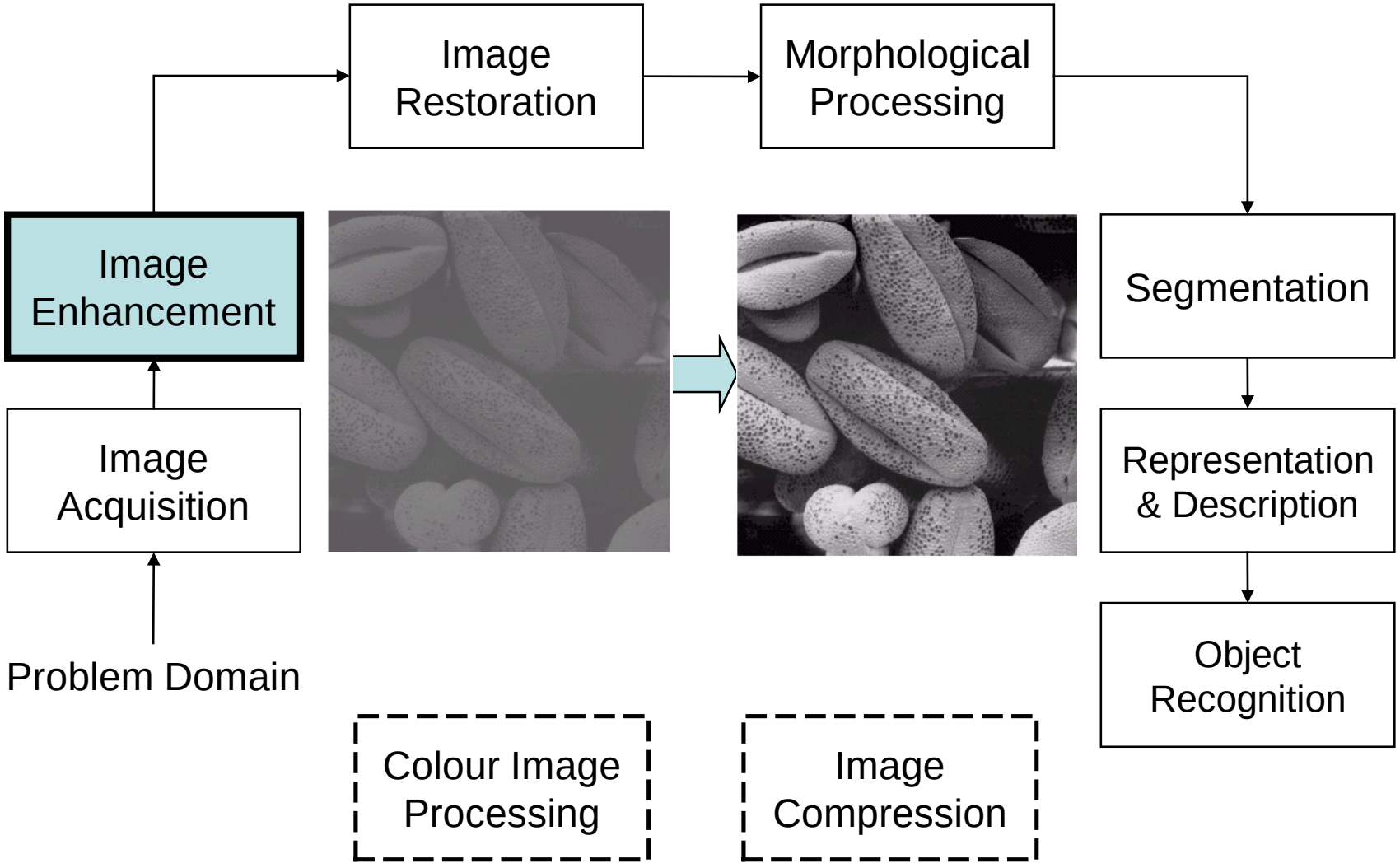
Contents

- ✓ What is Image Enhancement?
- ✓ Different Kinds of Image Enhancement
- ✓ Contrast Intensification
- ✓ Noise Cleaning or Smoothing
- ✓ Edge Sharpening

Phases of Digital Image Processing



Phases of Digital Image Processing: Image Enhancement



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

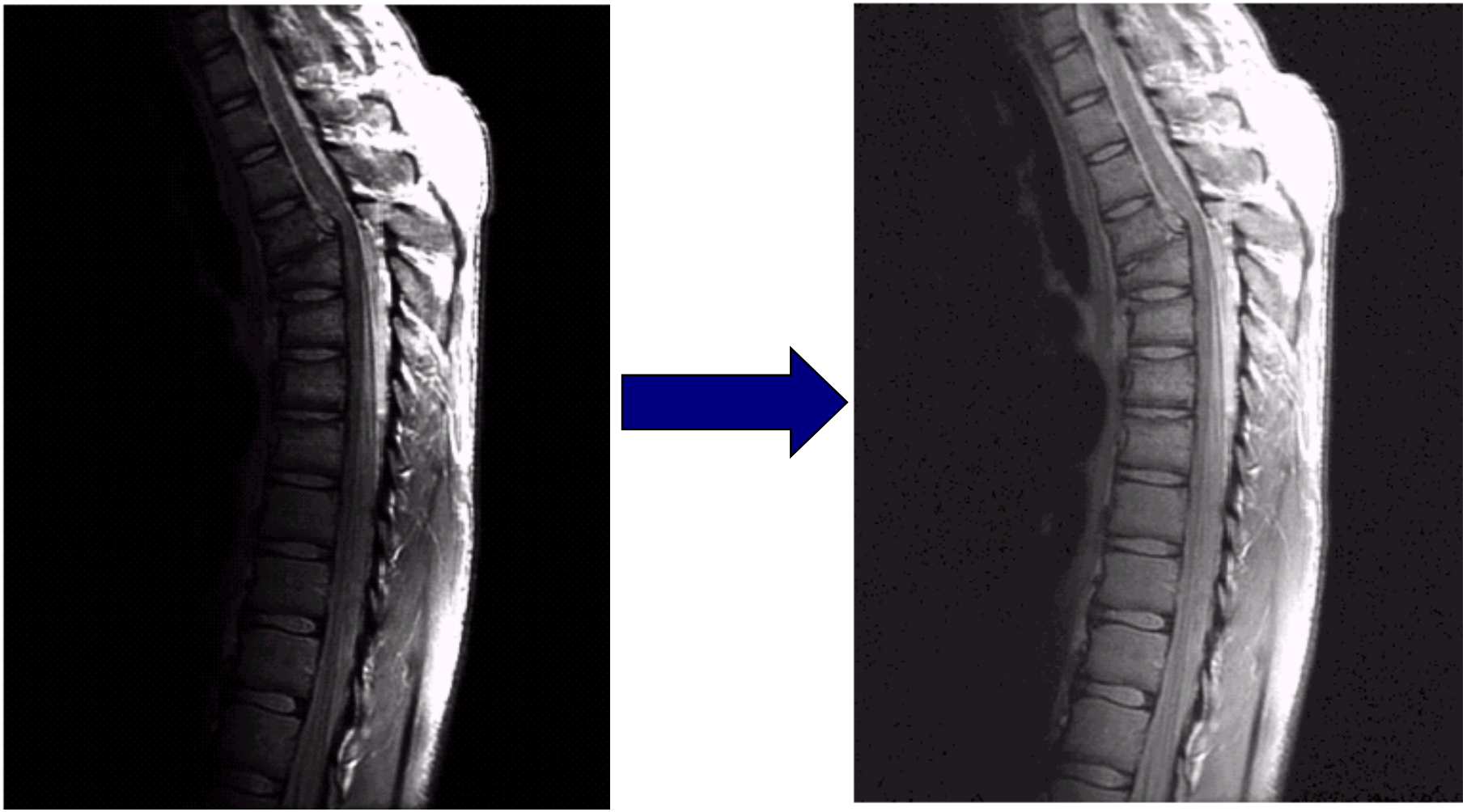
What Is Image Enhancement?

Image enhancement is the process of making images more useful.

The reasons for doing this include:

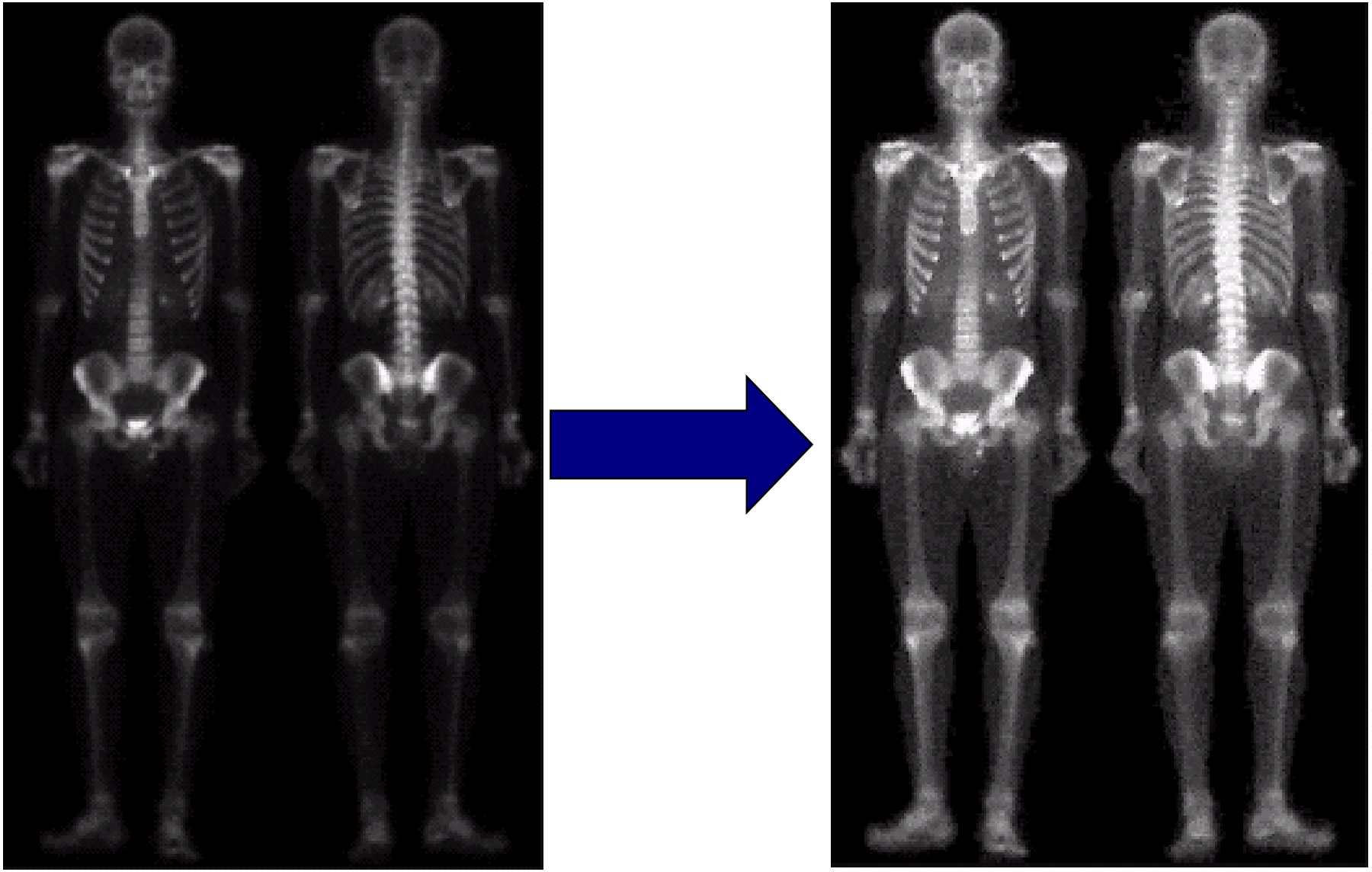
- Highlighting interesting detail in images.
- Removing noise from images.
- Making images more visually appealing.

Image Enhancement Examples



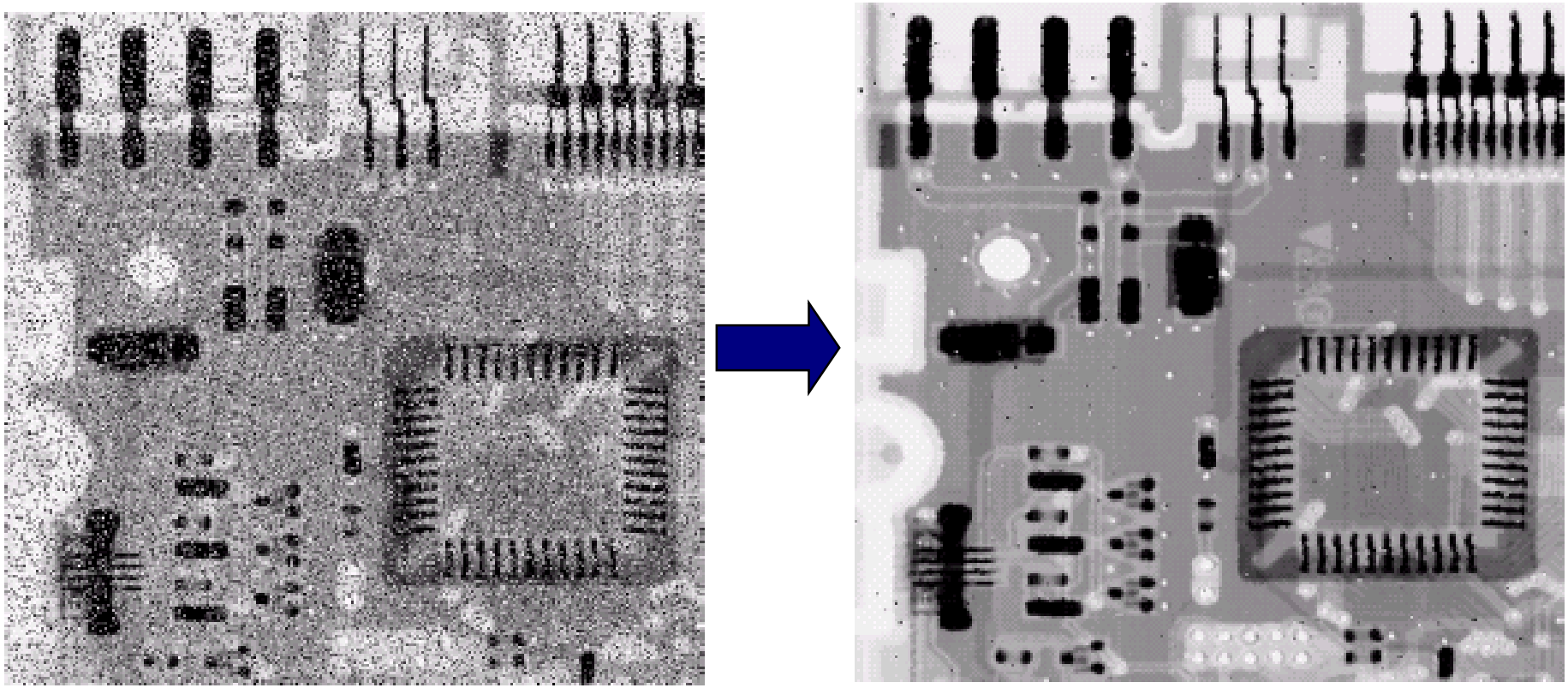
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Image Enhancement Examples



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

Image Enhancement Examples



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

Image Enhancement Examples

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

Spatial and Frequency Domains

There are two broad categories of image enhancement techniques:

Spatial domain techniques

Direct manipulation of image pixels

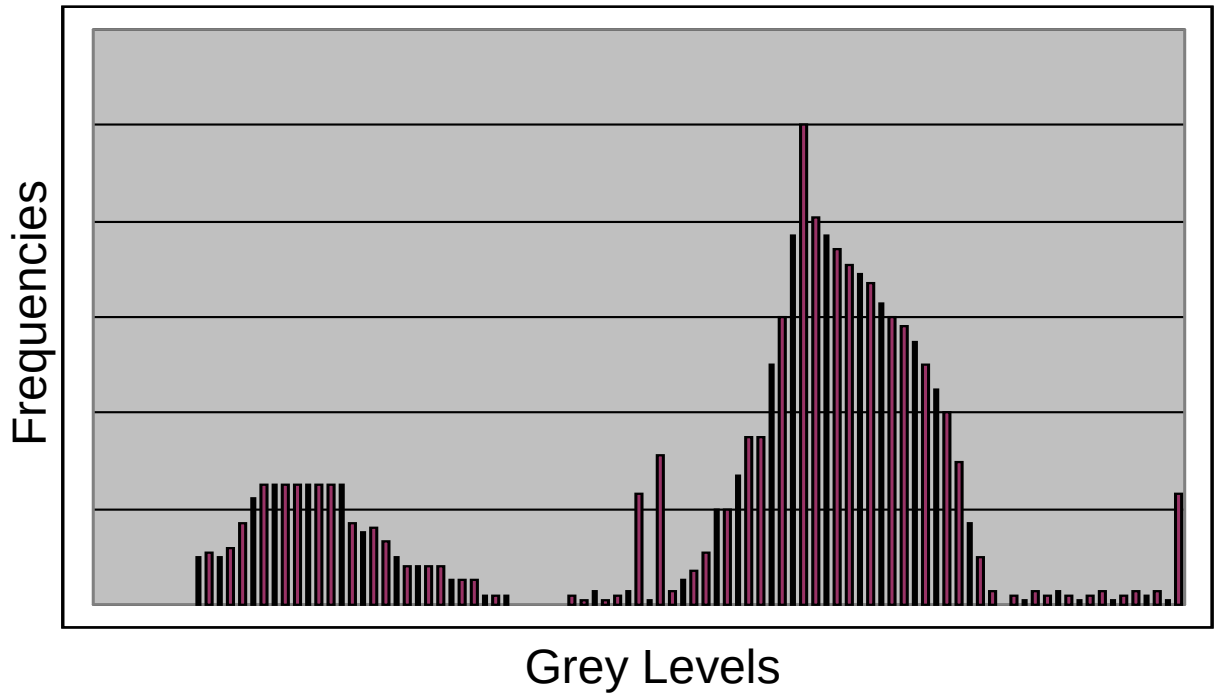
Frequency domain techniques

Manipulation of Fourier transform or wavelet transform of an image.

Image Histograms

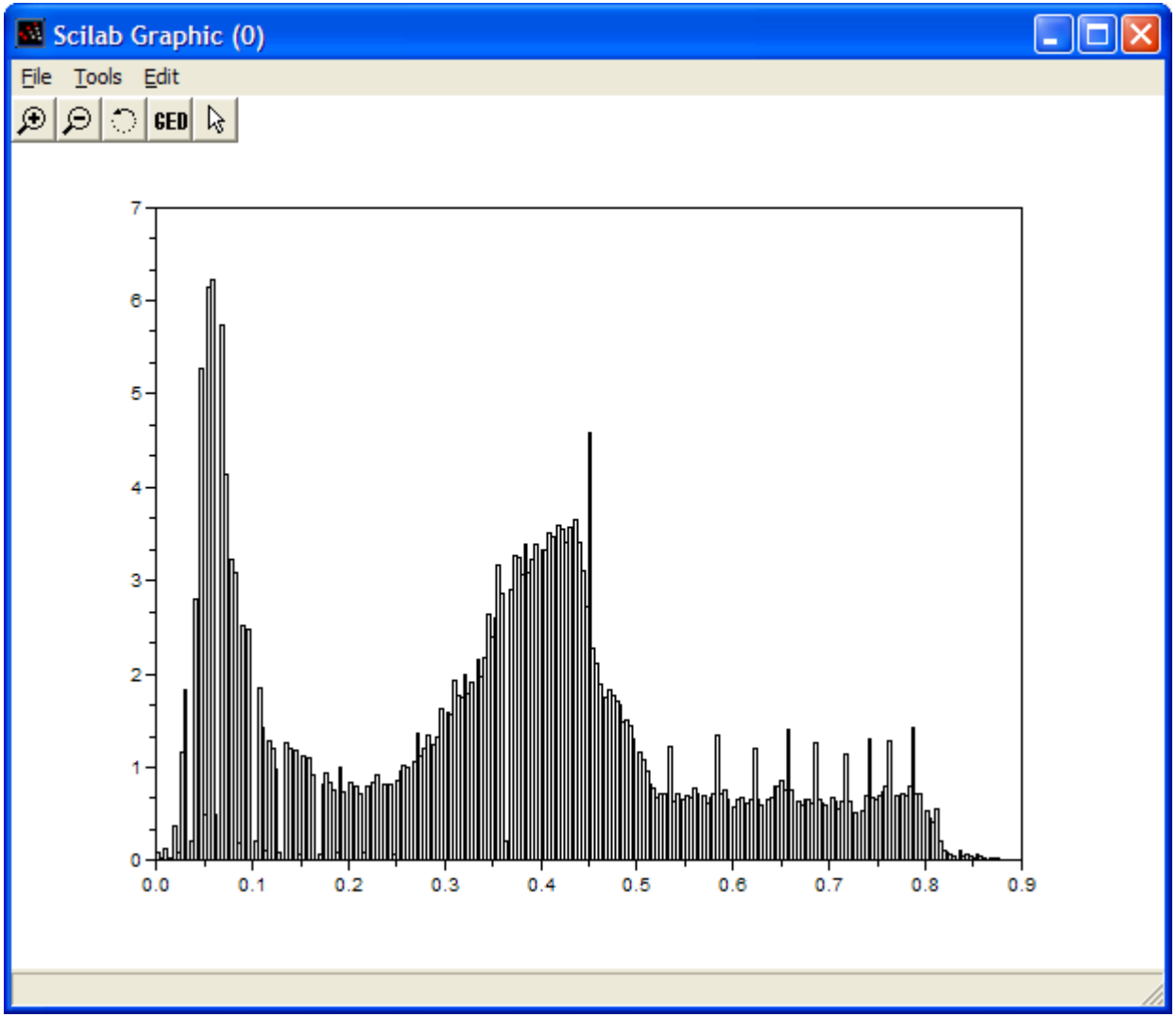
The image histogram shows the distribution of grey levels in the image.

Useful in image processing, particularly in image segmentation.



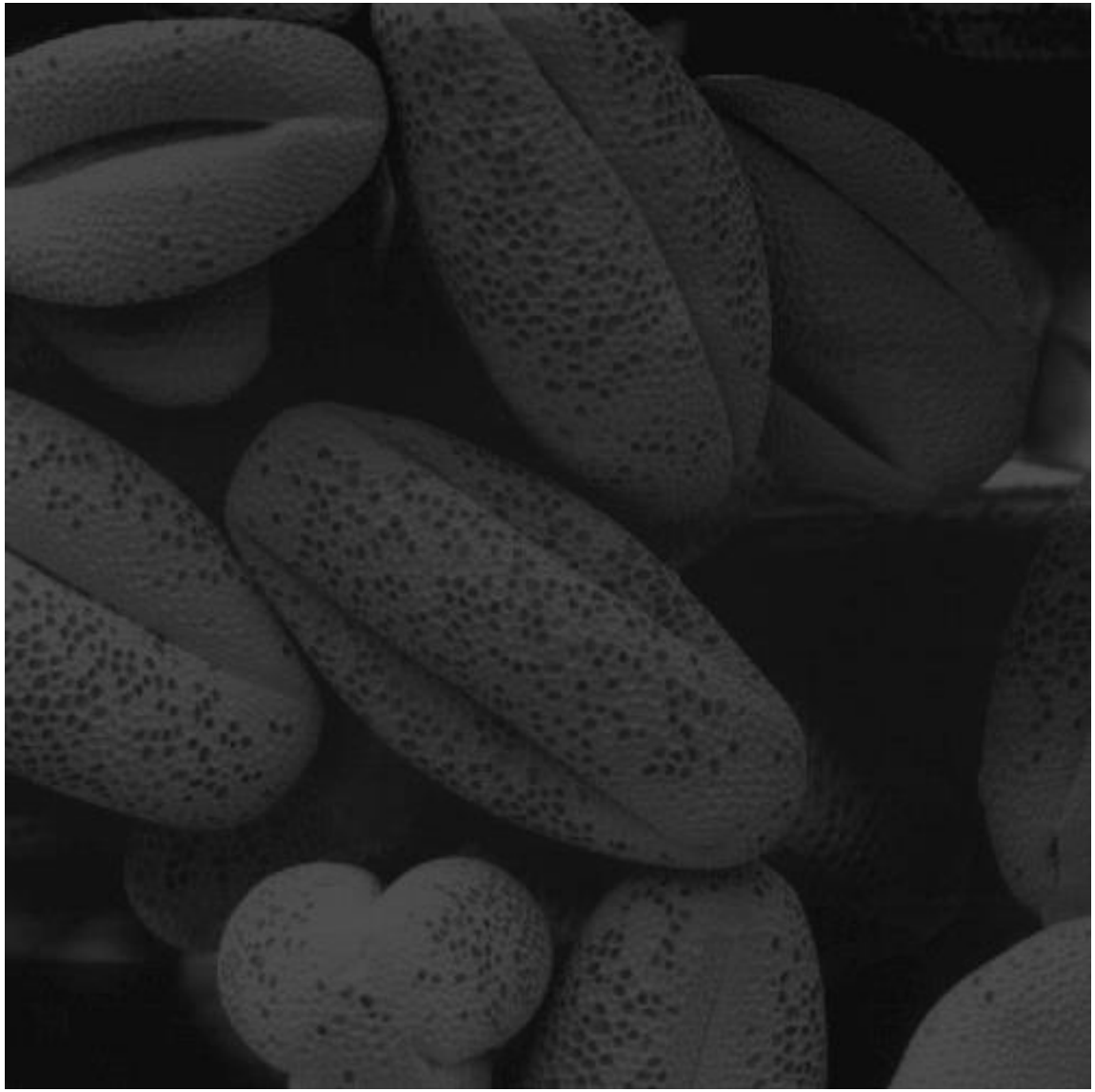
Histogram Examples

Histogram Examples



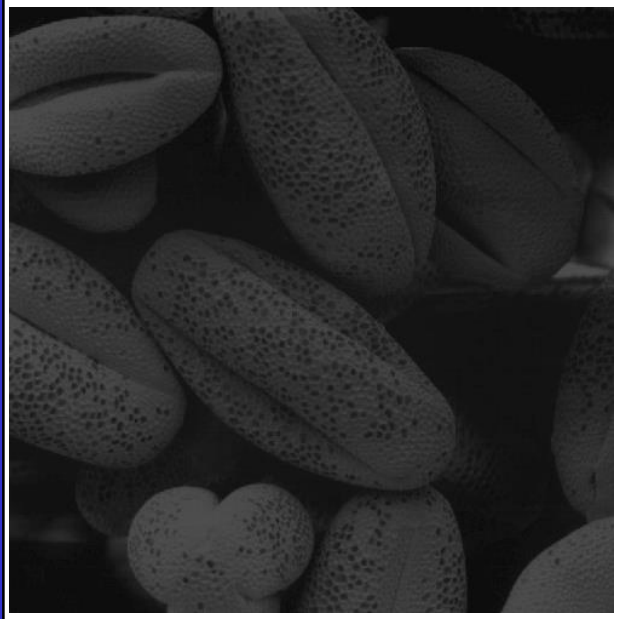
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

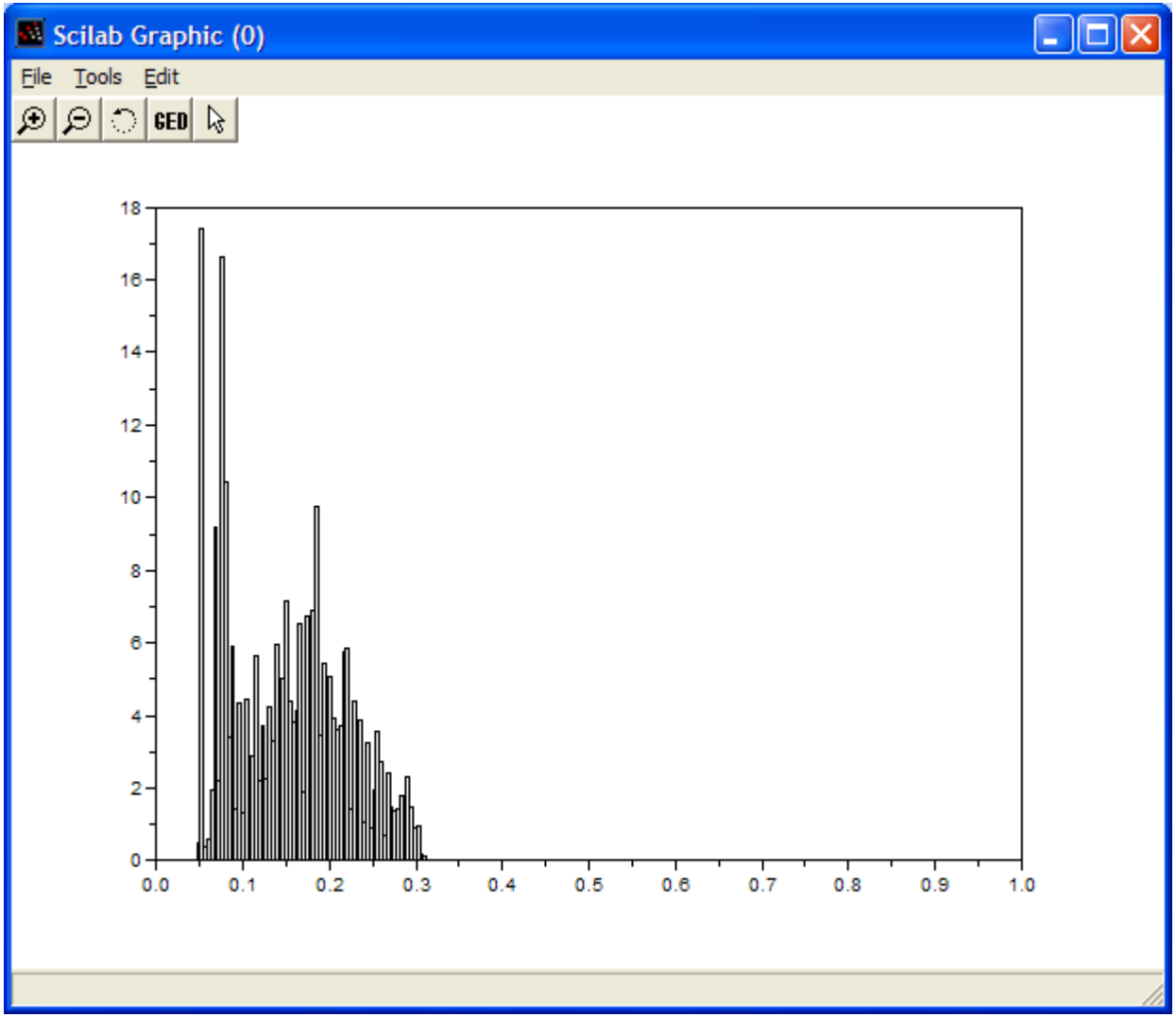
Histogram Examples



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

Histogram Examples

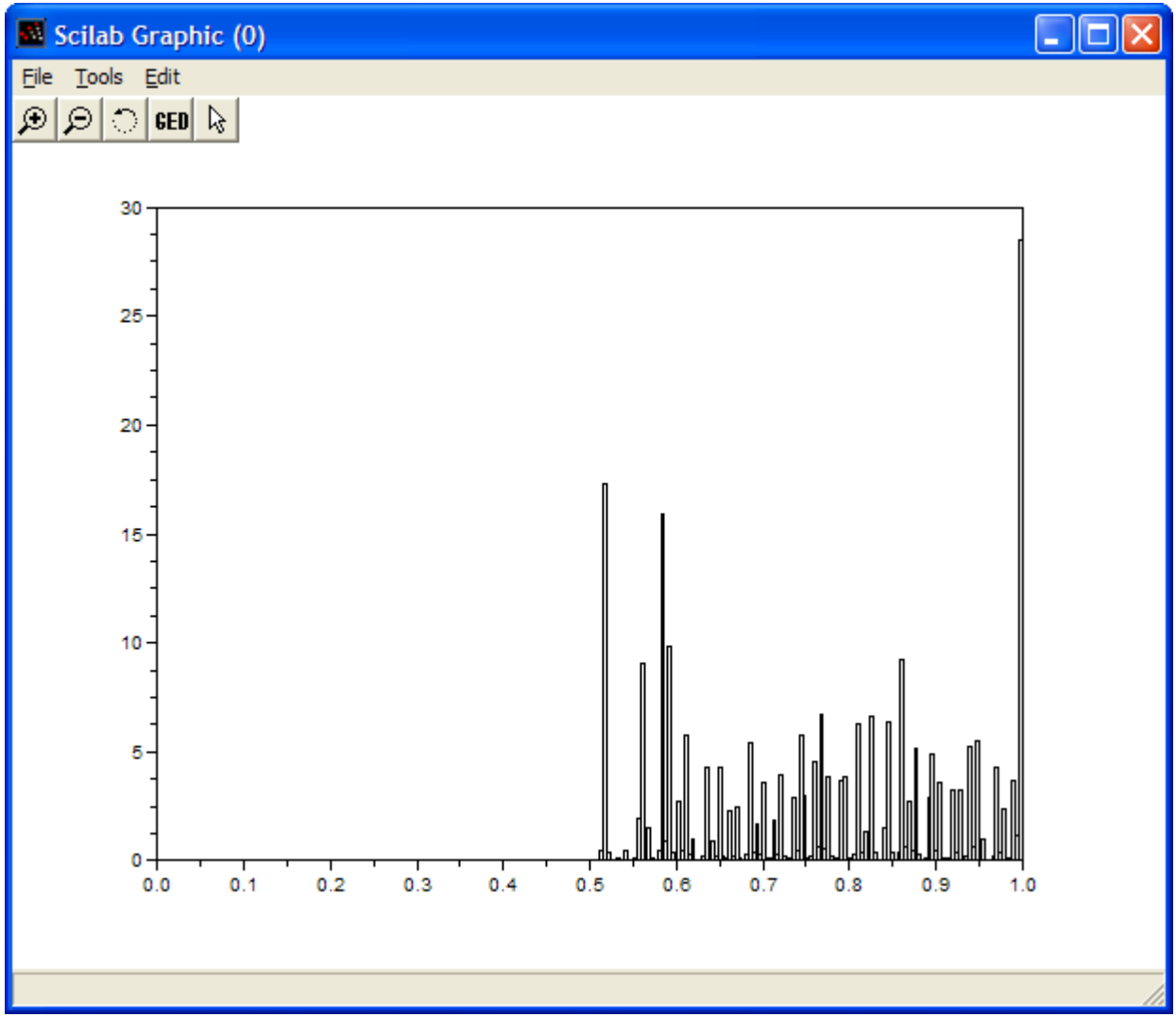




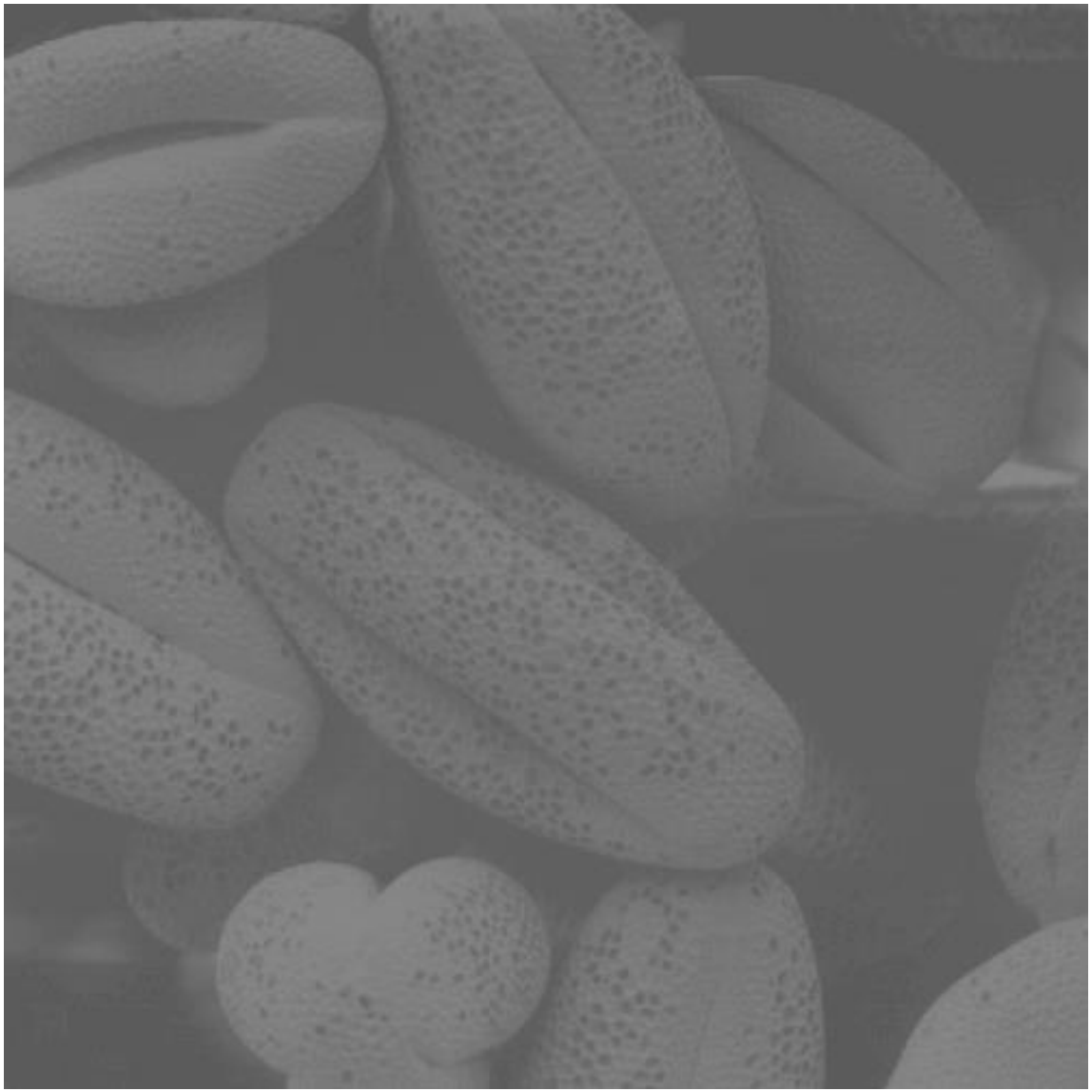
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Histogram Examples

Histogram Examples

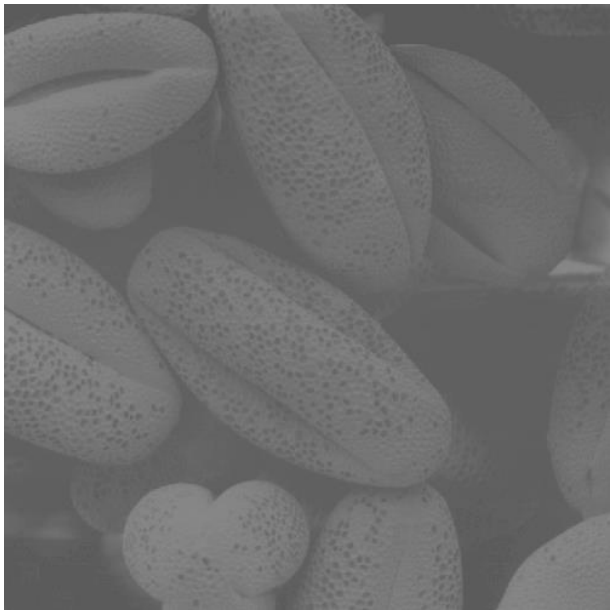


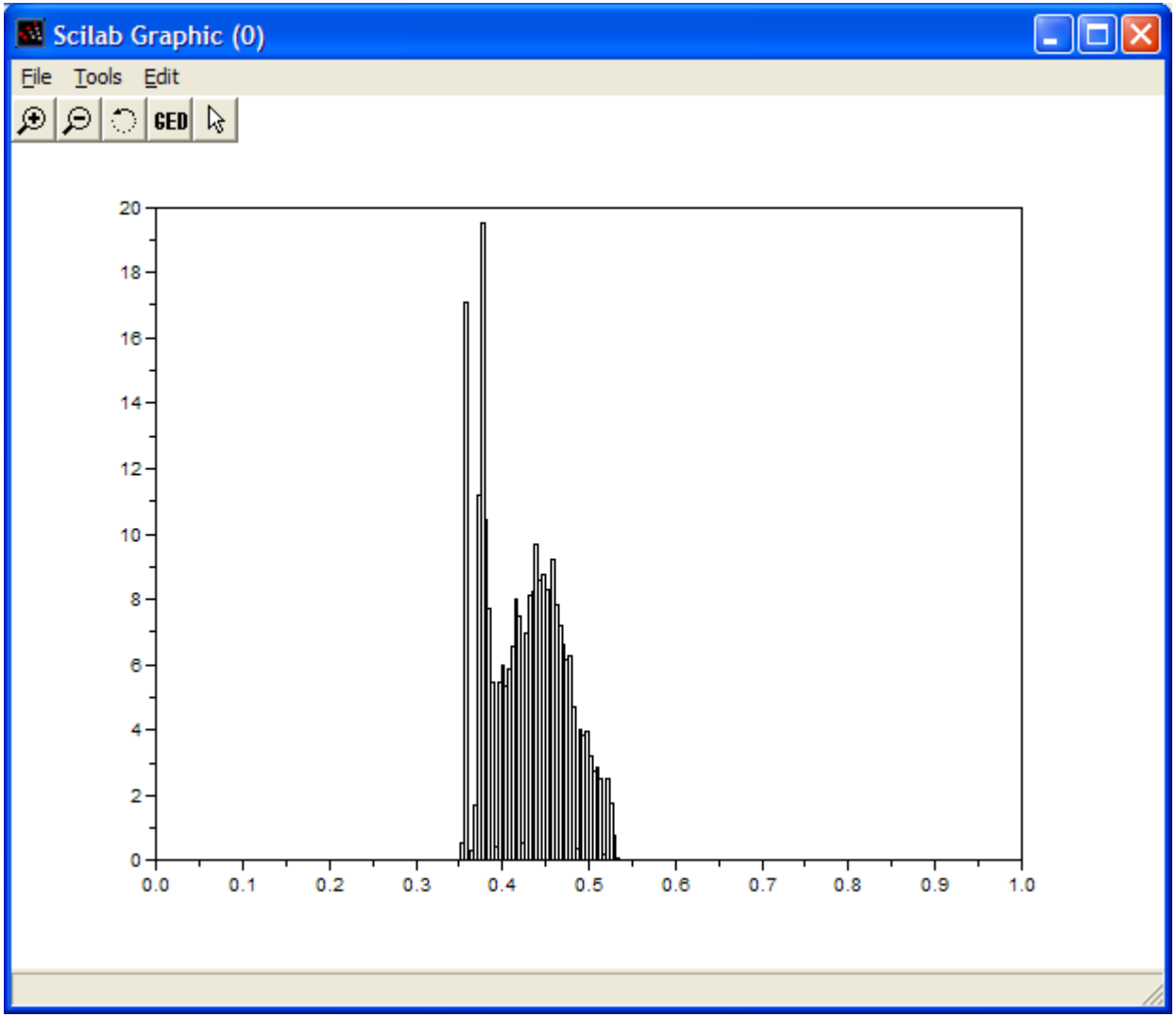
Histogram Examples



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

Histogram Examples

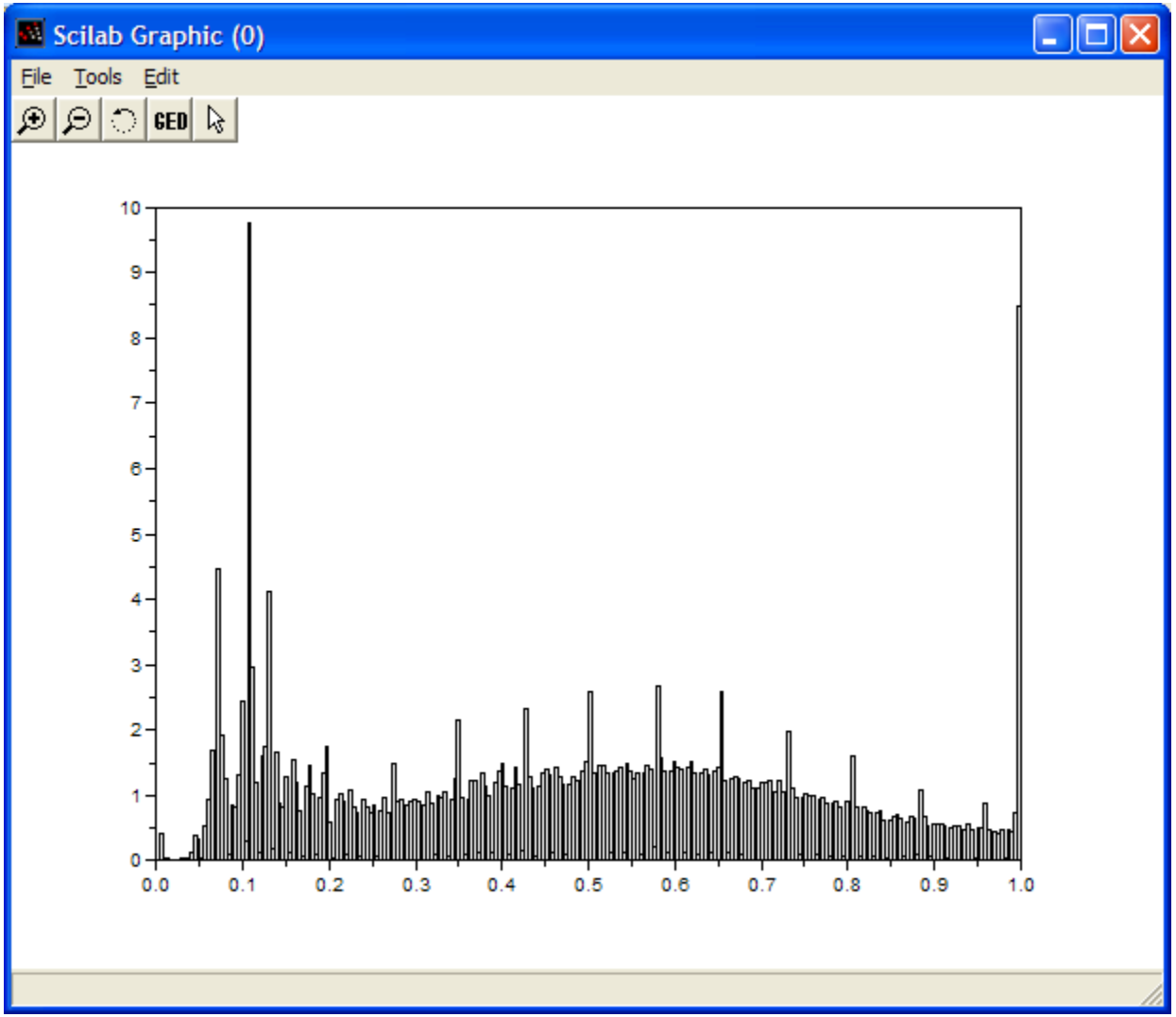




Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Histogram Examples

Histogram Examples



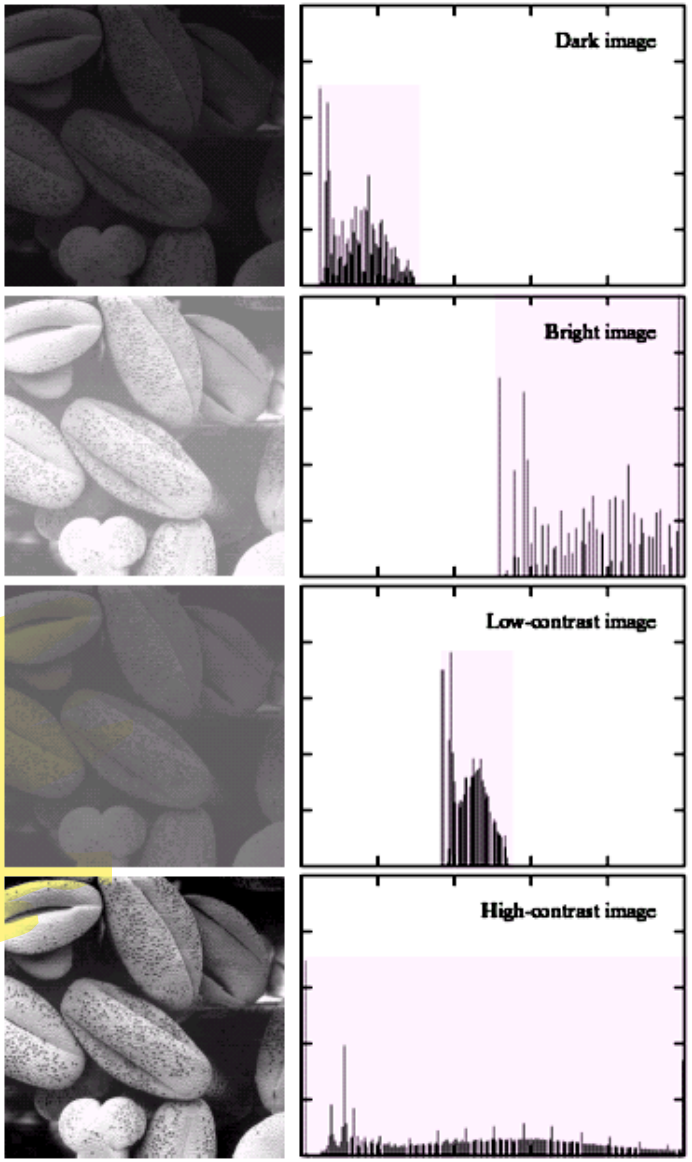
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Histogram Examples

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.



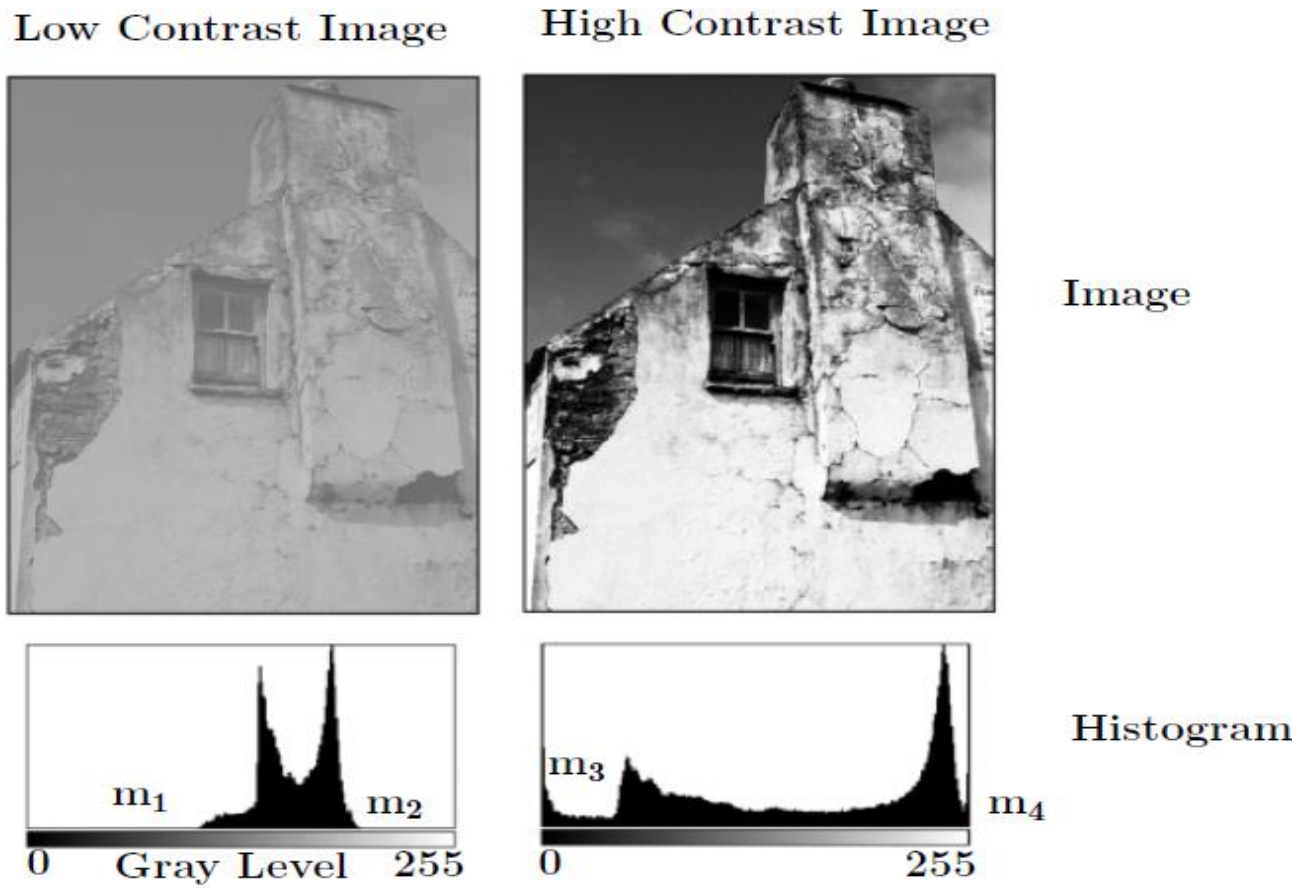
Contrast Stretching

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?

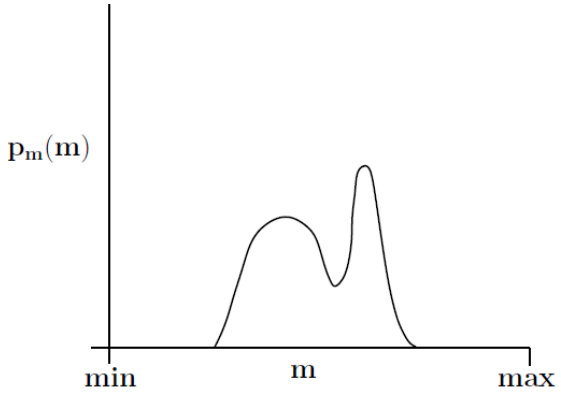
- Linear Stretching
- Non-linear Stretching
- Histogram Equalization

Contrast Stretching

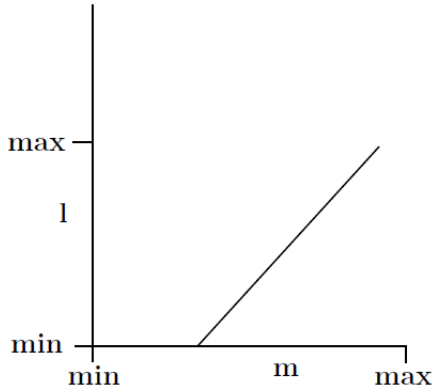


$$[m_1, m_2] \subset [m_3, m_4]$$

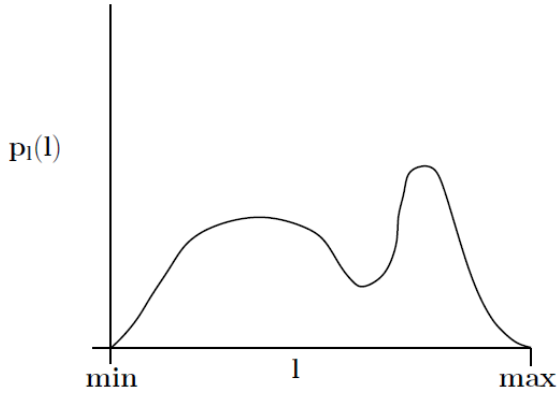
Contrast Stretching



Gray level histogram of input image



Transformation function



Gray level histogram of processed image

Property of Transformation Function

$$l = \mathrm{T}(m)$$

- $\mathrm{T}(m)$ must be monotonically increasing in the interval $[m_{min}, m_{max}]$,
  $m_1 \leq m_2 \Rightarrow l_1 = \mathrm{T}(m_1) \leq l_2 = \mathrm{T}(m_2)$
- $l_{min} \leq l \leq l_{max}$, i.e., the transformed gray level must lie within the allowed range of gray level.

Linear Stretching

$$l = \mathrm{T}(m) = \frac{l_{max} - l_{min}}{m_{max} - m_{min}}(m - m_{min}) + l_{min}$$

This transformation function shifts and stretches the gray level range (and gray level histogram) of the input image to occupy the entire dynamic range $[l_{min}, l_{max}]$.

$$\bar{\mathrm{g}}(r, c) = \frac{l_{max} - l_{min}}{m_{max} - m_{min}} \{\mathrm{g}(r, c) - m_{min}\} + l_{min}$$

Problem

Suppose m be the gray level of input image which is to be transformed to l by linear stretching, where l is the gray level of the output image. Let n_i and $n_{i'}$ are the number of pixels having i^{th} gray level in the input and the output images, respectively. Suppose for an 8-level image we have the following frequency table for the input gray levels.

i	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
n_i	0	0	a	b	c	d	e	0

, a to e > 0

Solution

$m_{min} = 2$, $m_{max} = 6$

As, we are considering 8-level image, $l_{min} = 0, l_{max} = 7$.

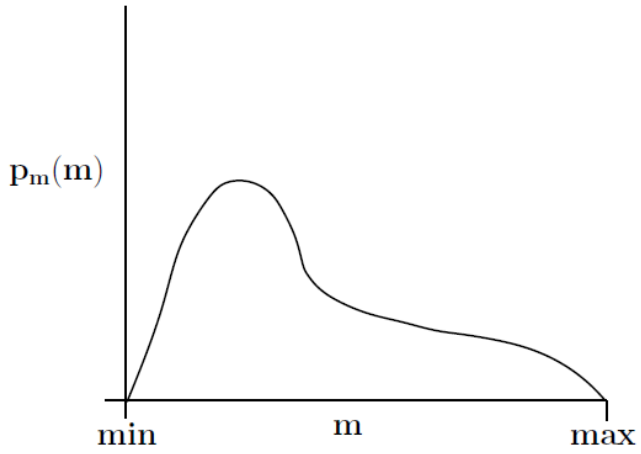
$$l = \frac{7-0}{6-2}(m-2) + 0 = \frac{7}{4}(m-2)$$

- For m = 3, we have $l = 1.75 \approx 2$
- For m = 4, we have $l = 3.5 \approx 4$

i	0	1	2	3	4	5	6	7
$n_{i'}$	a	0	b	0	c	d	0	e

Non-linear Stretching

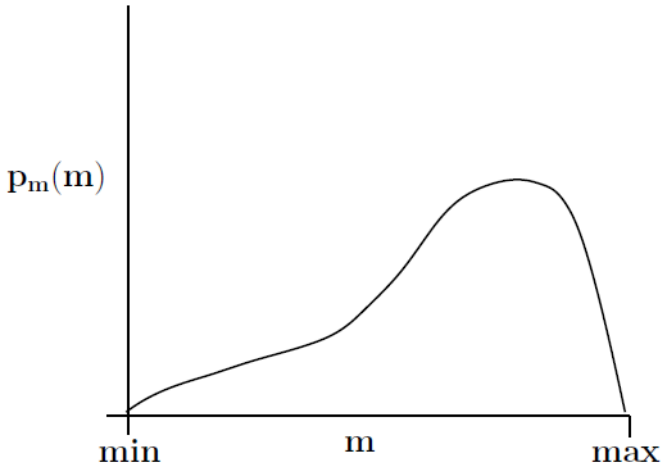
- When the graylevel histogram shows high population of pixels in the lower graylevel zone of the gray scale, we may want to stretch this portion more at the cost of compressing the higher graylevel zone.
- For positively skewed histogram, logarithmic transformation function performs better than linear stretching.



Non-linear Stretching

- When the graylevel histogram shows high population of pixels in the high graylevel zone of the gray scale, we may want to stretch this portion more at the cost of compressing the lower graylevel zone.

- For negatively skewed histogram, exponential transformation function performs better than linear stretching.



Non-linear Stretching

$$l = (l_{max} - l_{min}) \frac{\ln(m - m_{min} + 1)}{\ln(m_{max} - m_{min} + 1)} + l_{min}$$

Logarithmic transformation function

$$\bar{g}(r, c) = (l_{max} - l_{min}) \frac{\ln(g(r,c) - m_{min} + 1)}{\ln(m_{max} - m_{min} + 1)} + l_{min}$$

Problem

Suppose m be the gray level of input image which is to be transformed to output image gray level l by logarithmic stretching. Suppose for an 8-level image, we have the following frequency table for the input gray levels.

i	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
n_i	a	b	c	d	e	f	g	h

, a to h > 0

Solution

$$m_{min} = 0\,,\ m_{max} = 7,\ l_{min} = 0,\ l_{max} = 7.$$

$$l = (7 - 0)\,\frac{\ln(m-0+1)}{\ln(7-0+1)} + 0\ = 7\,\frac{\ln(m+1)}{\ln 8}$$

- For m = 1, we have l = 2.3 ≈ 2
- For m = 2, we have l = 3.7 ≈ 4

i	0	1	2	3	4	5	6	7
$n_{i'}$	a	0	b	0	c	d+e	f	g+h

Histogram Equalization

- Contrast stretching improves the image quality using some position and distribution independent transformation function.
- In histogram equalization, an image is enhanced in such a way that the gray level histogram (p.d.f) of the processed image attains a desired shape.
- In histogram equalization all gray levels would be equiprobable in the output image.

Histogram Equalization

- It is known that if there are L events with probability p_i associated with i^{th} event (i =0,1,2,…,L-1) such that $\sum_{i=0}^{L-1} p_i = 1$, then the entropy or information content can be defined as $E = -\sum_{i=0}^{L-1} p_i log p_i$
- The E is maximum when

$$p_i = \text{constant for all } i, = 1/L$$

- In this process, the p.d.f would be uniform (theoretically)

Histogram Equalization

- M = No. of rows in the given digital image
- N = No. of columns in the given digital image
- {0,1,2,…,l_{max} = L-1} = Allowable gray levels in the given digital image
- n_i = No. of pixels having gray level i
- p_i = Probability that a pixel has gray level i = $\frac{n_i}{MN}$
- m_i = i^{th} normalized gray level in the given digital image = $\frac{i}{L-1}$
- l_i = i^{th} gray level in the output image

The transform function, $l_i = (L-1)\sum_{j=0}^{i} p_j$

Problem

Suppose m be the gray level of input image, l be the gray level of output image.

m	0	1	2	3	4	5	6	7
n_m	123	78	281	417	639	1054	816	688

8-level image of size 64 × 64

Solution

$\sum_{m=0}^{7} n_m = 4096,$

$p_m = n_m/4096.$

Cumulative probability

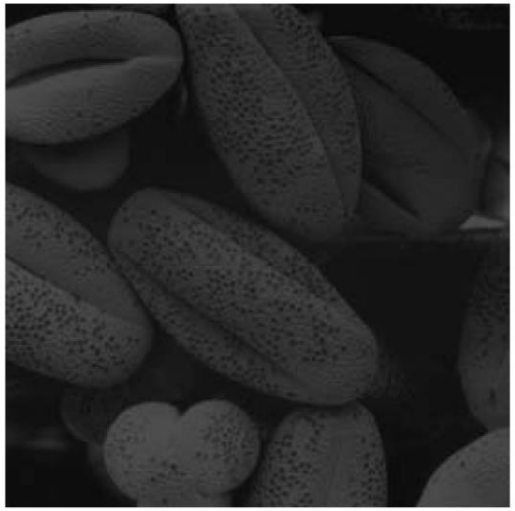
$c_m = \sum_{i=0}^{m} p_i$

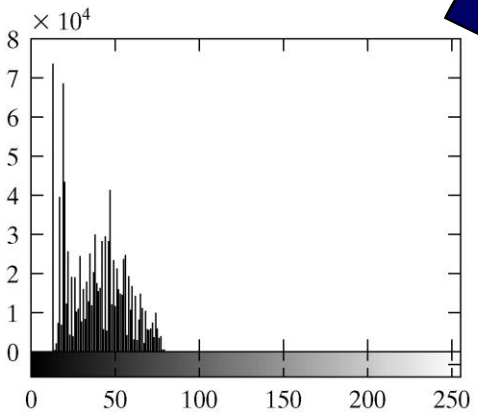
The output gray level is computed as $l = 7\ c_m$

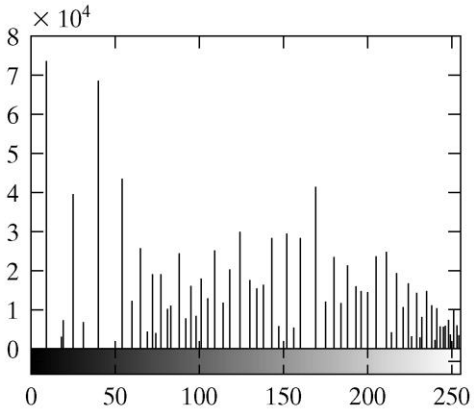
m	n_m	p_m	c_m	l
0	123	0.03	0.03	0.21 ≈ 0
1	78	0.019	0.049	0.34 ≈ 0
2	281	0.069	0.118	0.82 ≈ 1
3	417	0.102	0.22	1.54 ≈ 2
4	639	0.156	0.376	2.62 ≈ 3
5	1054	0.257	0.633	4.43 ≈ 4
6	816	0.199	0.832	5.82 ≈ 6
7	688	0.168	1.0	7.00 ≈ 7

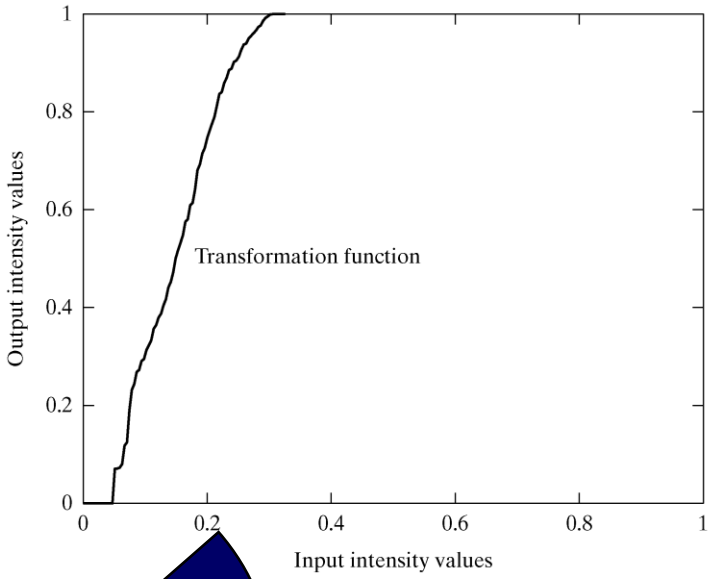
l	0	1	2	3	4	5	6	7
n_l	201	281	417	639	1054	0	816	688

Equalisation Transformation Function



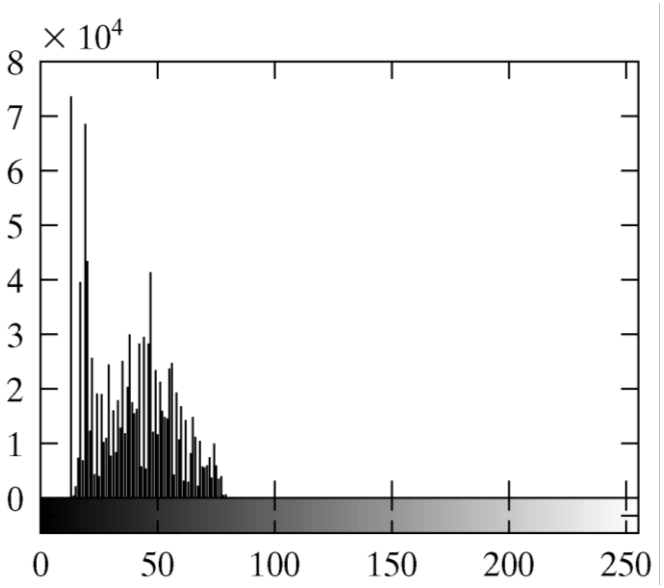


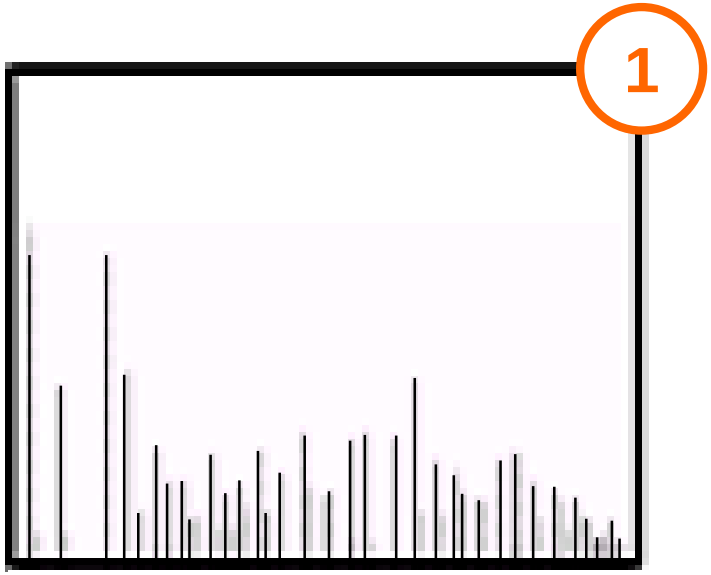


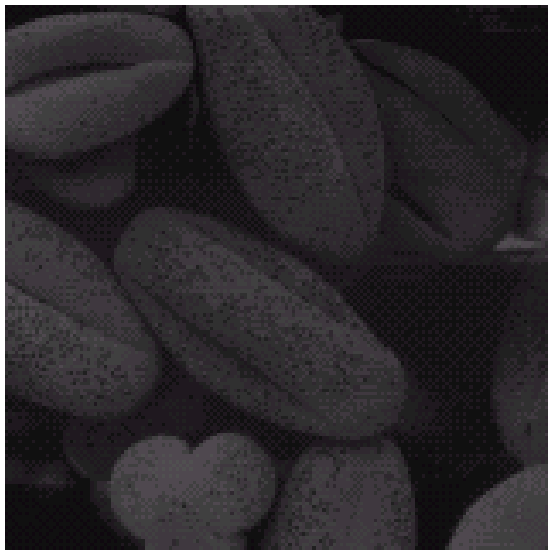


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Equalisation Examples

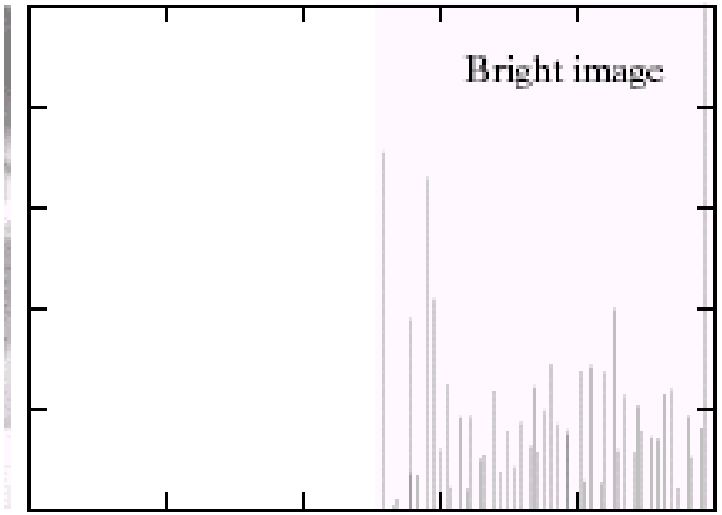


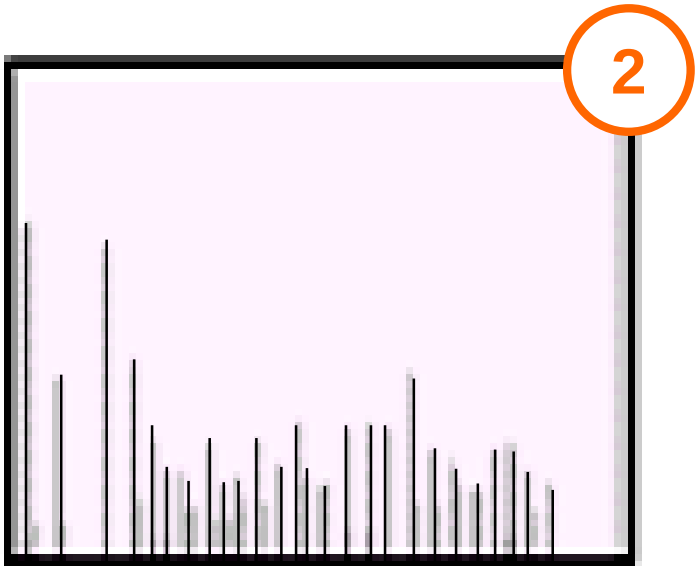




Images taken from Gonzalez & Woods, Digital Image Processing (2002)

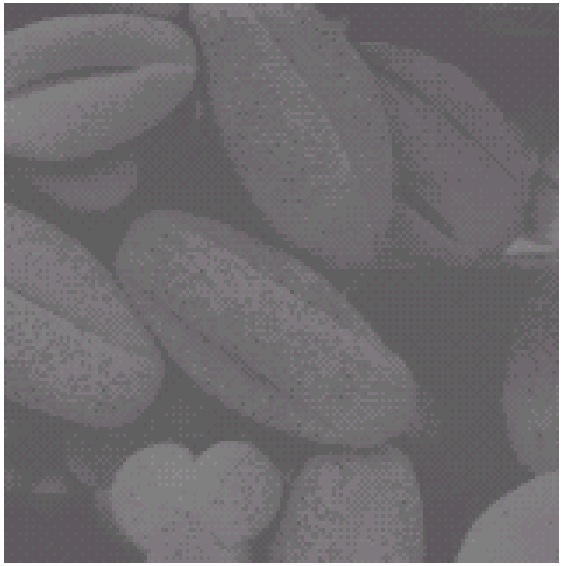
Equalisation Examples

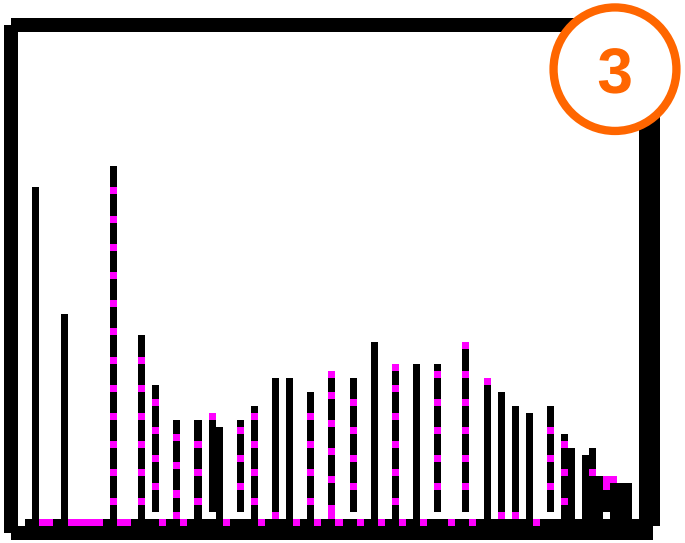


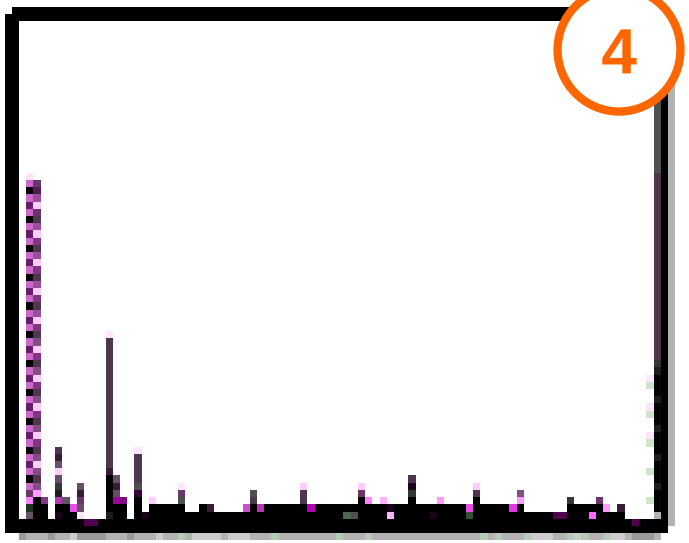


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Equalisation Examples



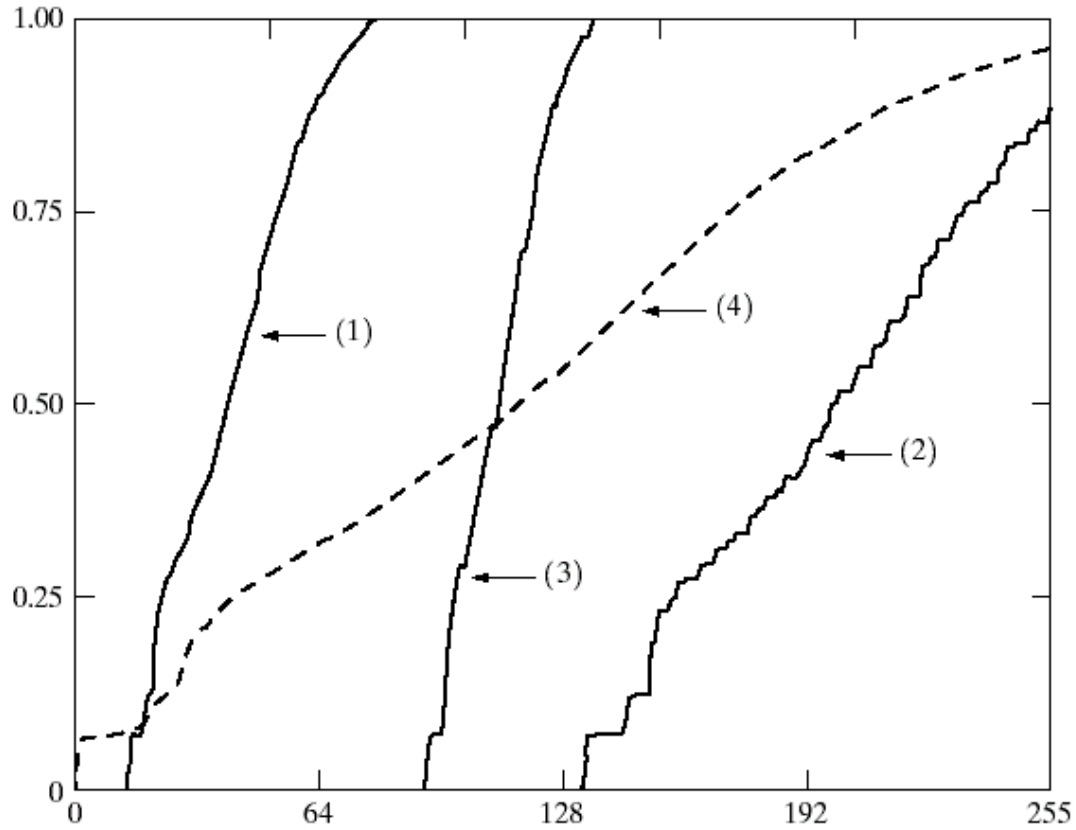




Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Equalisation Transformation Functions

The functions used to equalise the images in the previous examples.



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

Noise Cleaning or Smoothing

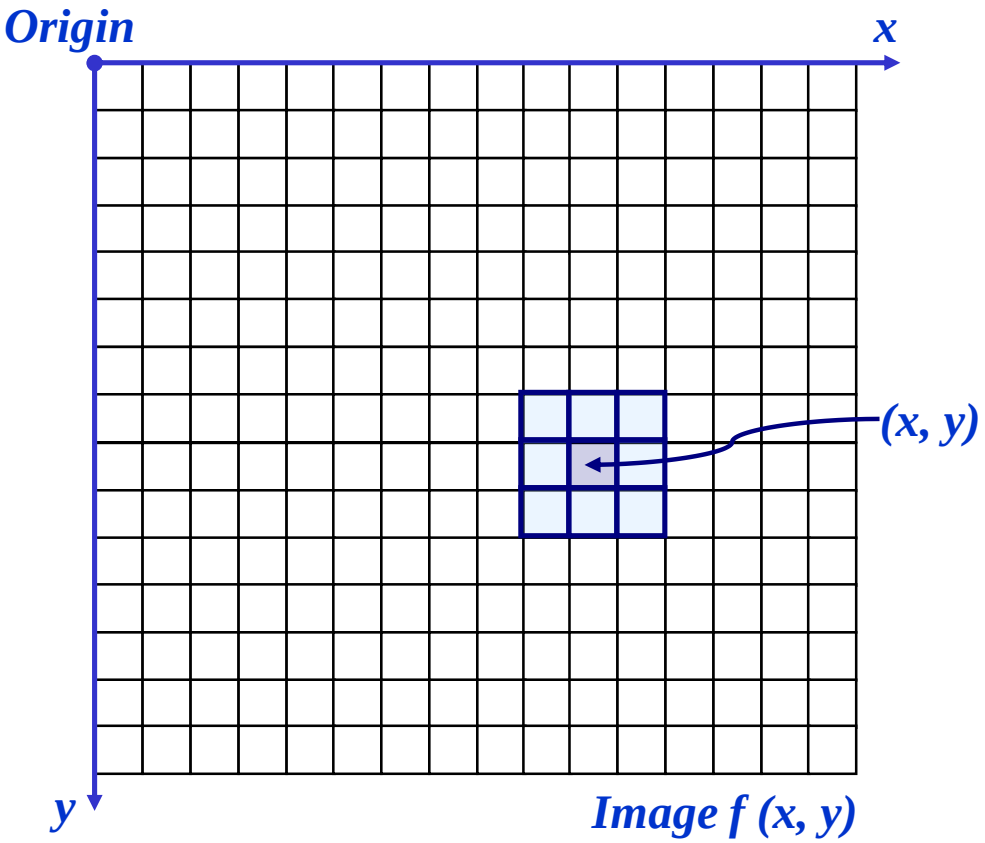
- Point Processing
- Neighbourhood Processing

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

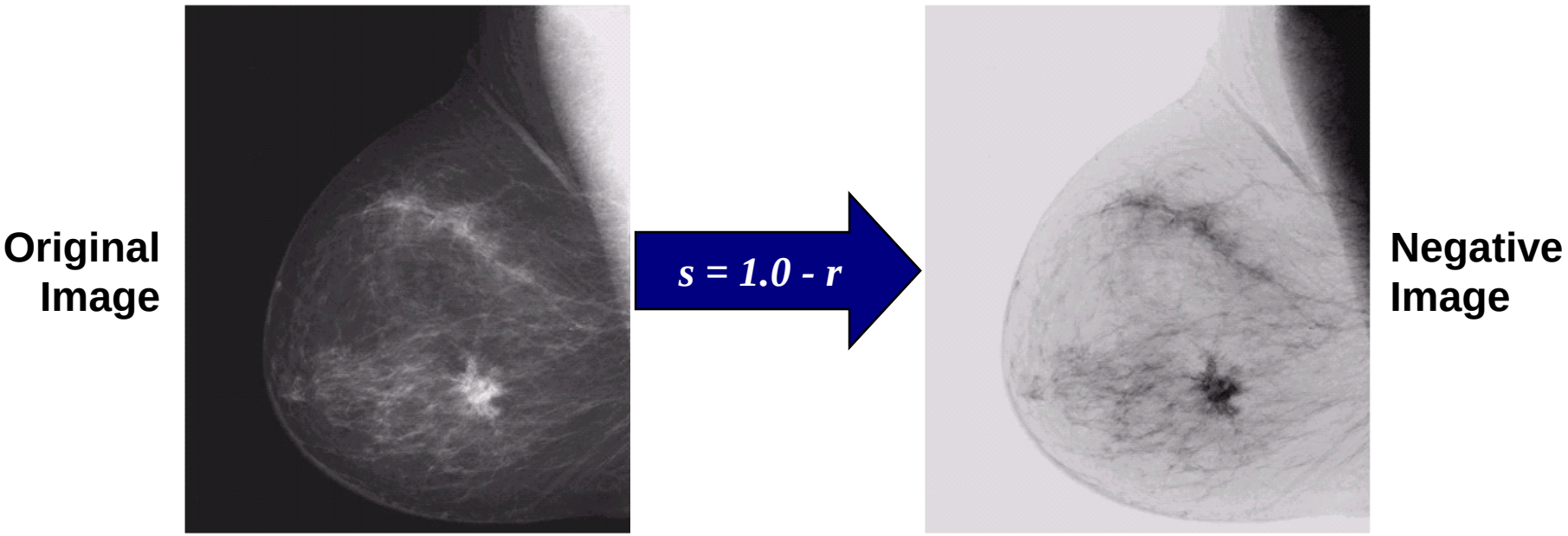
$$s = T(r)$$

where s refers to the processed image pixel value and r refers to the original image pixel value.

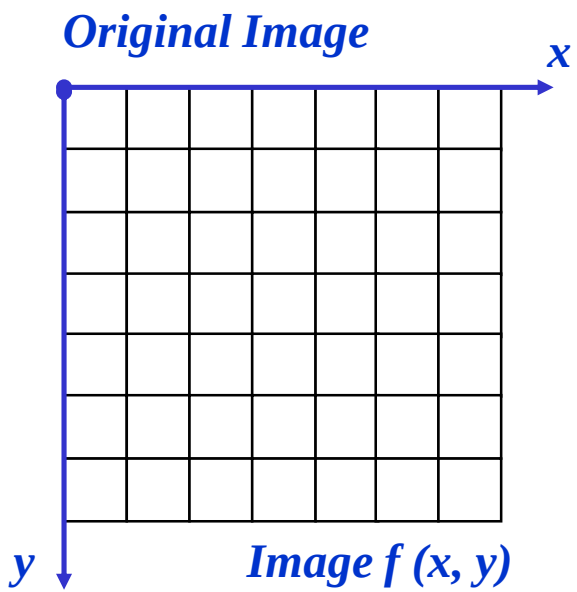
Point Processing Example: Negative Images

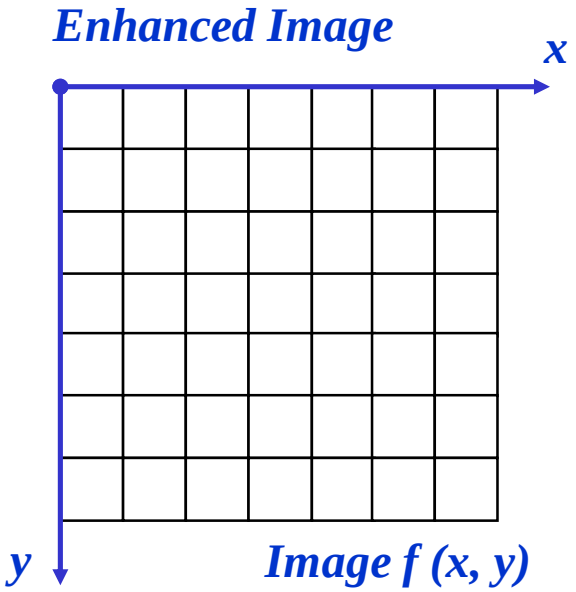
Negative images are useful for enhancing white or grey detail embedded in dark regions of an image.

Note how much clearer the tissue is in the negative image of the mammogram below.



Point Processing Example: Negative Images

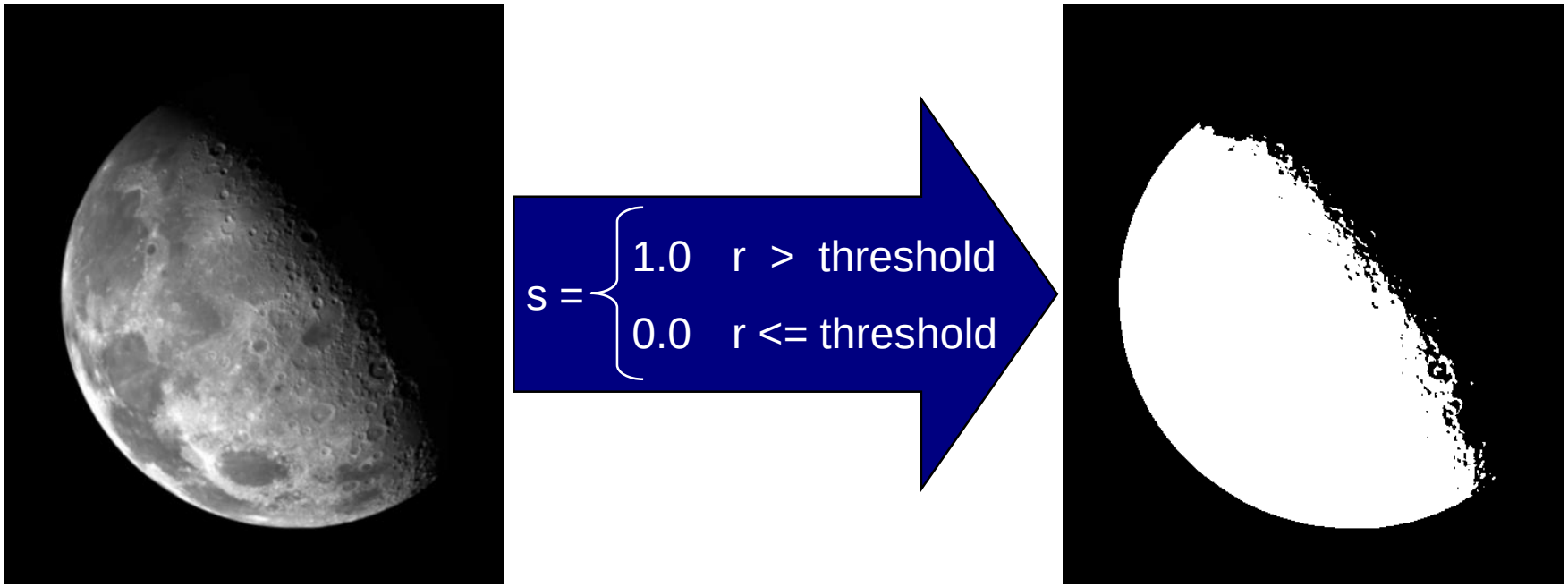




$$s = intensity_{max} - r$$

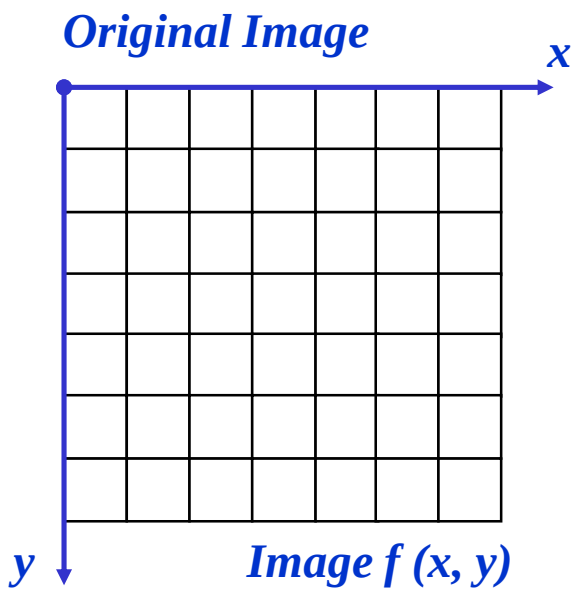
Point Processing Example: Thresholding

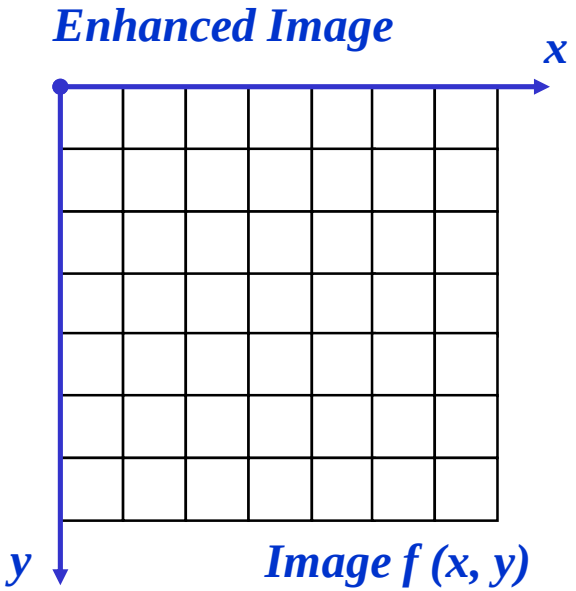
Thresholding transformations are particularly useful for segmentation in which we want to isolate an object of interest from a background.



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

Point Processing Example: Thresholding





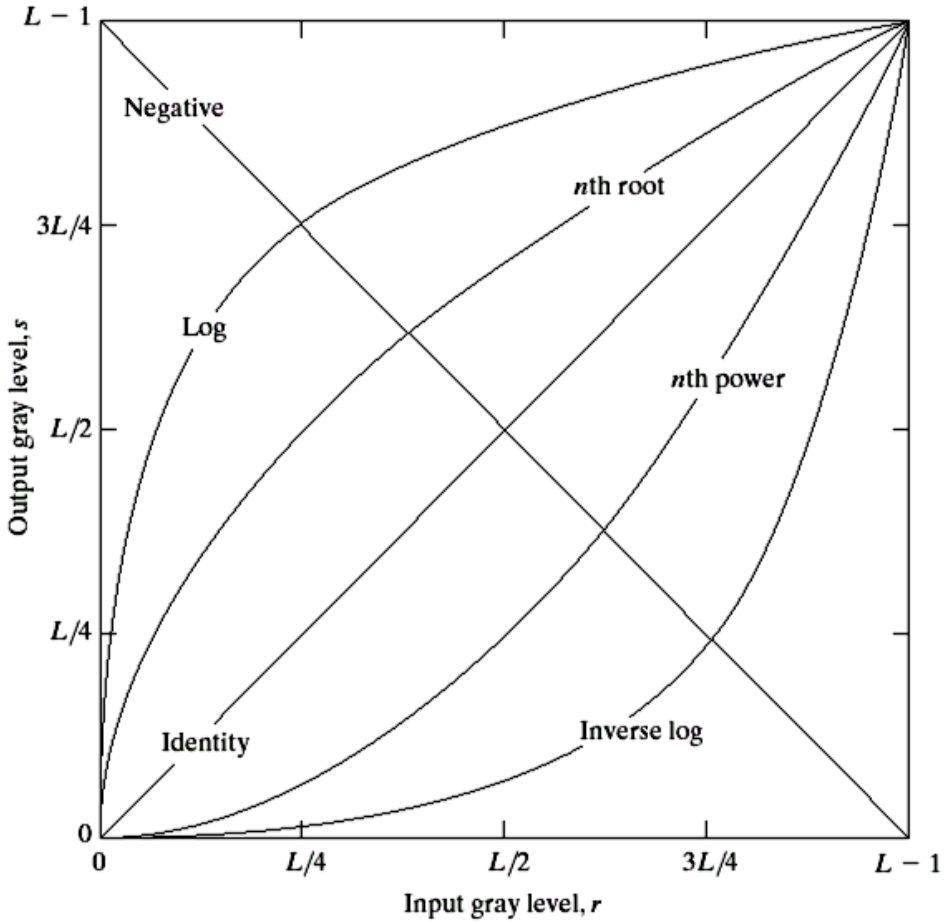
$$s = \begin{cases} 1.0 & r > threshold \\ 0.0 & r <= threshold \end{cases}$$

Basic Gray Level Transformations

There are many different kinds of gray level transformations.

Three of the most common are:

- 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

$$s = c \times log(1 + r)$$

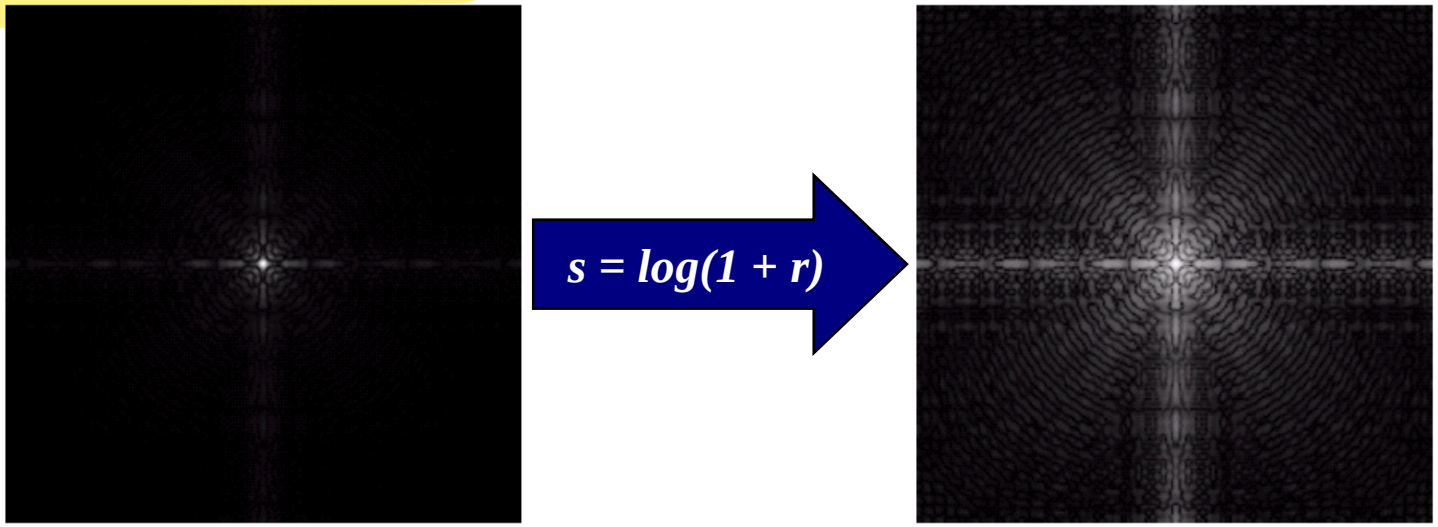
The log transformation maps a narrow range of low input grey level values into a wider range of output values.

The inverse log transformation performs the opposite transformation.

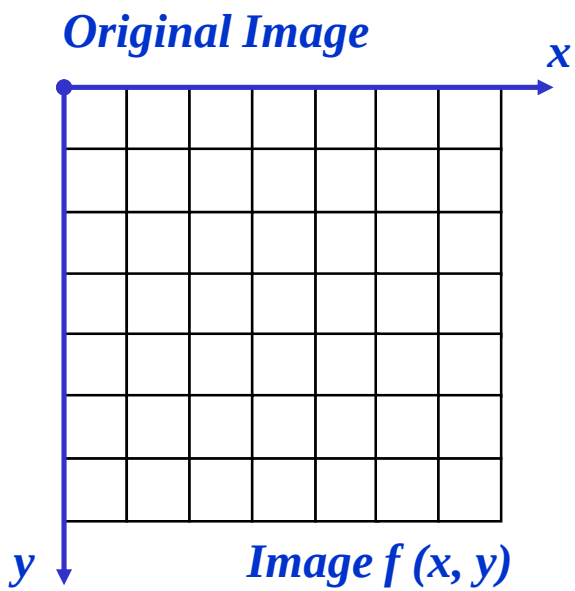
Logarithmic Transformations

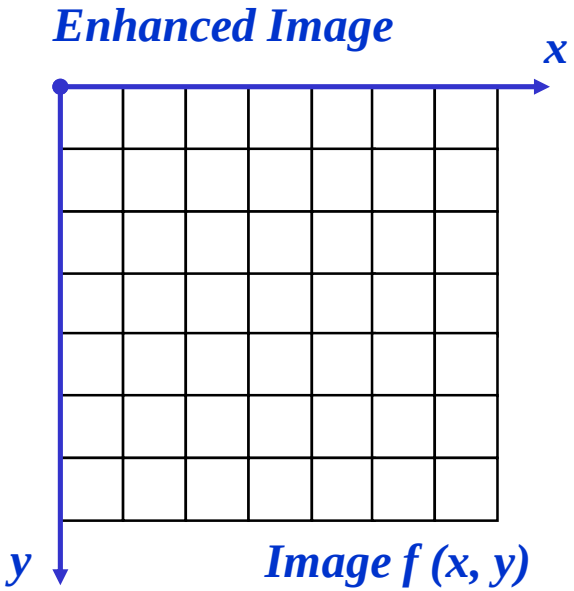
Log functions are particularly useful when the input grey level values may have an extremely large range of values.

In this example the Fourier transform of an image is put through a log transform to reveal more detail.



Logarithmic Transformations





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

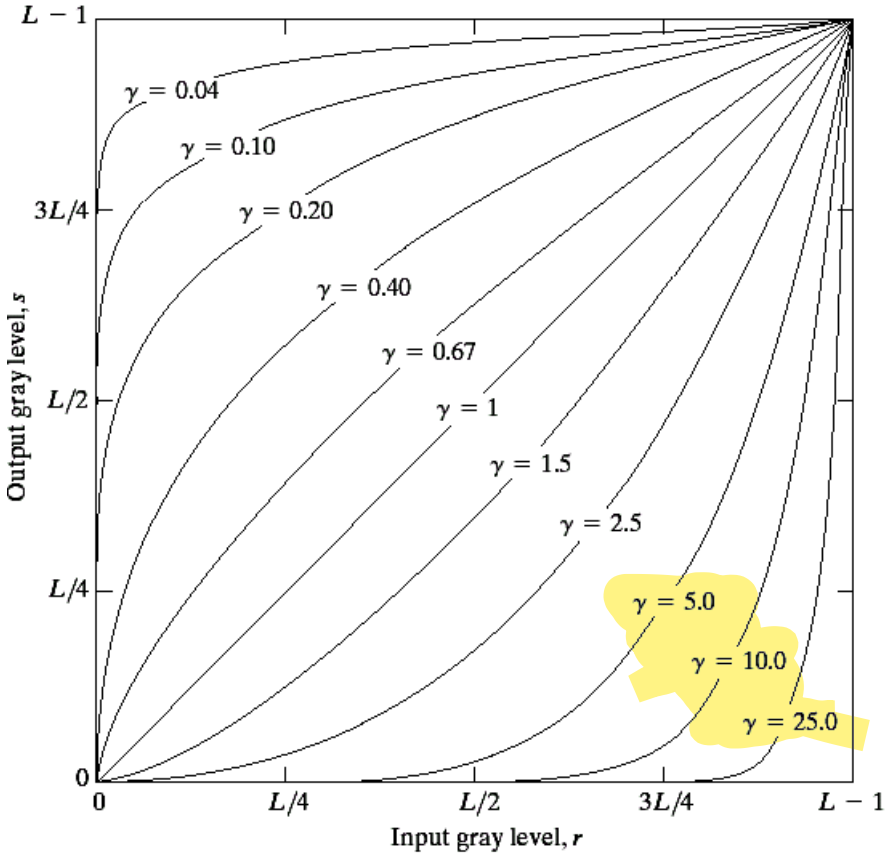
Power Law Transformations

Power law transformations have the following form

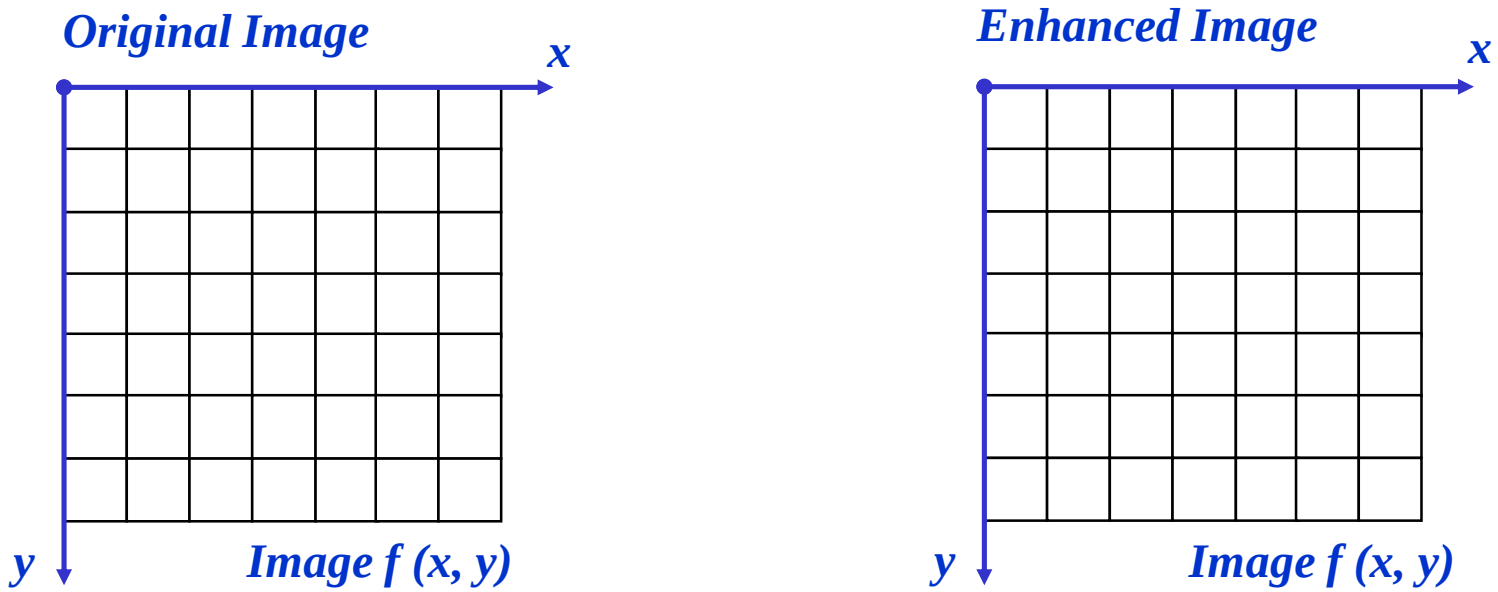
$$s = c \times r^{\gamma}$$

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.



Power Law Transformations

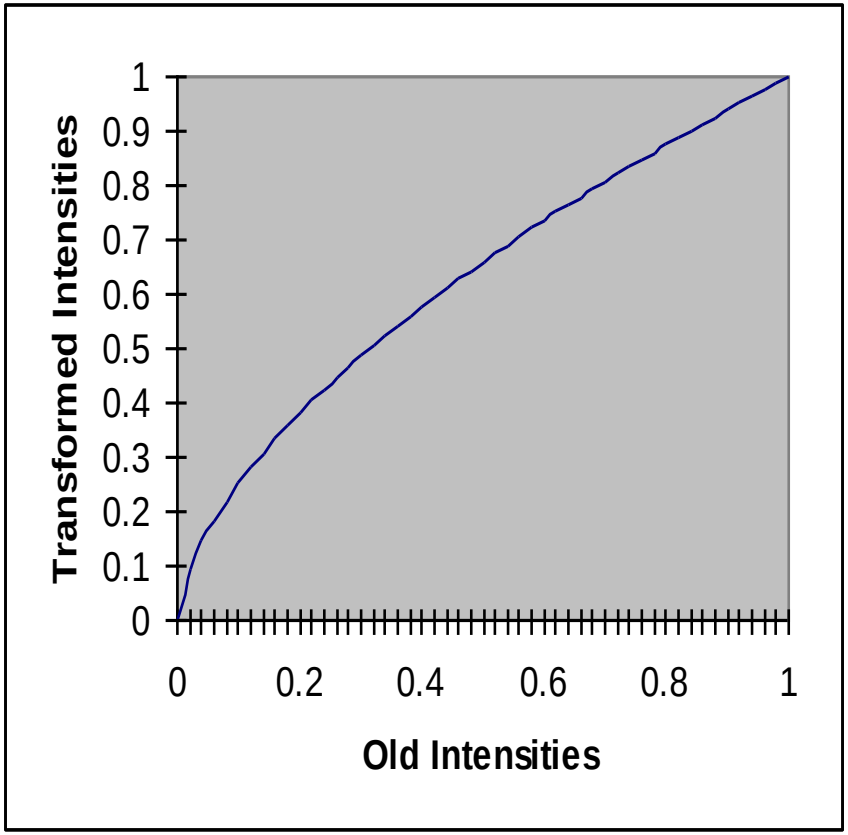


We usually set c to 1.

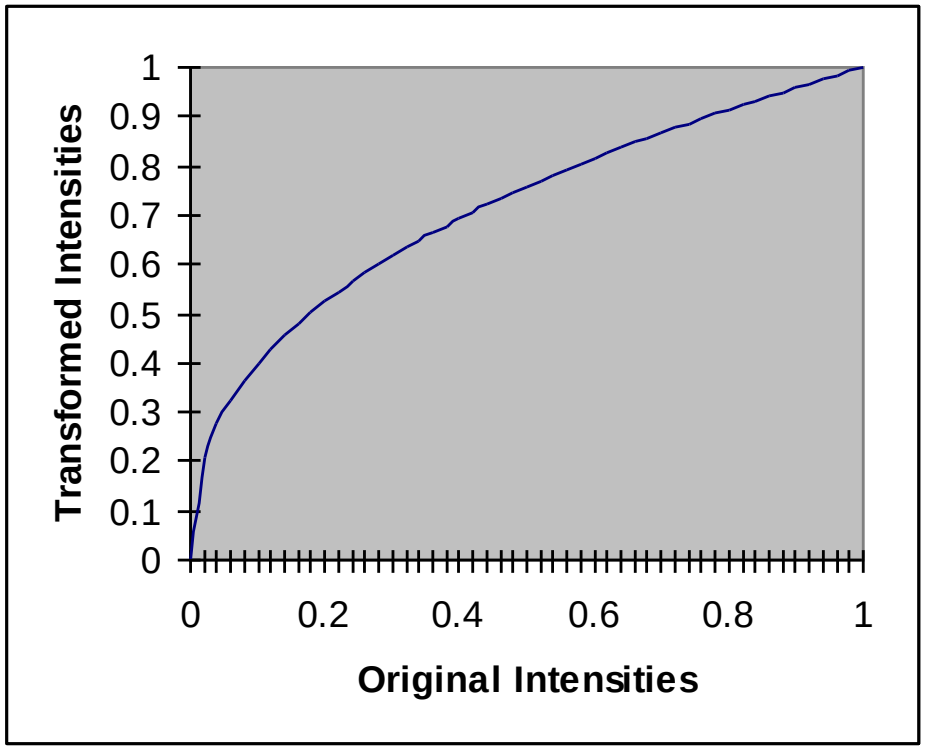
Grey levels must be in the range [0.0, 1.0].

Power Law Example

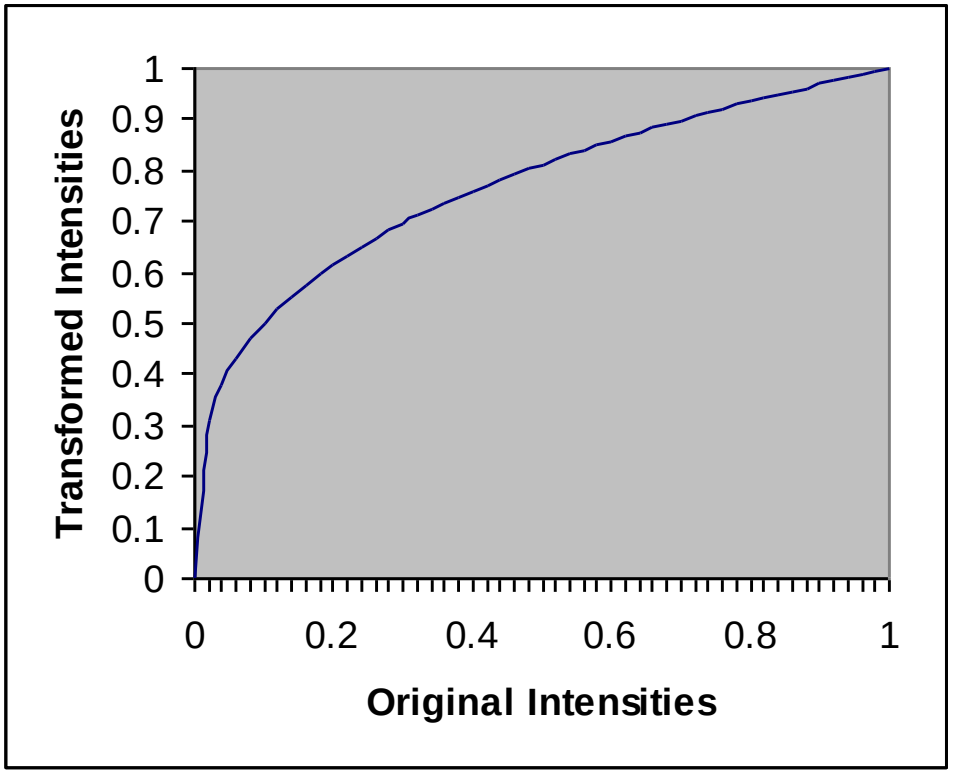
Power Law Example



Power Law Example



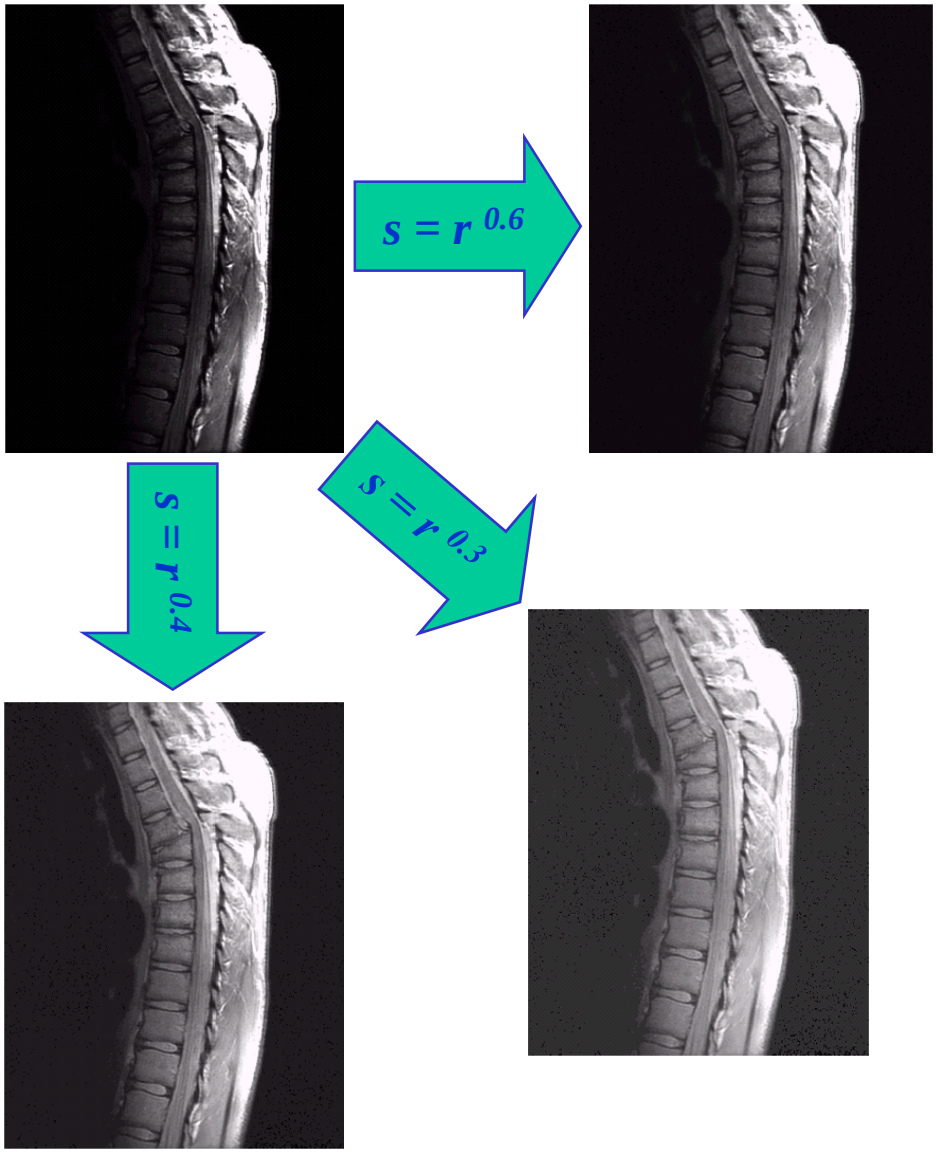
Power Law Example



Power Law Example

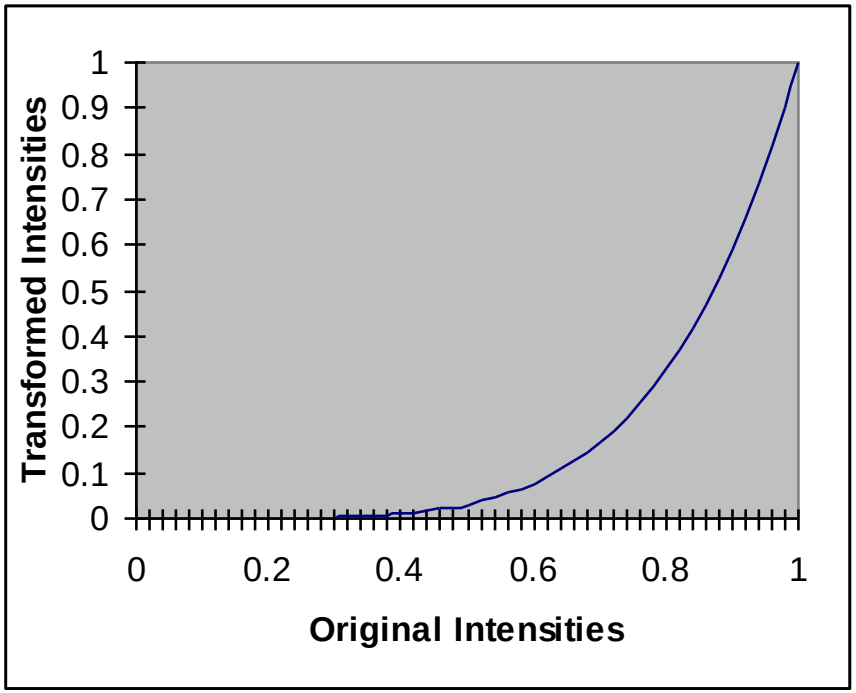
The images to the right show a magnetic resonance (MR) image of a fractured human spine.

Different curves highlight different detail.



Power Law Example

Power Law Example

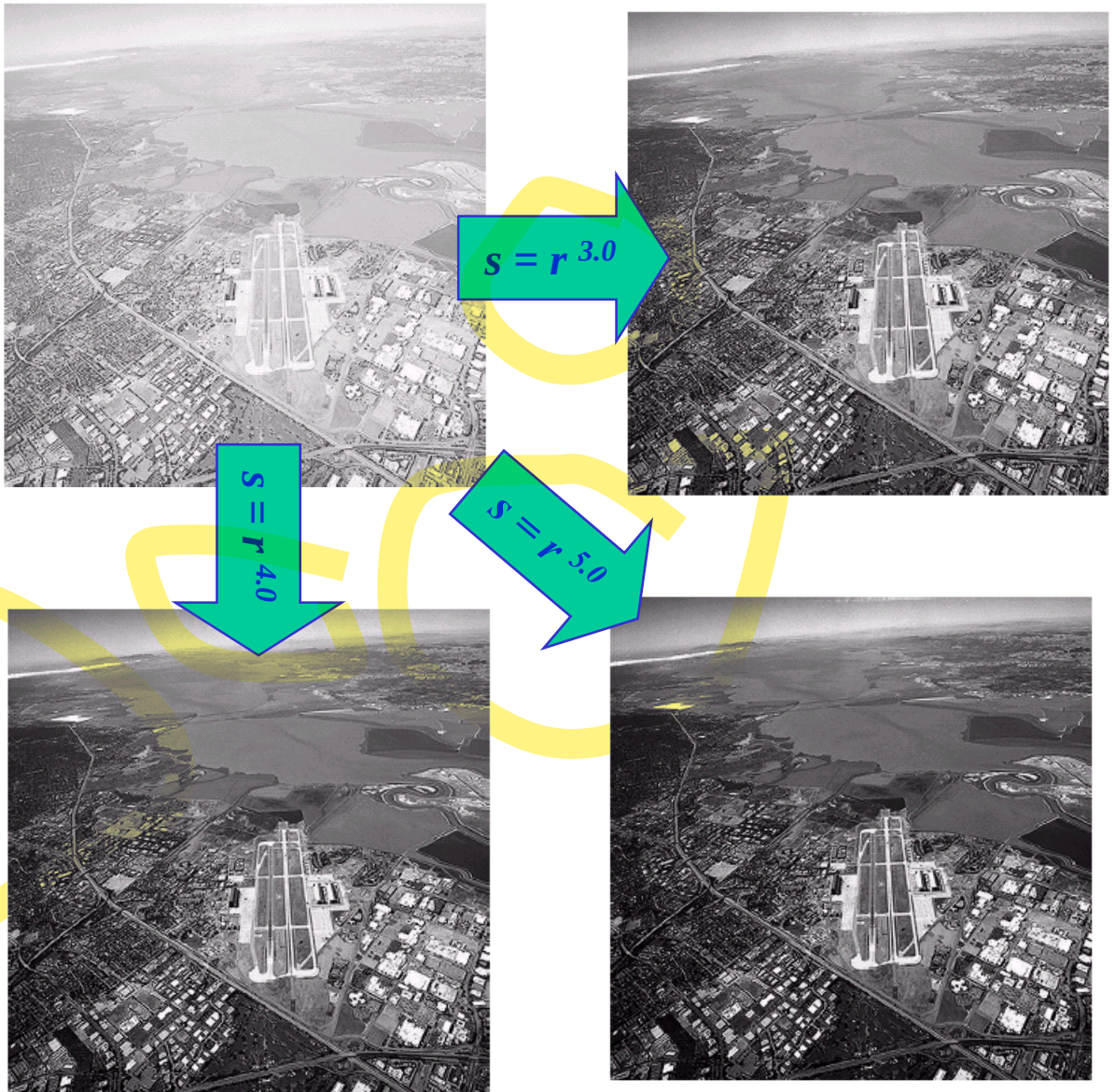


Power Law Example

An aerial photo of a runway is shown.

Power law transforms are used to darken the image.

Different curves highlight different detail.

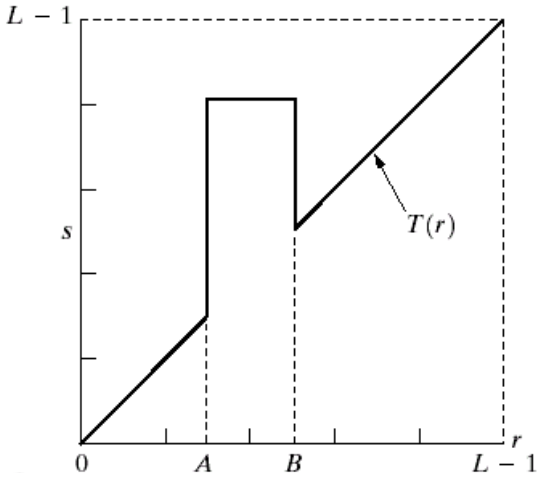


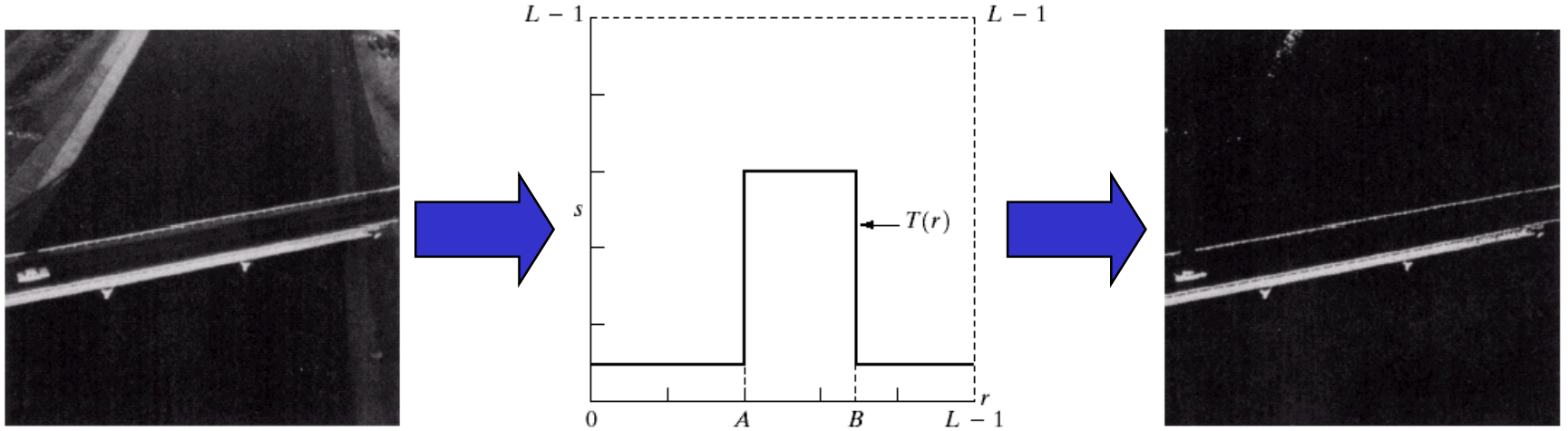
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

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.



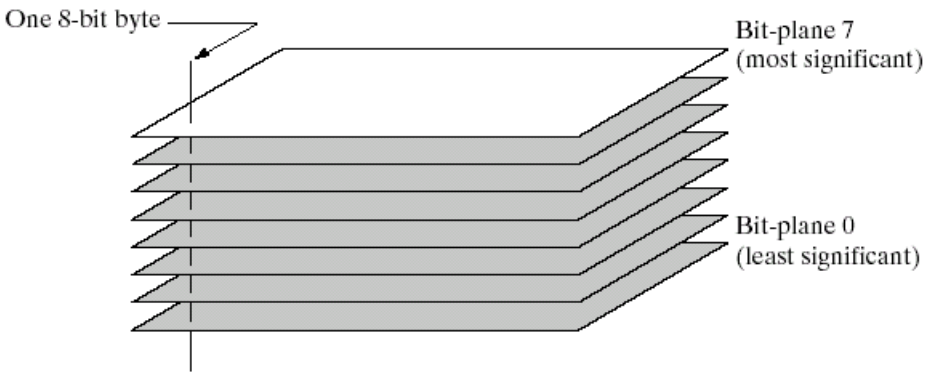


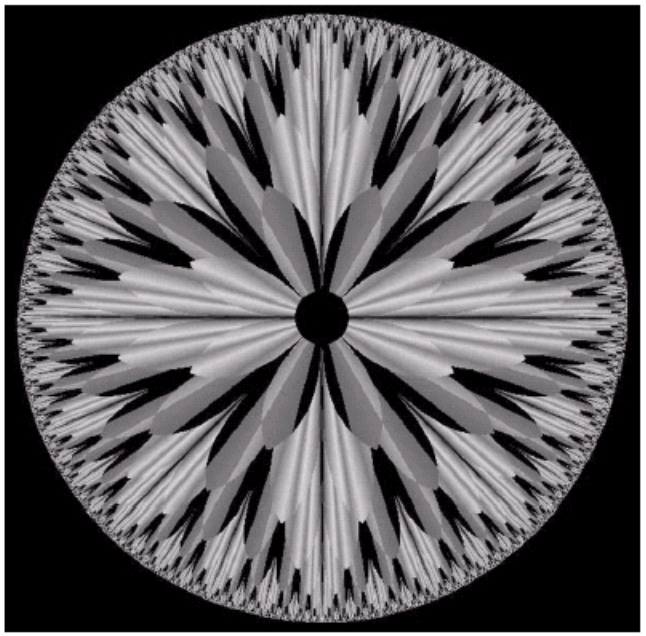
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing

Often by isolating particular bits of the pixel values in an image we can highlight interesting aspects of that image.

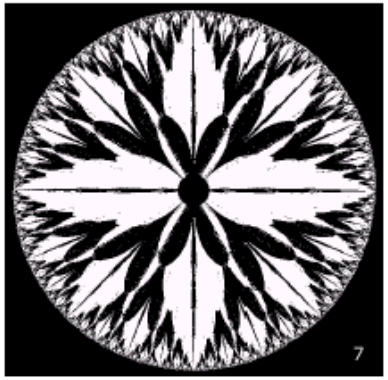
- Higher-order bits usually contain most of the significant visual information.
- Lower-order bits contain subtle details.

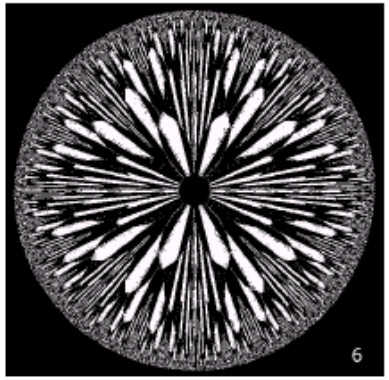




Bit Plane Slicing

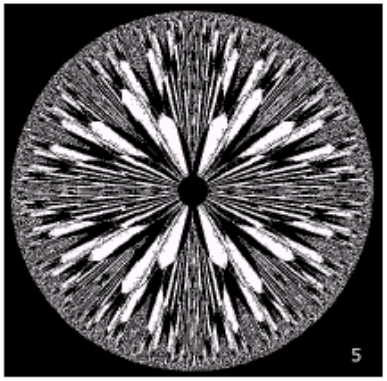
[10000000]

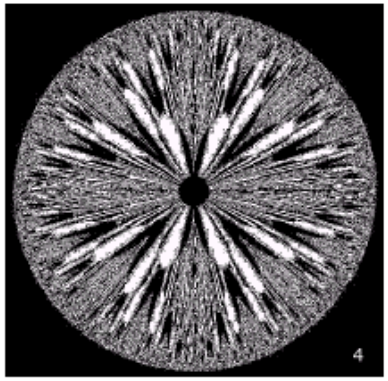


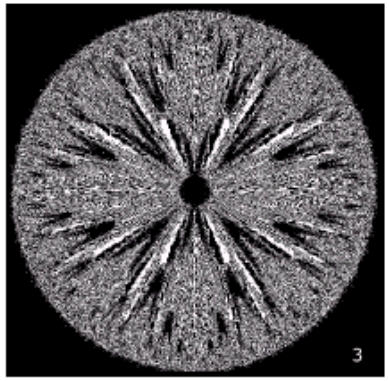


[01000000]

[00100000]

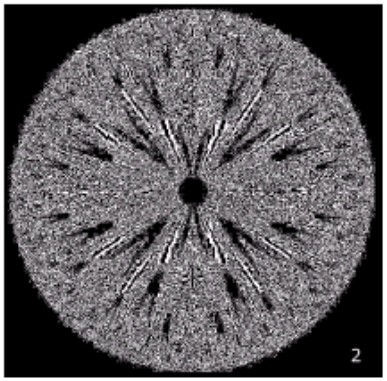


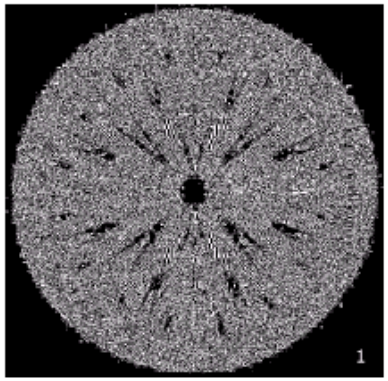


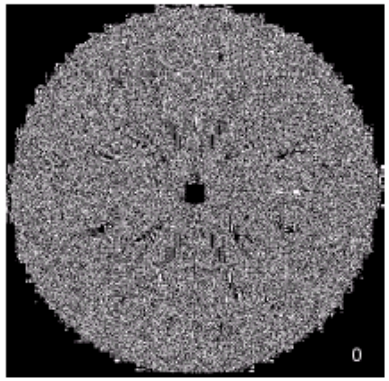


[00001000]

[00000100]







[00000001]

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

Bit Plane Slicing

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

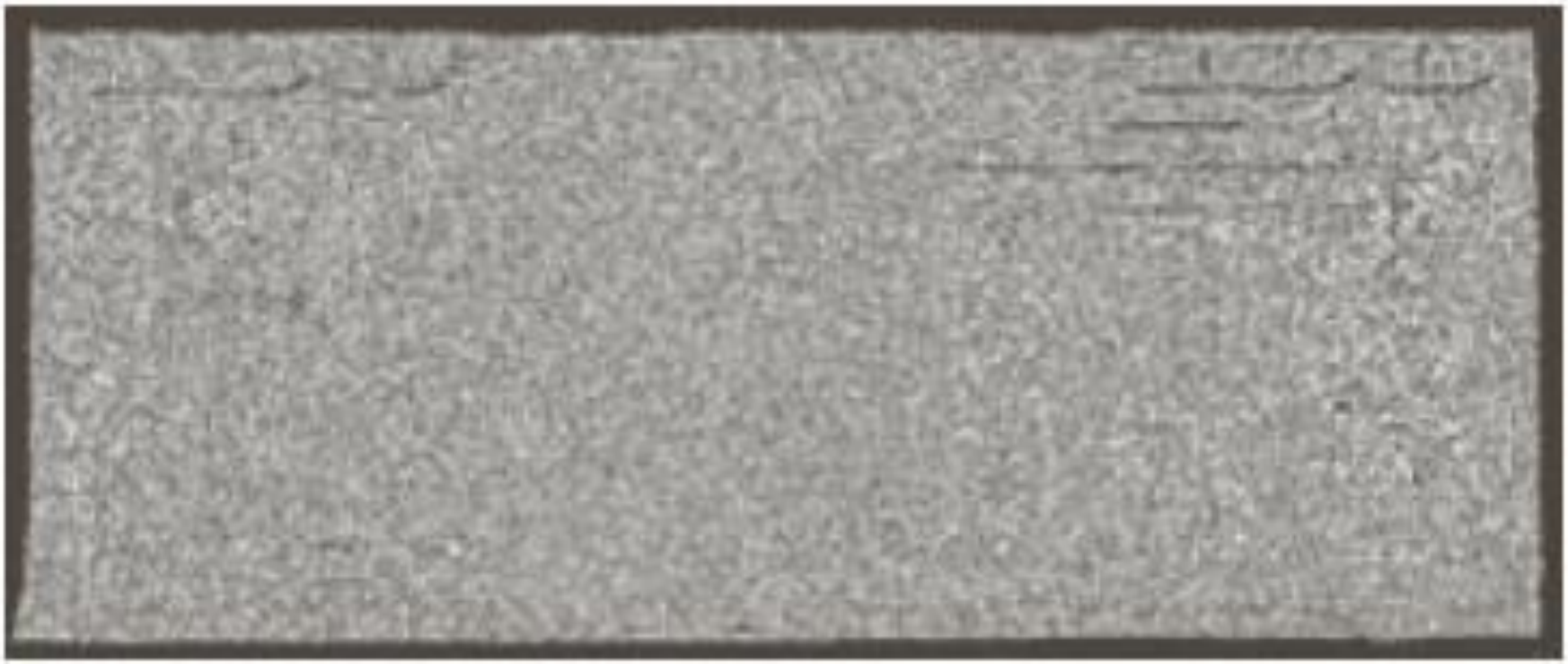
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing

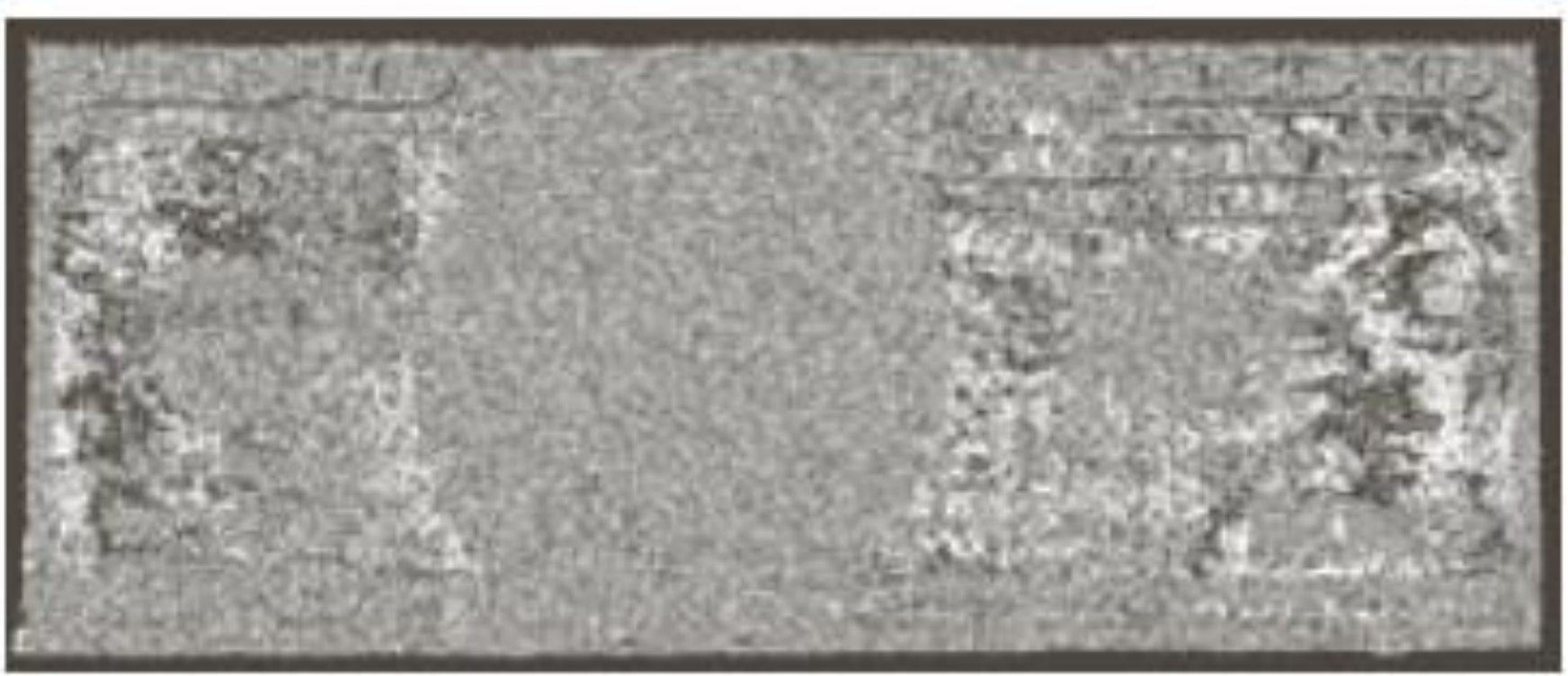
Bit Plane Slicing

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

Bit Plane Slicing

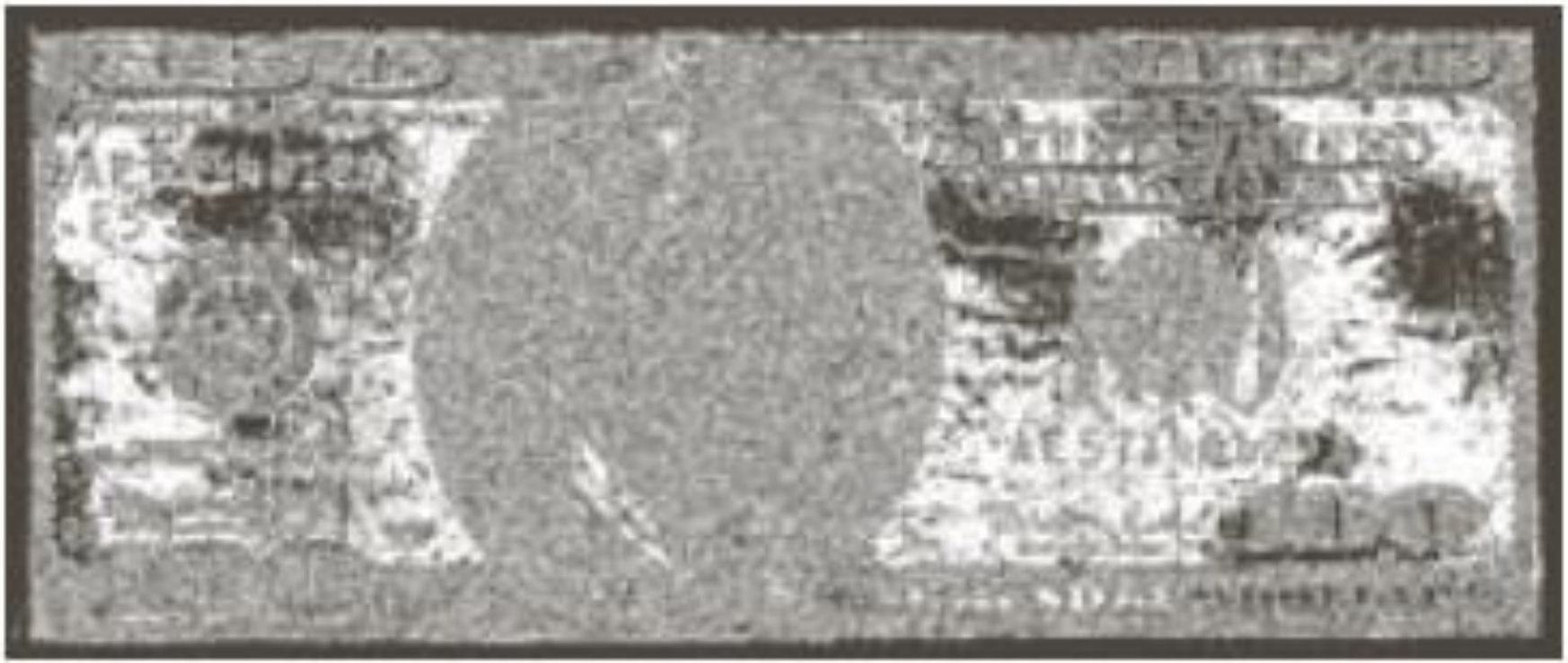


Bit Plane Slicing



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

Bit Plane Slicing



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

Bit Plane Slicing

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

Bit Plane Slicing

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

Bit Plane Slicing

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

Bit Plane Slicing

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

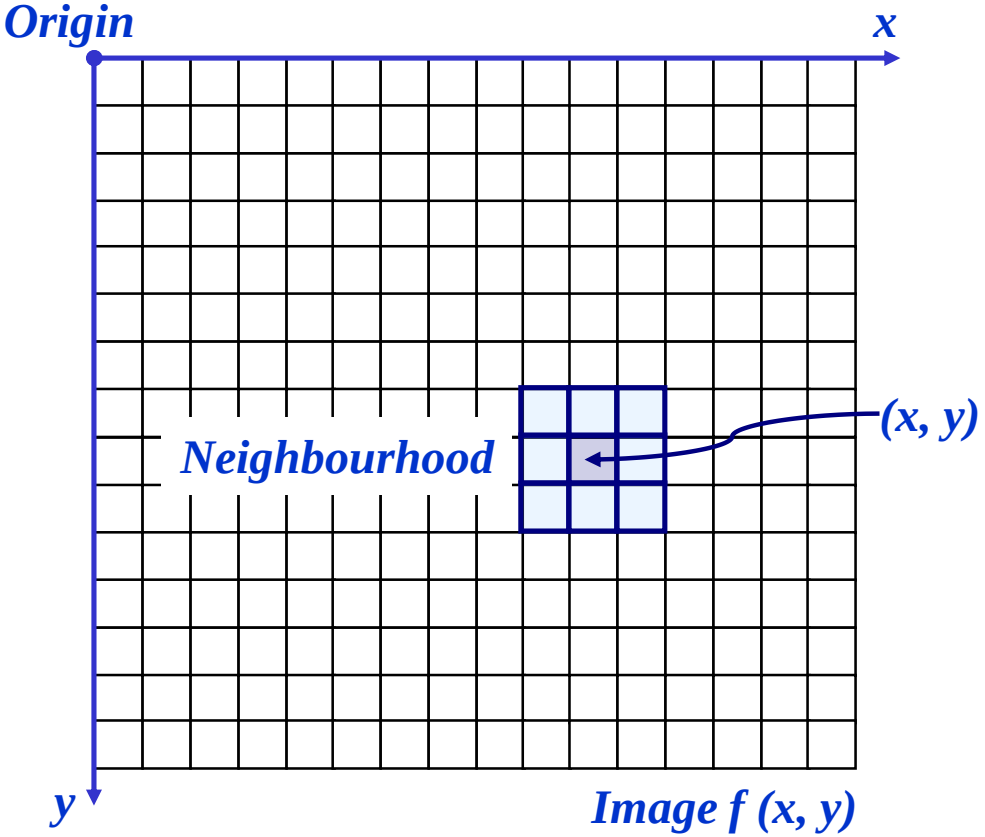
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

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

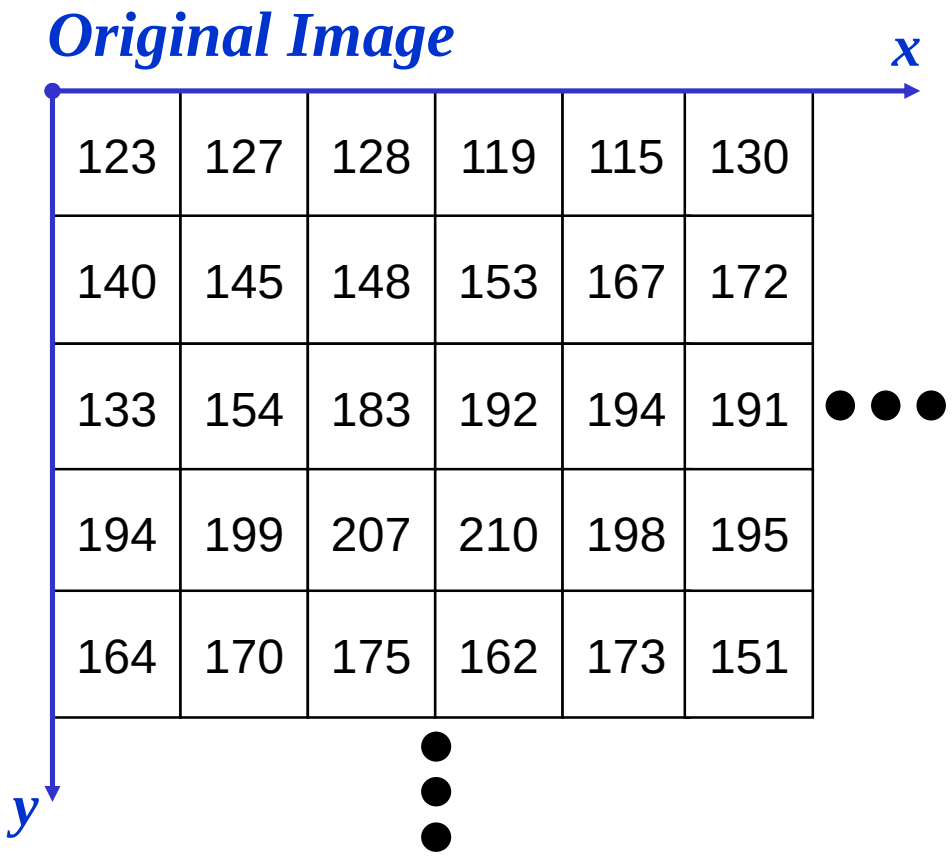
Some simple neighbourhood operations:

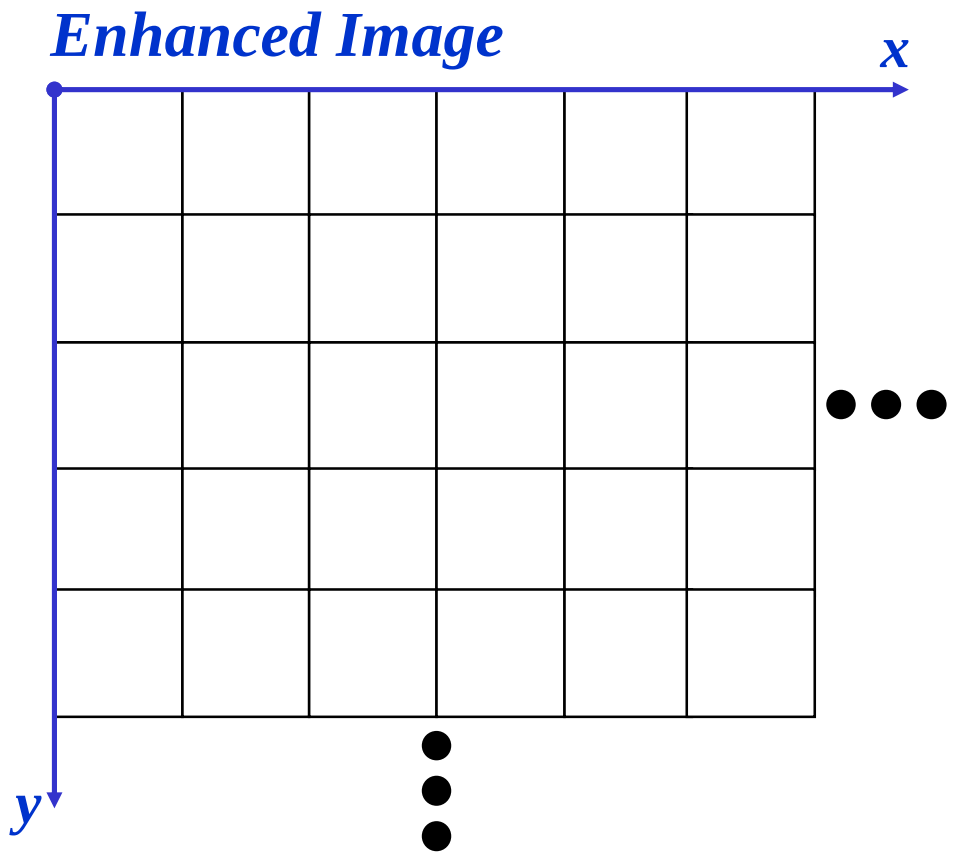
Min: Set the pixel value to the minimum in the neighbourhood.

Max: Set the pixel value to the maximum in the neighbourhood.

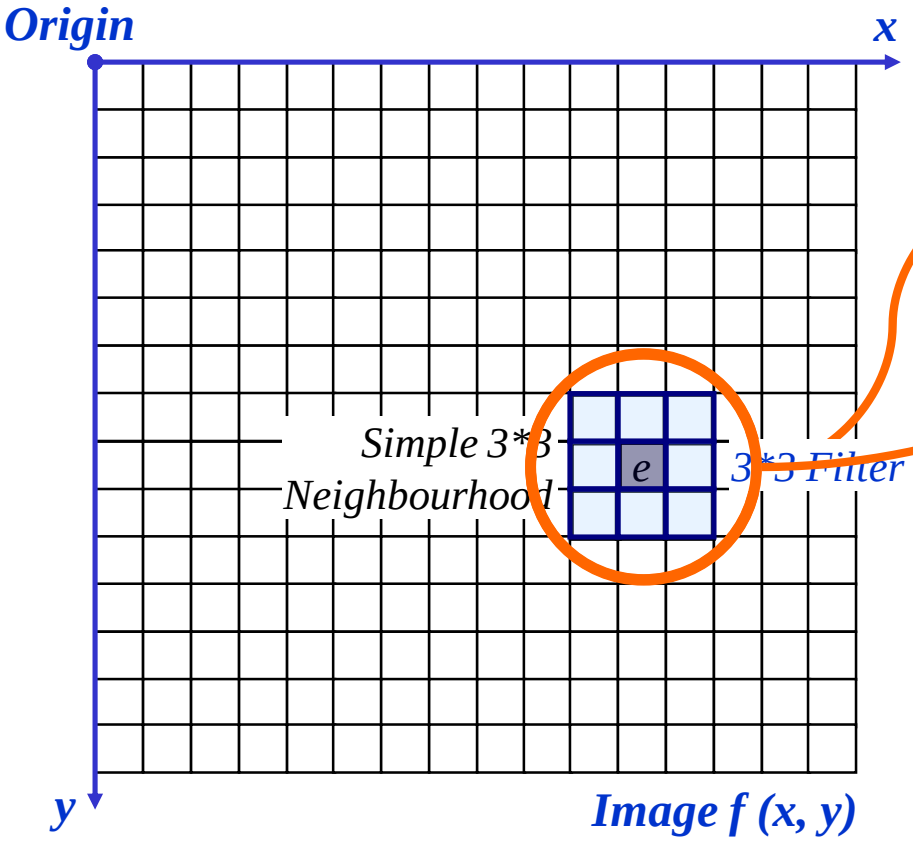
Median: The median value of a set of numbers is the midpoint value in that set (e.g. from the set [1, 7, 15, 18, 24] 15 is the median). Sometimes the median works better than the average.

Simple Neighbourhood Operations Example





The Spatial Filtering Process : Convolution



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

Original Image Pixels

*

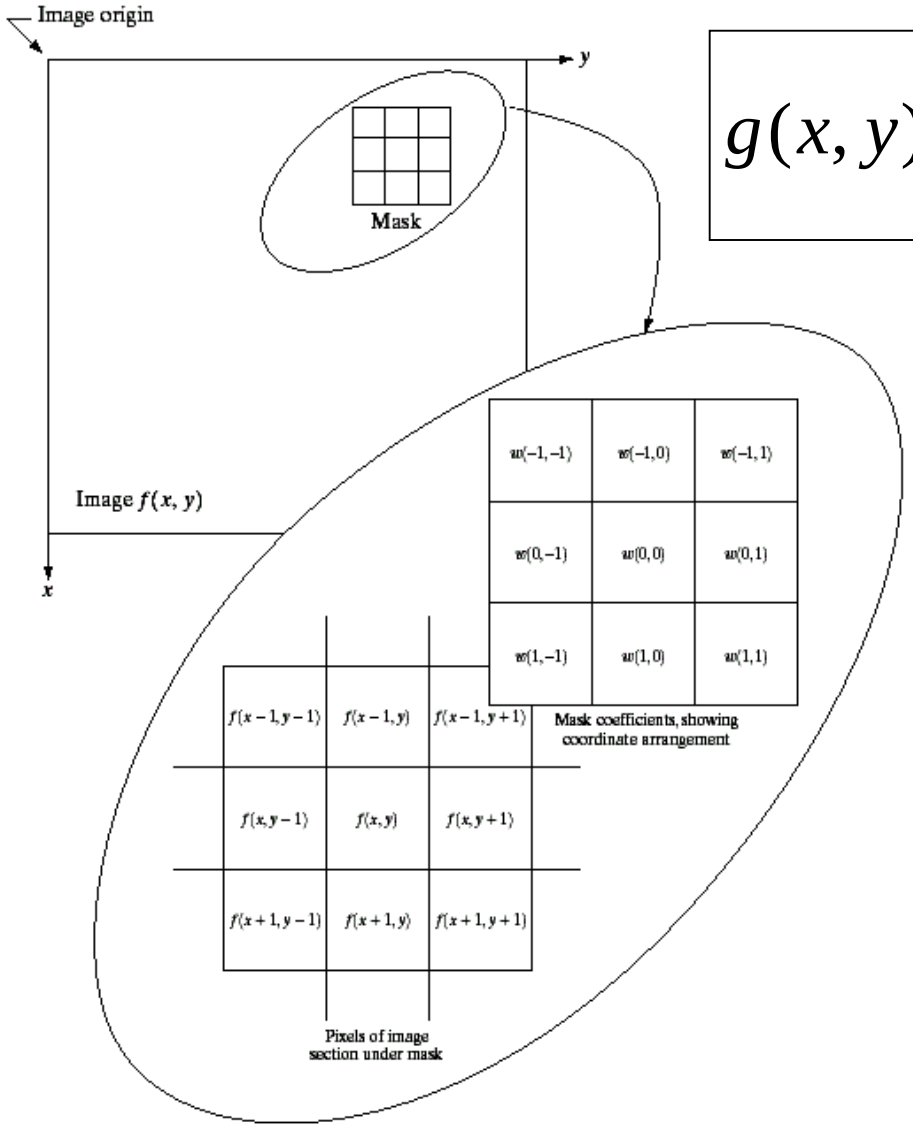
r	*s*	*t*
u	*v*	*w*
x	*y*	*z*

Filter

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

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

Spatial Filtering: Equation Form



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

$$a = (m-1)/2$$
$$b = (n-1)/2$$

Size of mask is $m \times n$

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

Types of Smoothing Filters

- Smoothing Linear Filter
  - Averaging Filter

- Order Statistics Filter /Non-linear Filter
  - Median Filter
  - Max Filter
  - Min Filter

Averaging Filters

One of the simplest spatial filtering operations we can perform is a smoothing operation.

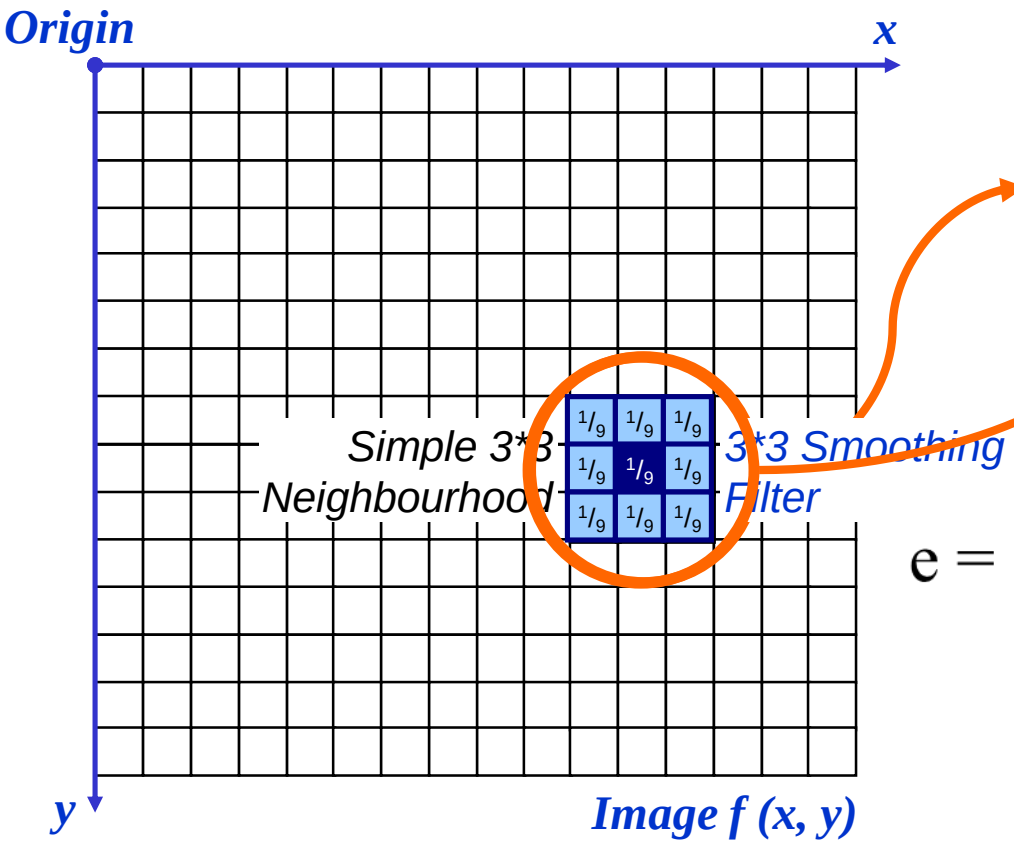
Simply average all of the pixels in a neighbourhood around a central value.

Especially useful in removing noise from images.

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

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

$$
\begin{aligned}
e = {} & 1/9*106 + \\
& 1/9*104 + 1/9*100 + 1/9*108 + \\
& 1/9*99 + 1/9*98 + \\
& 1/9*95 + 1/9*90 + 1/9*85 \\
= {} & 98.3333 \approx 98
\end{aligned}
$$

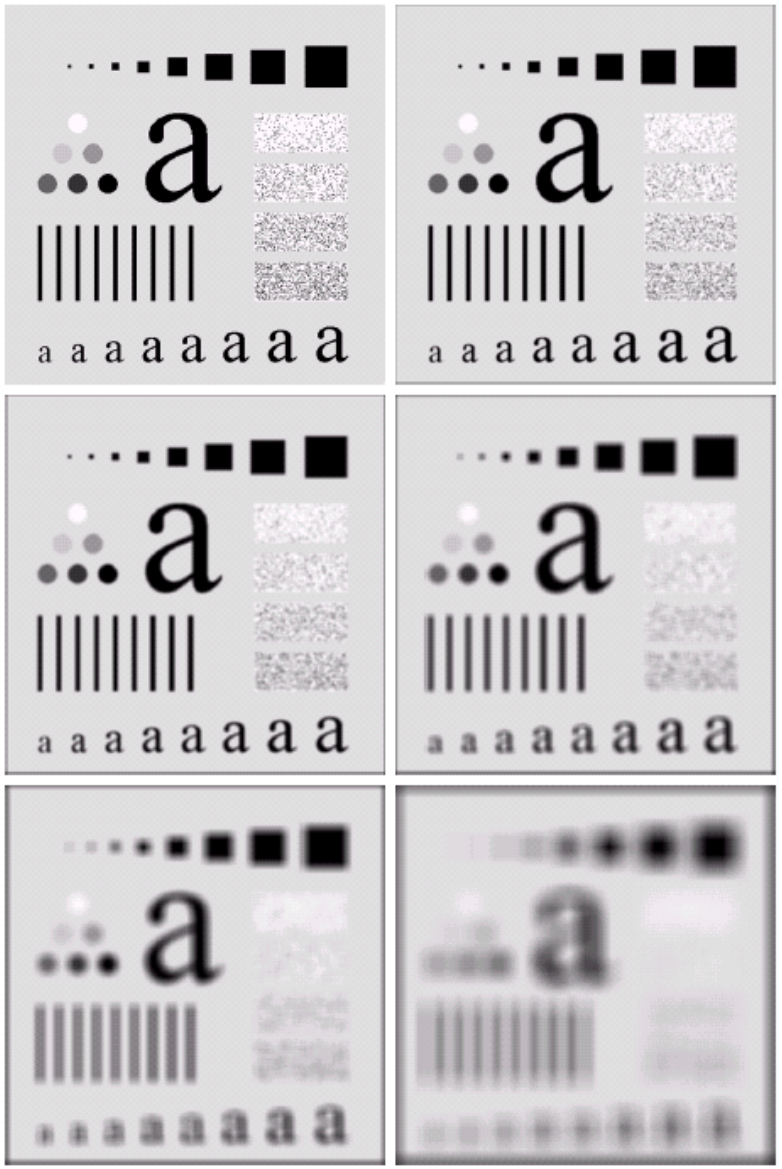
The above process is repeated for every pixel in the original image to generate the smoothed image.

Image Smoothing Example

The image at the top left is an original image of size 500×500 pixels.

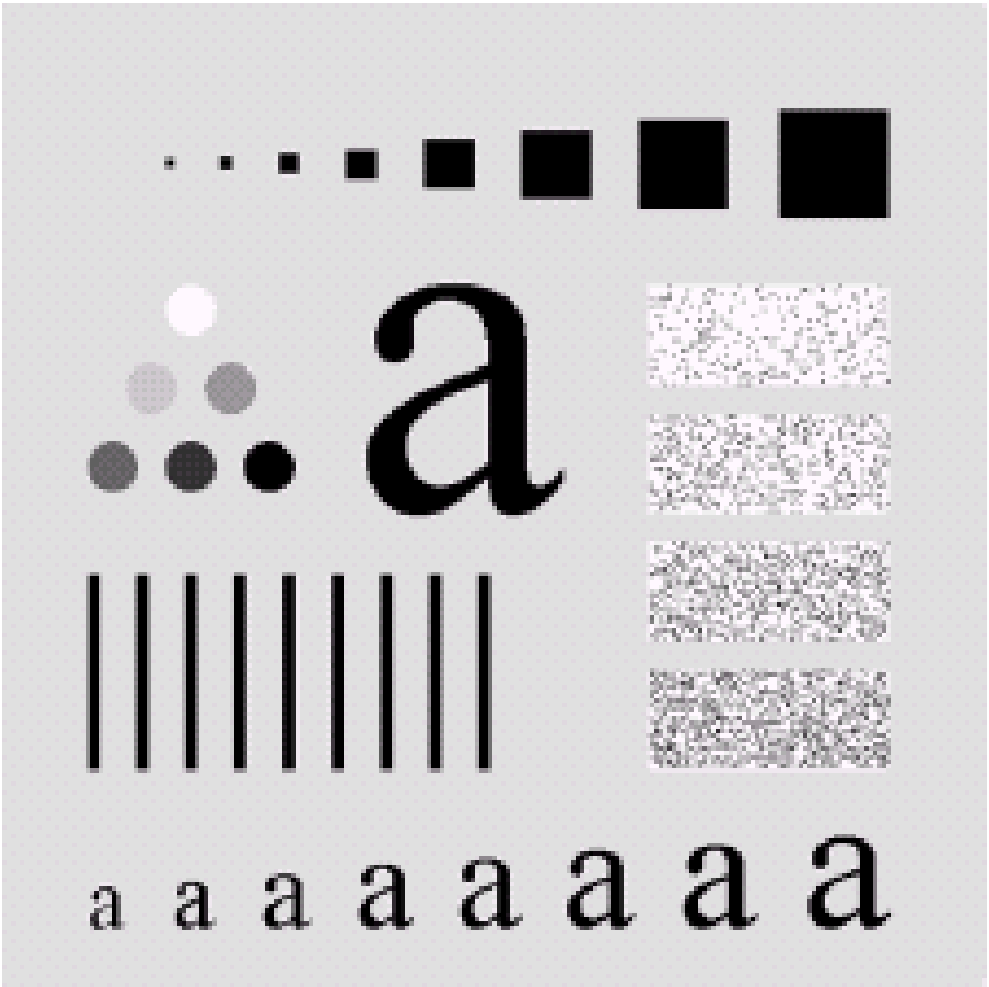
The subsequent images show the image after filtering with an averaging filter of increasing size
3, 5, 9, 15, and 35.

Notice that how detail in image begins blurred when the filter size becomes larger.



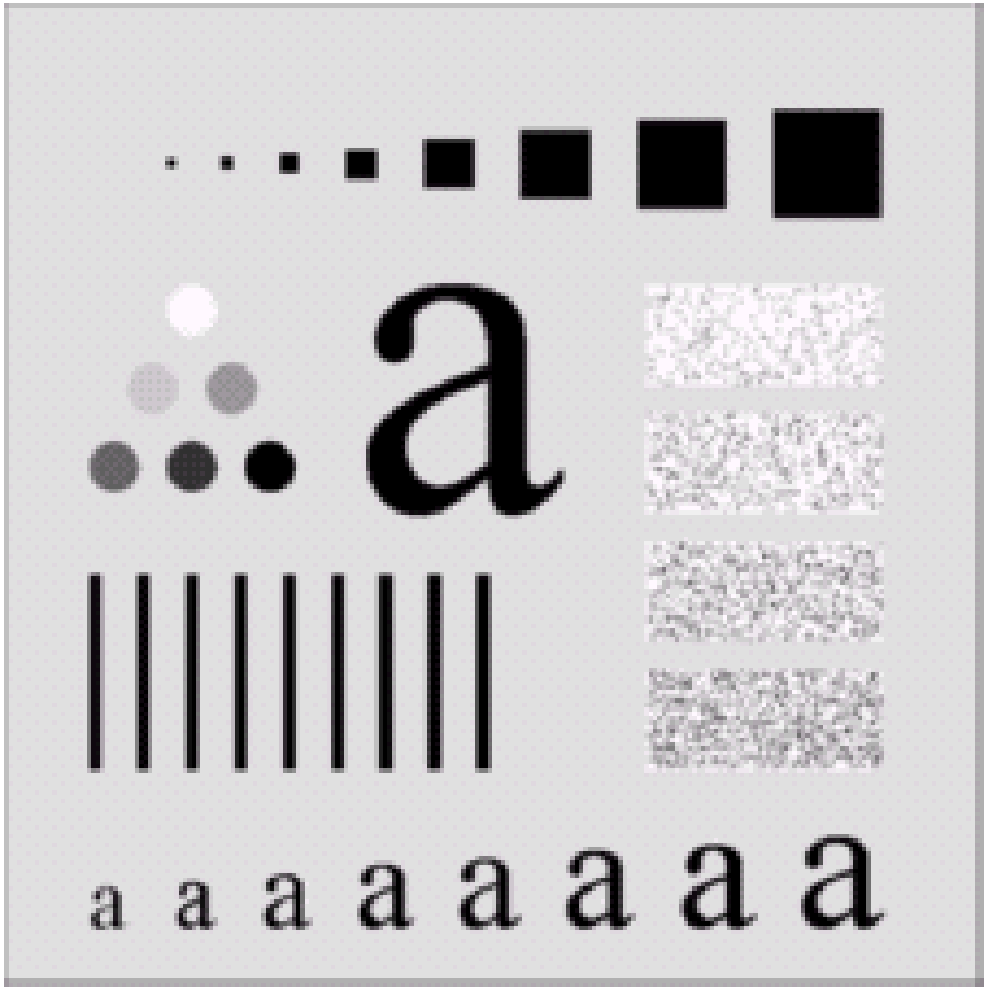
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Image Smoothing Example



Original image of size 500×500 pixels

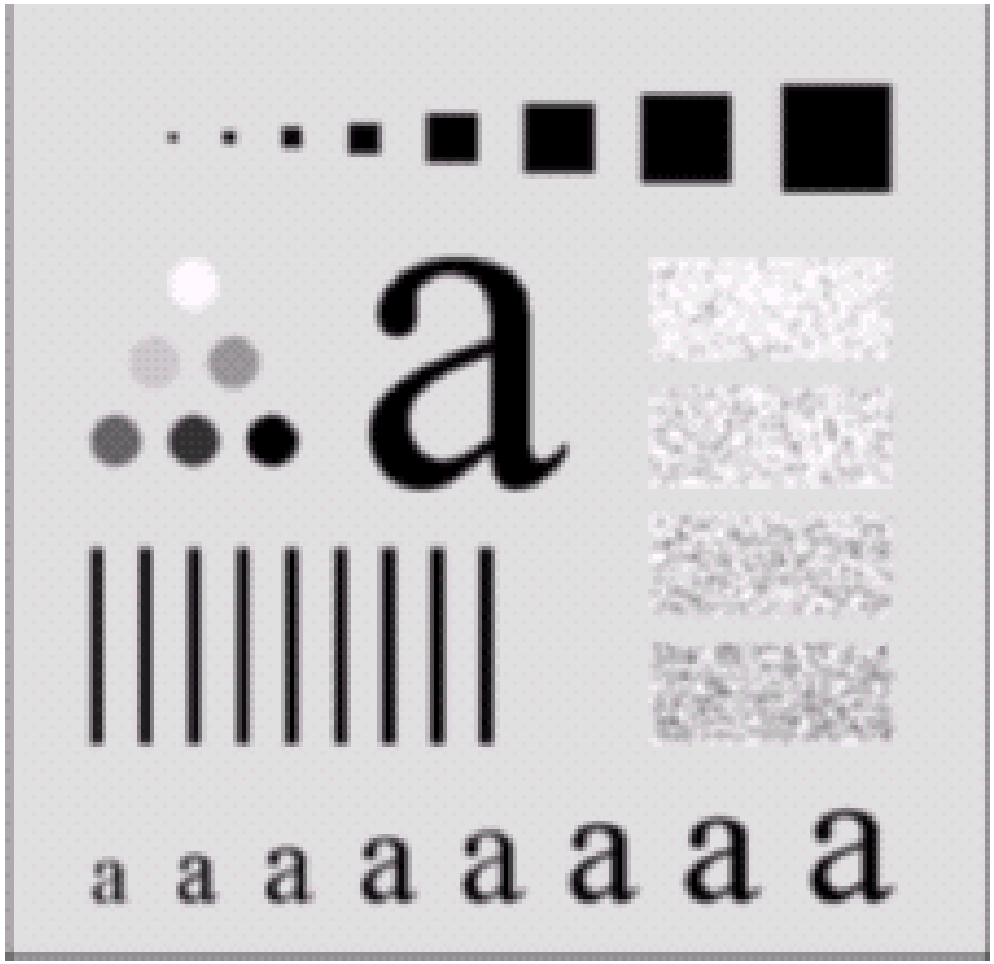
Image Smoothing Example



After filtering with an averaging filter of size 3×3

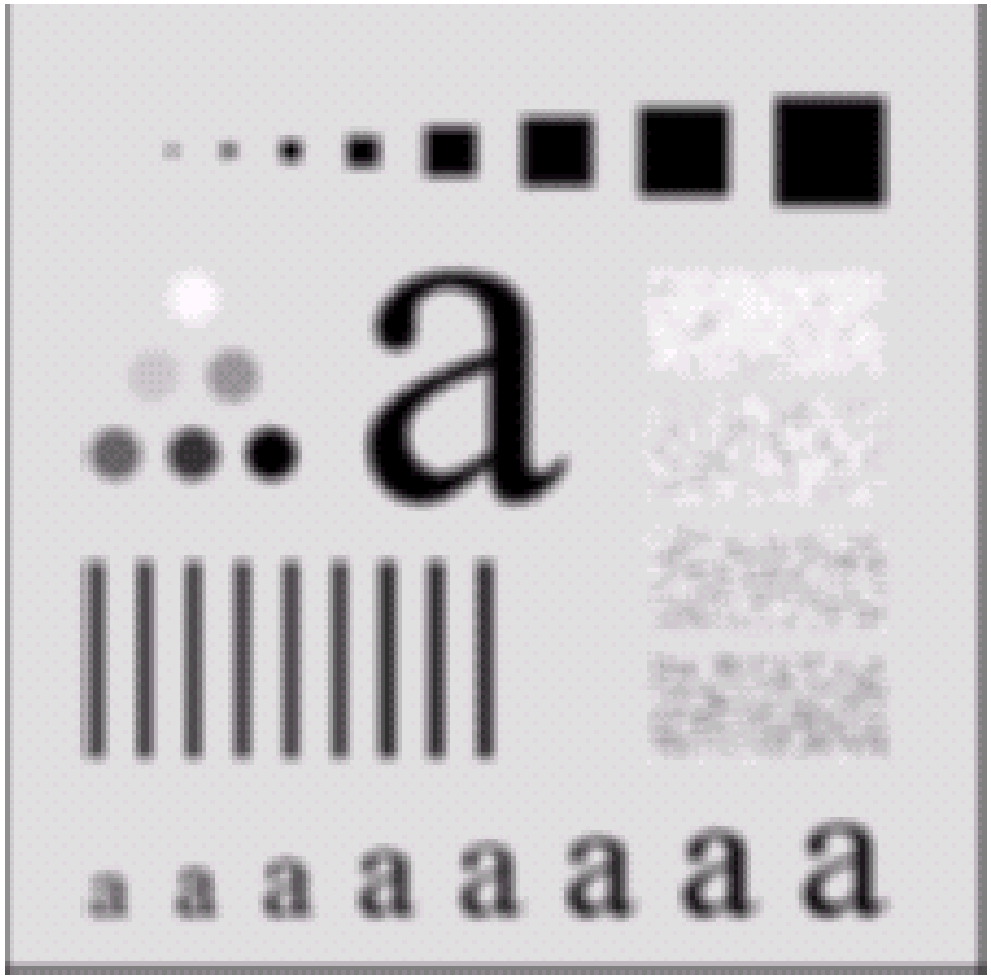
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Image Smoothing Example



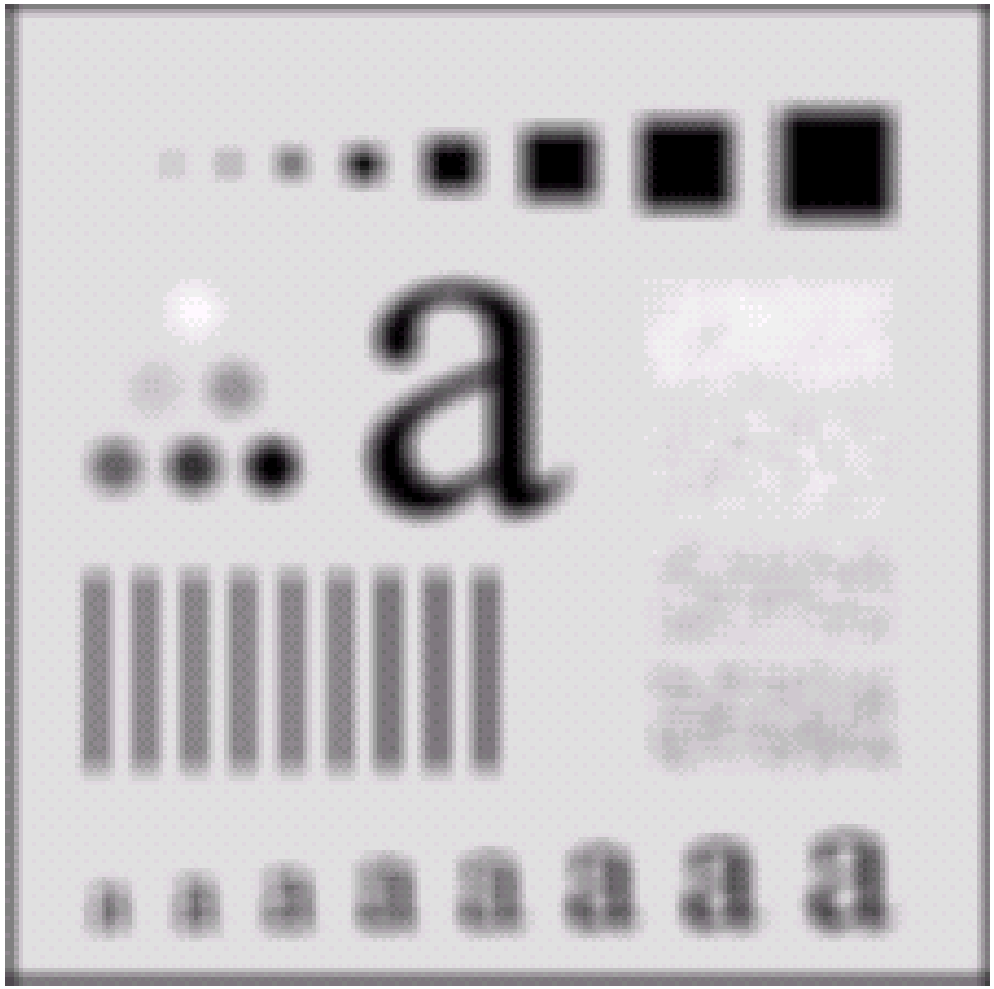
After filtering with an averaging filter of size 5×5

Image Smoothing Example



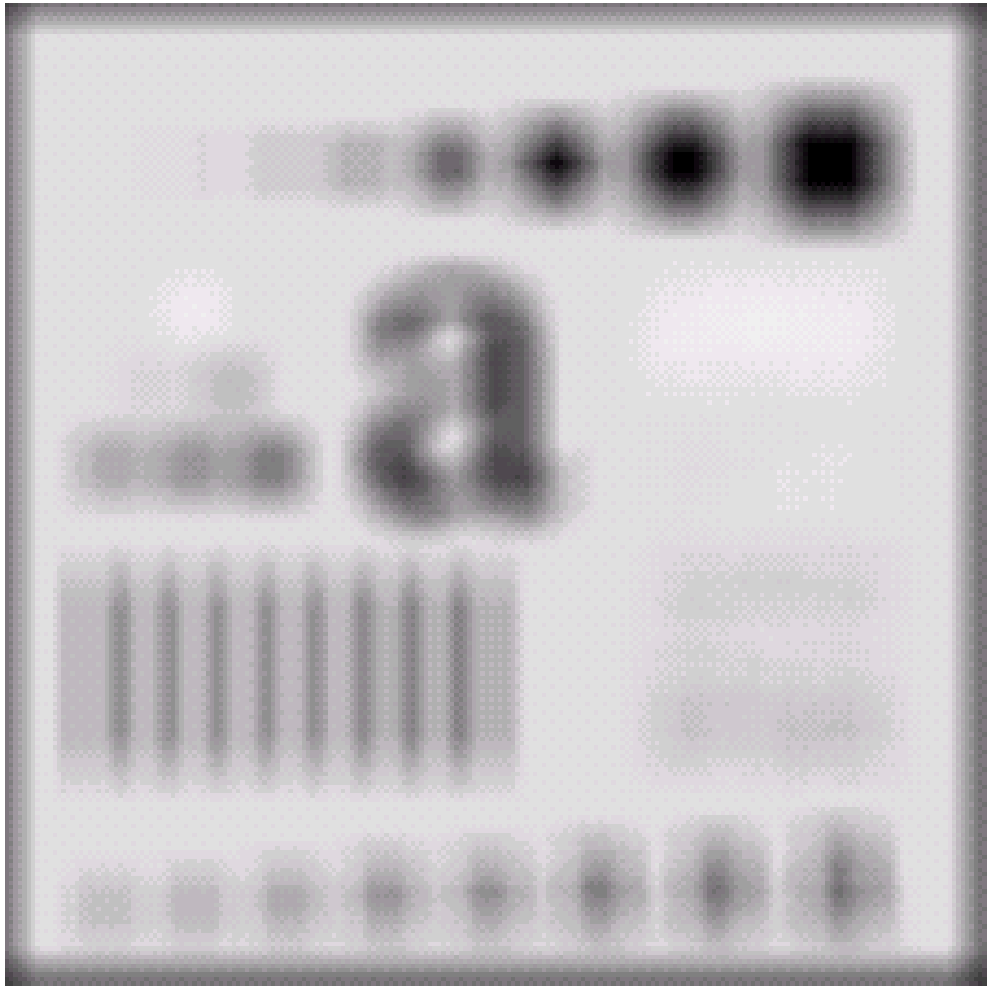
After filtering with an averaging filter of size 9×9

Image Smoothing Example



After filtering with an averaging filter of size 15×15

Image Smoothing Example



After filtering with an averaging filter of size 35×35

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.

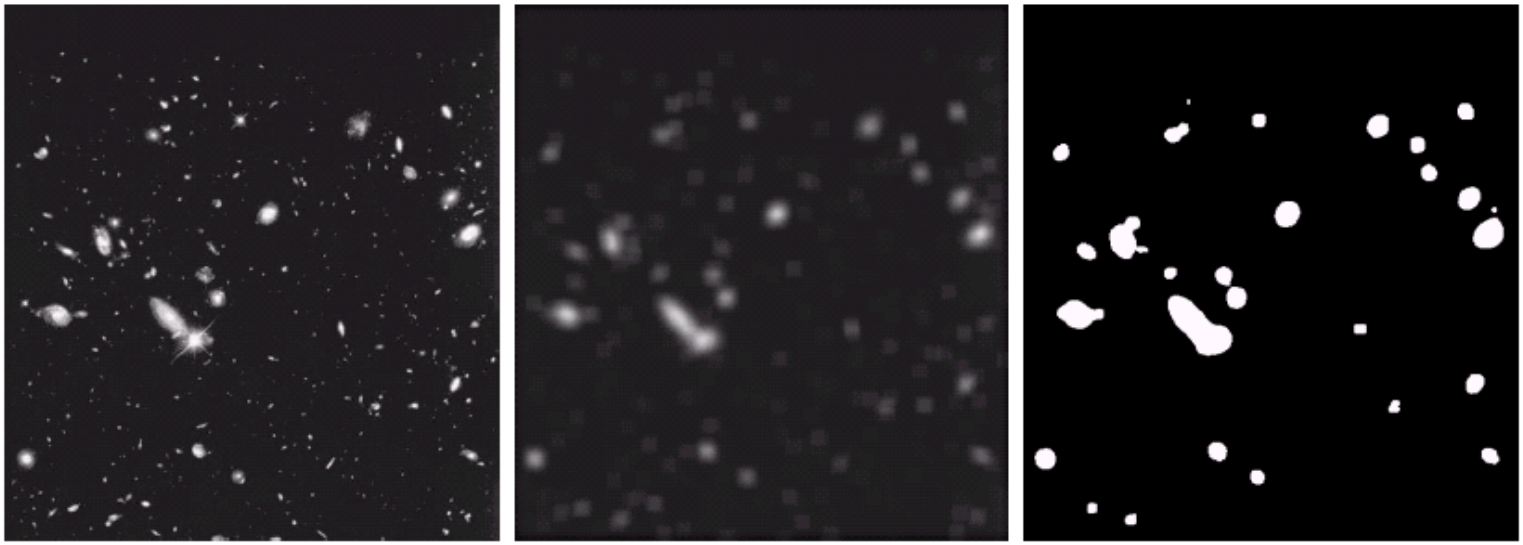
Often referred to as a *weighted averaging*.

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

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.



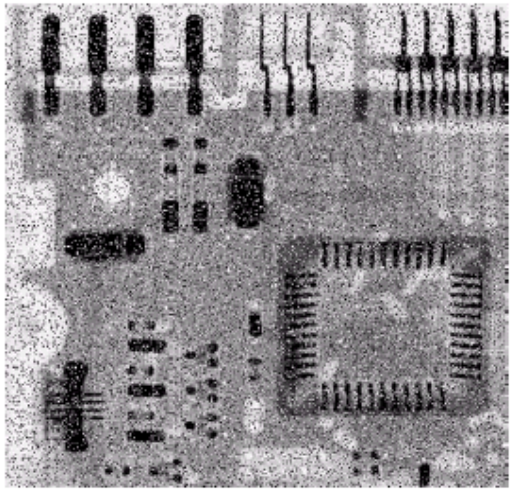
Original Image Smoothed Image Thresholded Image

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

Order Statistics Filter /Non-linear Filter

- **Median Filter :** The median value of a set of numbers is the midpoint value in that set (e.g. 17 is the median of the set [1, 5, 17, 28, 44]).
- **Max Filter :** Set the pixel value to the maximum in the neighbourhood.
- **Min Filter :** Set the pixel value to the minimum in the neighbourhood.

Averaging Filter Vs. Median Filter



Original Image With Noise

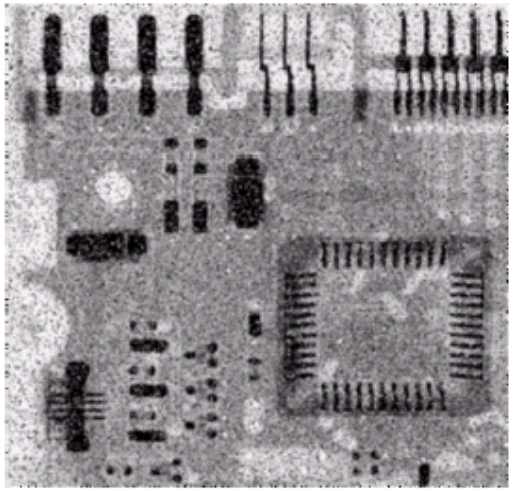


Image After Averaging Filter

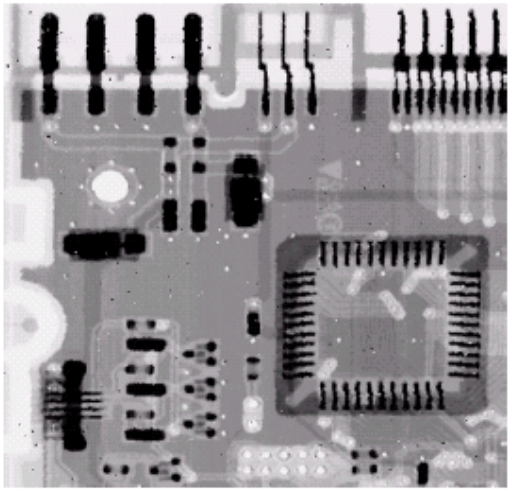
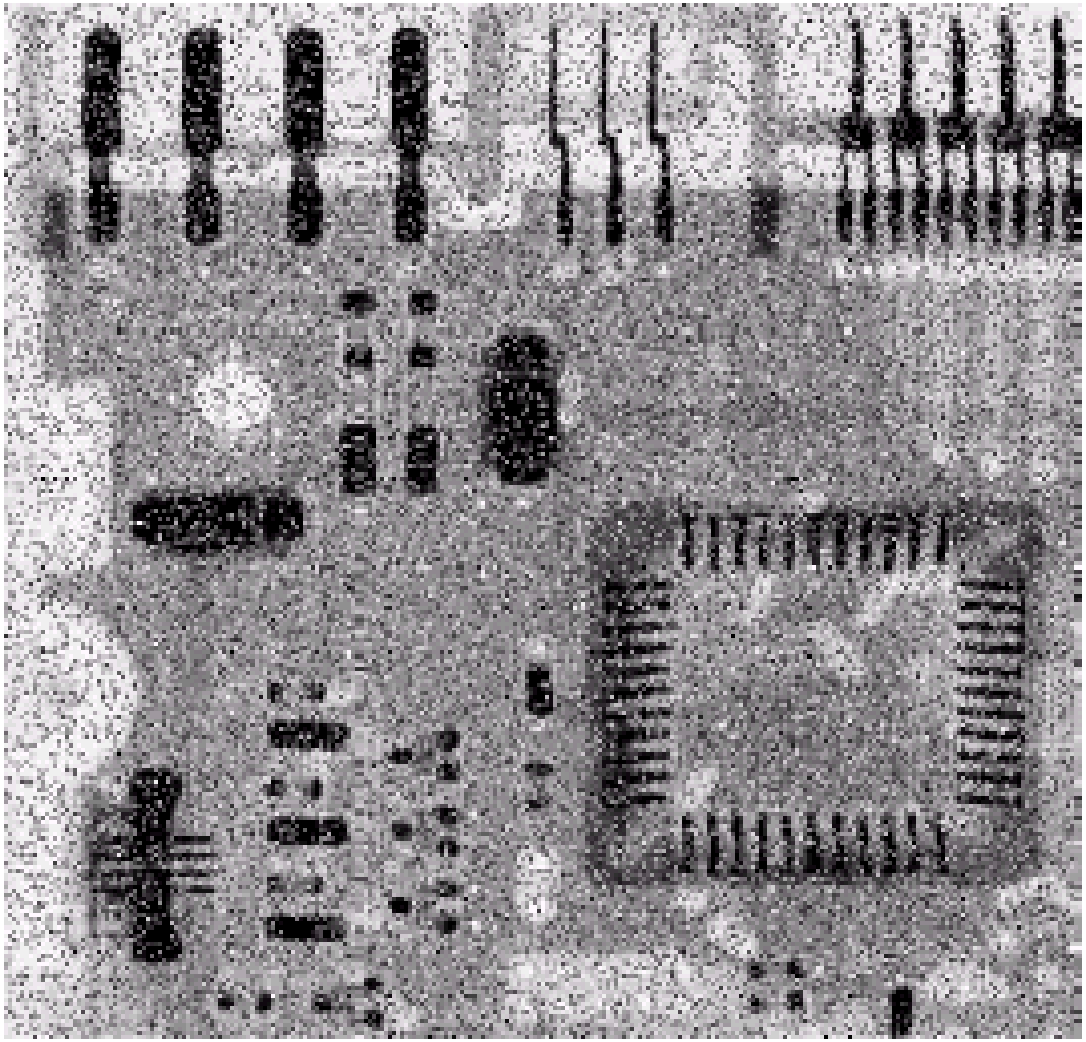


Image After Median Filter

Filtering is often used to remove noise from images.

Sometimes a median filter works better than an averaging filter.

Averaging Filter Vs. Median Filter Example



Original image with noise

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

Averaging Filter Vs. Median Filter Example

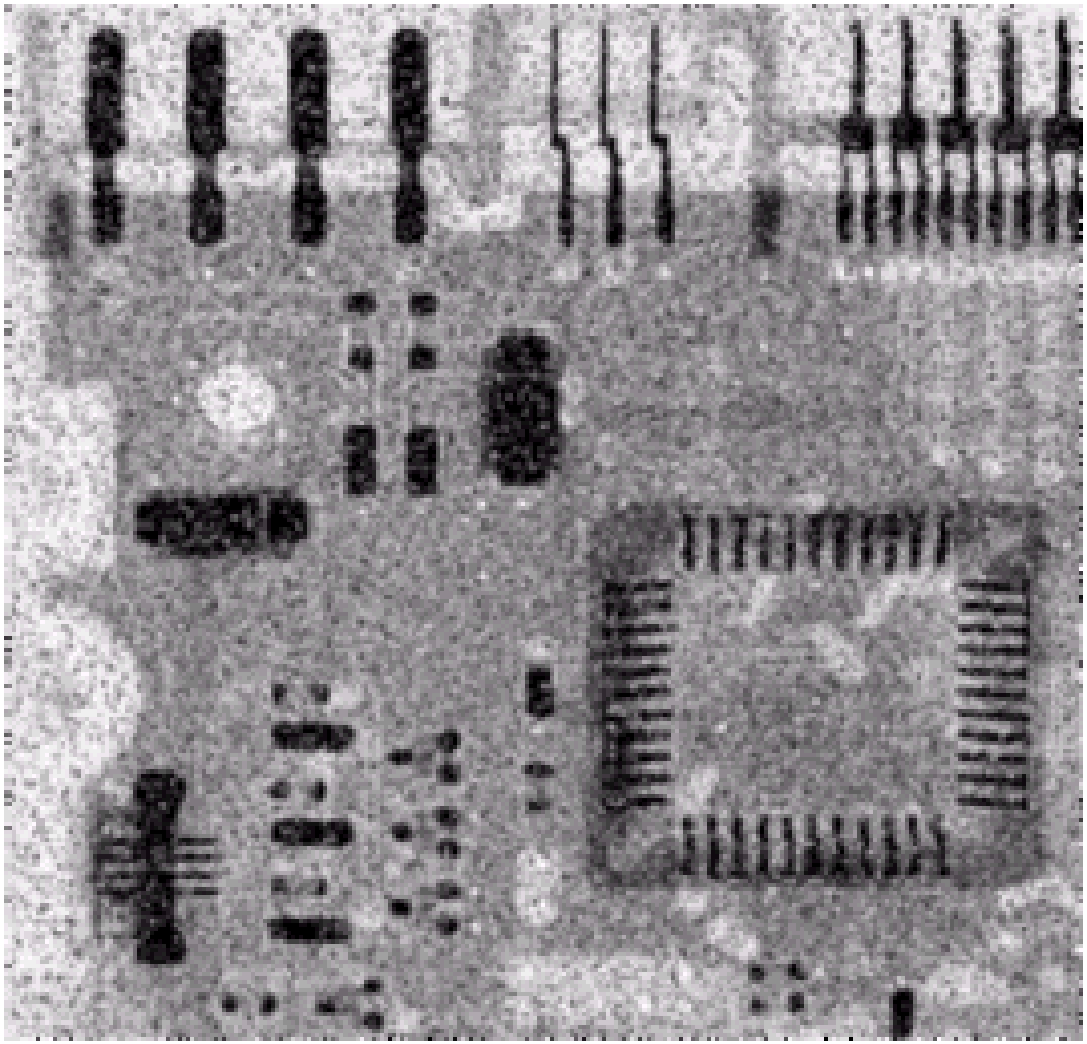


Image after Averaging filter

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

Averaging Filter Vs. Median Filter Example

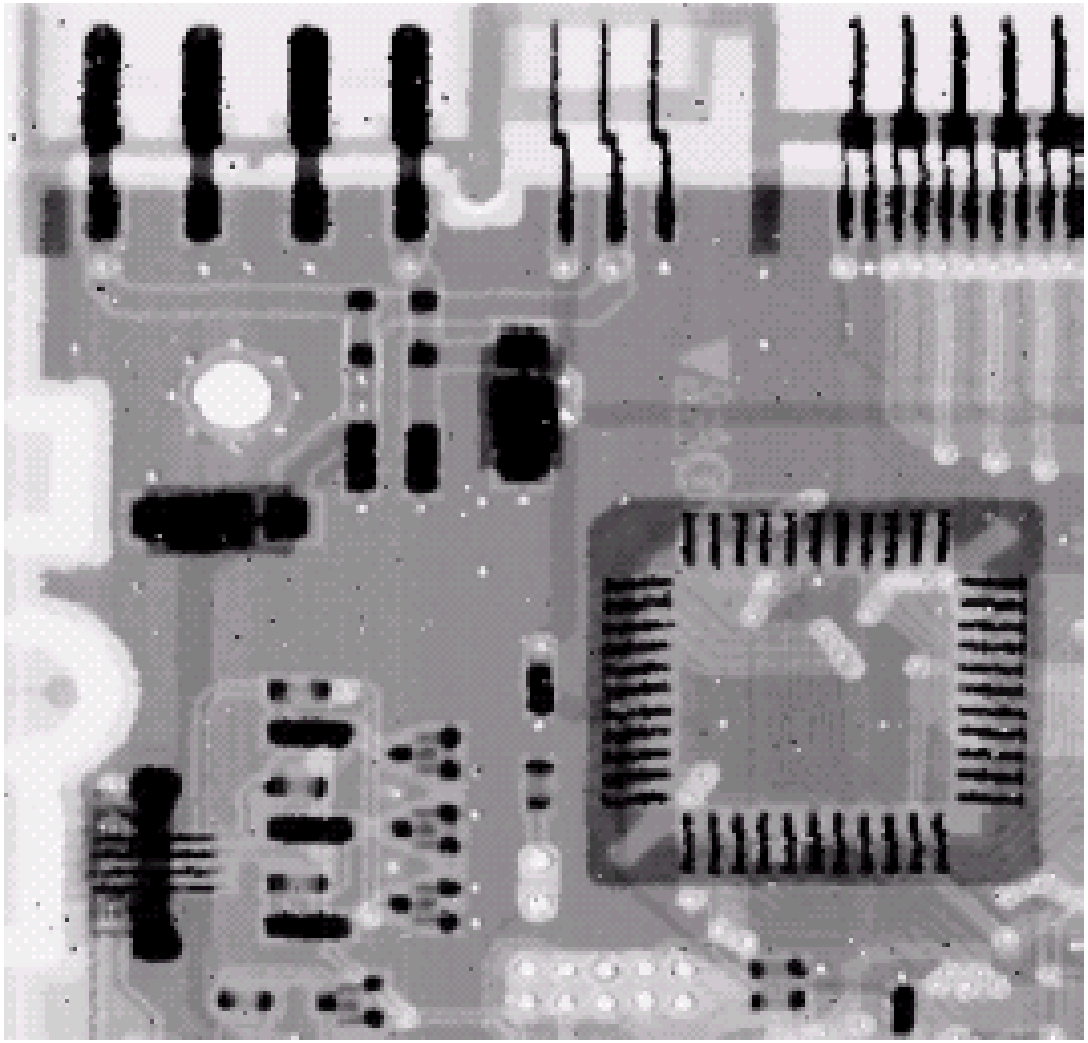
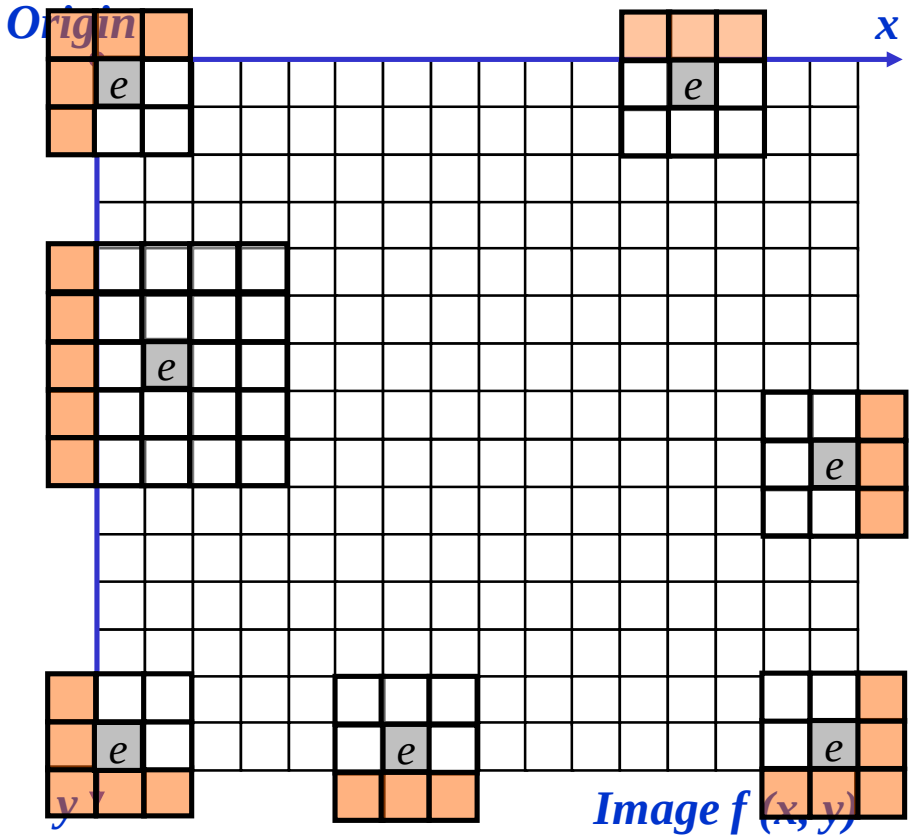


Image after Median filter

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

At The Edges!

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



At The Edges!

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

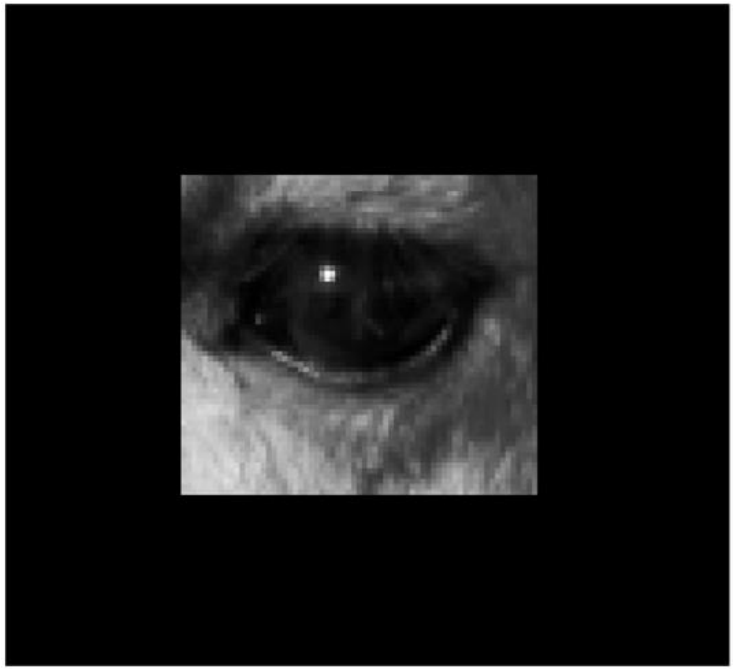
  Typically with either all white or all black pixels.

- Replicate border pixels

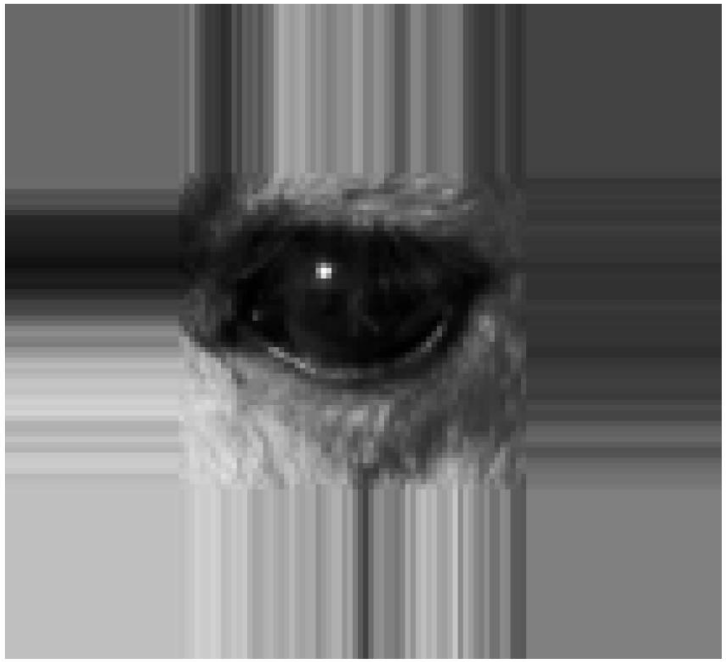
- Allow pixels *wrap around* the image

  Can cause some strange image artifacts.

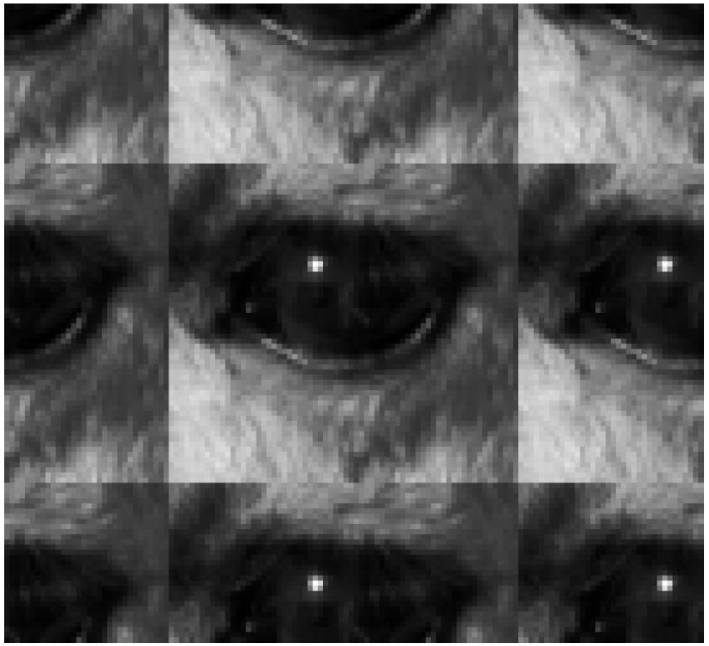
Pad the Image



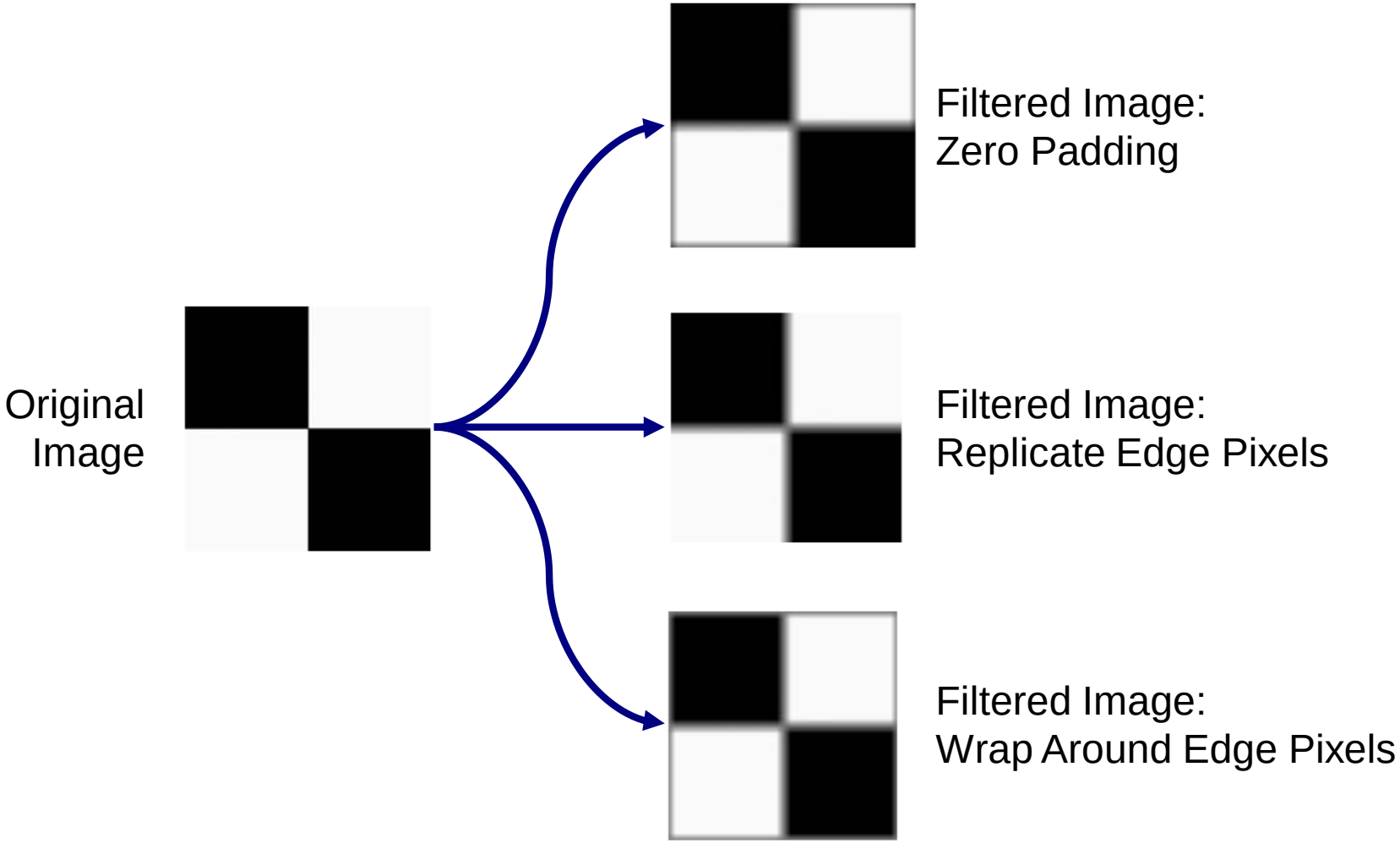
Replicate Border Pixels



Allow Pixels *Wrap Around* the Image



Strange Things Happen At The Edges!



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

Edge Sharpening

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

Sharpening Spatial Filters

We have seen that smoothing filters remove noises but also remove fine detail.

Sharpening spatial filters seek to highlight fine detail by

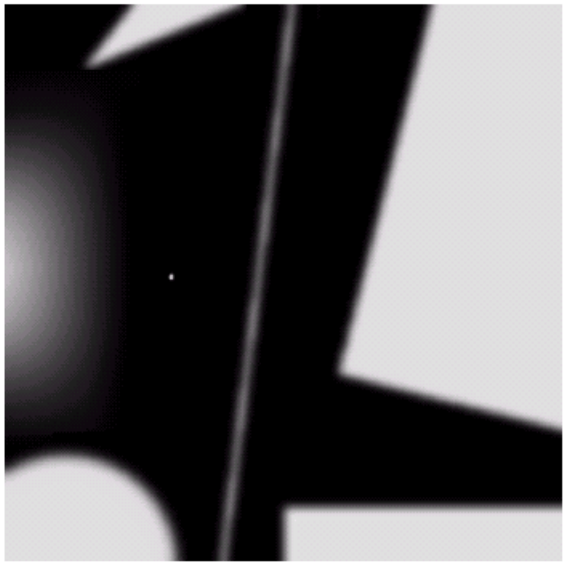
- Removing blurring from images
- Highlighting edges

Sharpening filters are based on spatial differentiation.

Spatial Differentiation

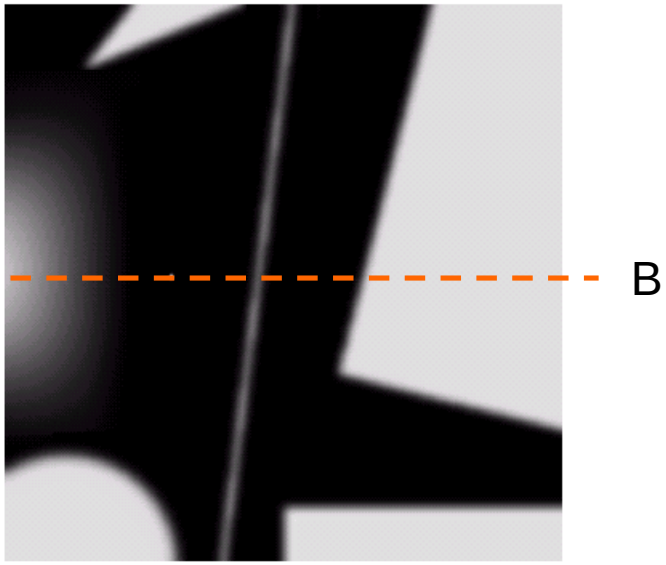
Differentiation measures the rate of change of a function.

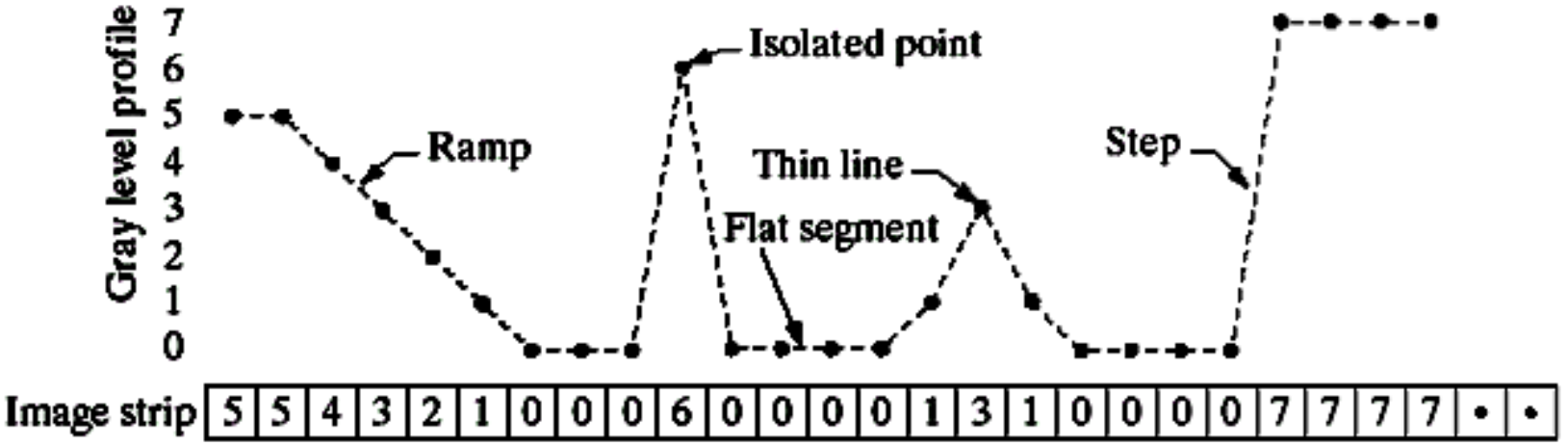
Let's consider a simple 1 dimensional example.



This image contains various solid objects, a line, and a single noise point.

Spatial Differentiation





1st Order Derivative

The 1st order derivative of a digital function is defined in terms of difference. There are various ways to define these differences.

We require that any definition we use for a 1st order derivative (1) must be zero in flat segments; (2) must be zero at the onset of a gray level step or ramp; and (3) must be nonzero along ramp.

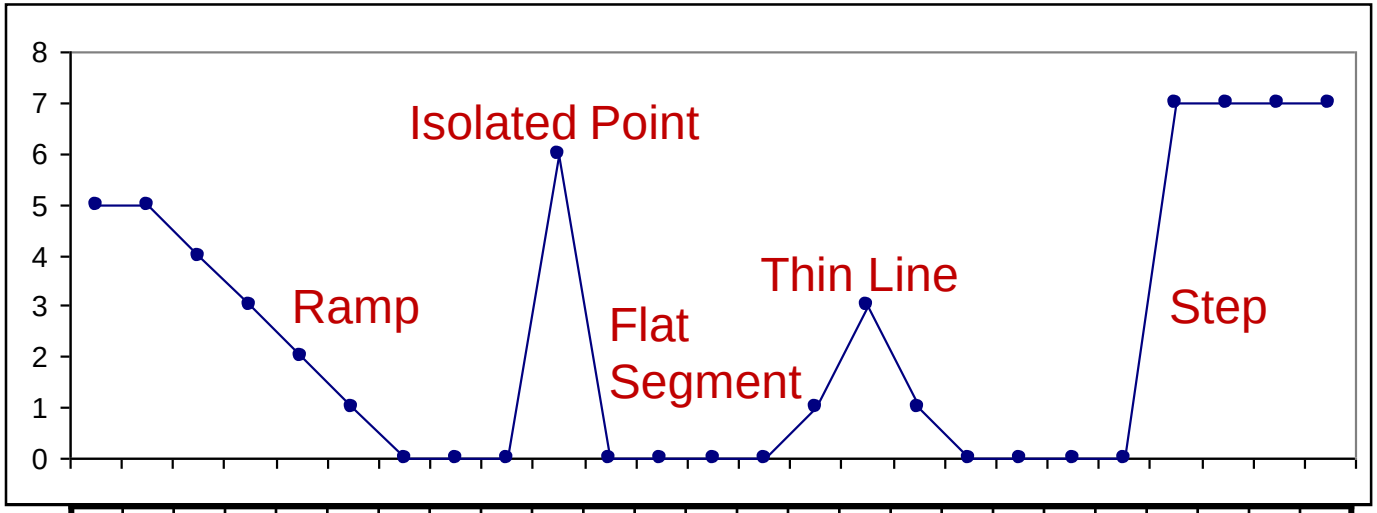
1st Order Derivative

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

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

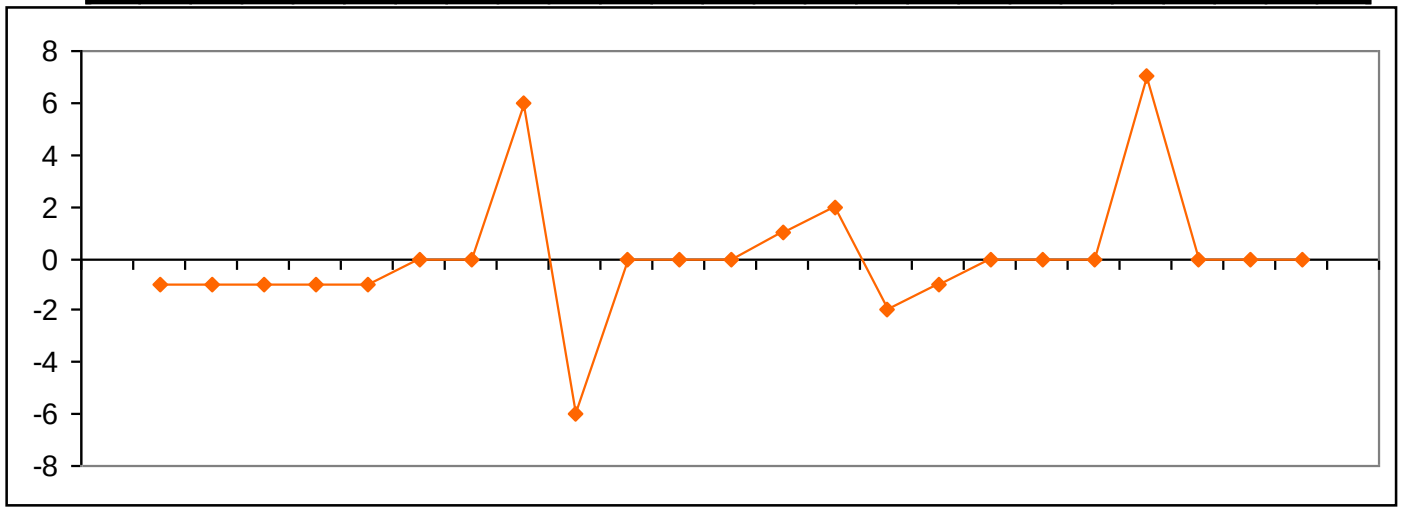
Difference between subsequent values and measures the rate of change of the function.

1st Order 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

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 Order Derivative Filtering

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 Order Derivative Filtering

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 Order Derivative Filtering

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

It 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

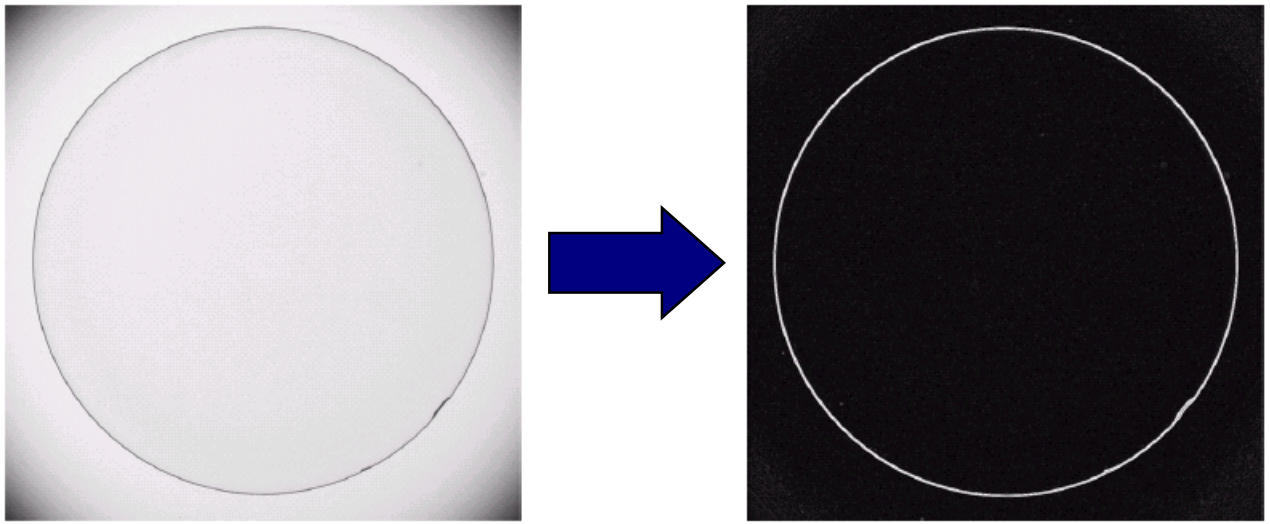
Based on the previous equations we can derive the *Sobel Operators.*

-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



Sobel filters are typically used for edge detection.

Prewitt Operators

Another type of edge detection operator, named *Prewitt Operators.*

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

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

2^{nd} Order Derivative

The 2^{nd} order derivative of a digital function is defined in terms of difference. There are various ways to define these differences.

We require that any definition we use for a 2^{nd} order derivative (1) must be zero in flat areas; (2) must be nonzero at the onset and end of a gray level step or ramp; and (3) must be zero along ramps of constant slop.

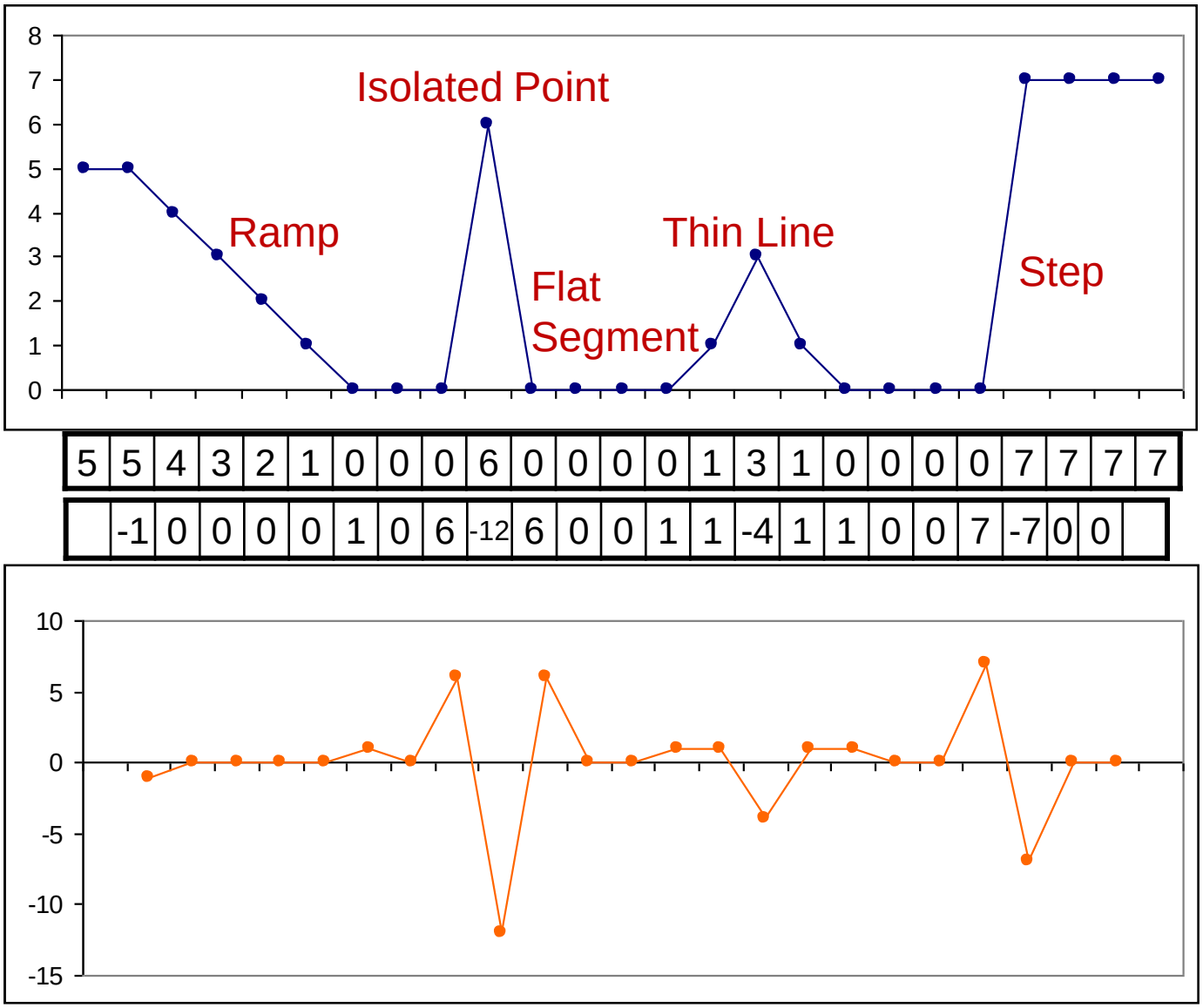
2nd Order Derivative

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

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

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

2nd Order Derivative



2nd Order Derivative Filtering

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

- Stronger response to fine detail, such as thin lines and isolated points
- Simpler implementation

The first sharpening filter is Laplacian filter.

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

The Laplacian

The Laplacian is defined as follows:

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

where the partial 2nd order derivative in the x direction is defined as follows:

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

and in the y direction as follows:

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

The Laplacian

So, the Laplacian can be given as follows:

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

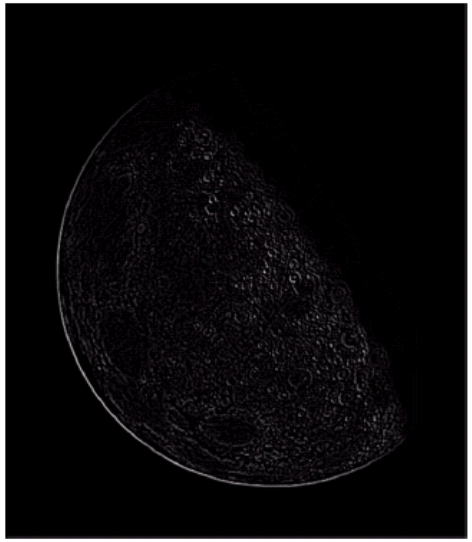
We can easily build a filter based on this

0	1	0
1	-4	1
0	1	0

The Laplacian

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

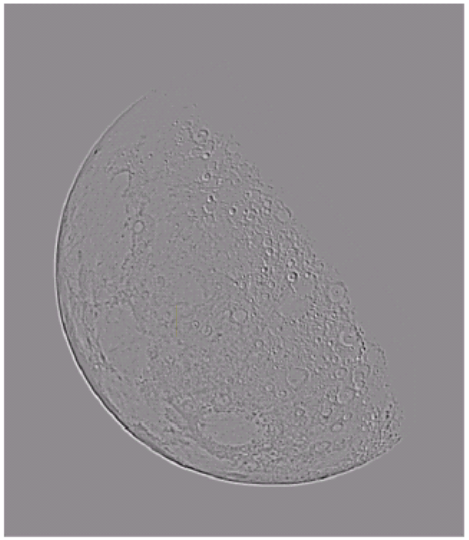
But That Is Not Very Enhanced!

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.

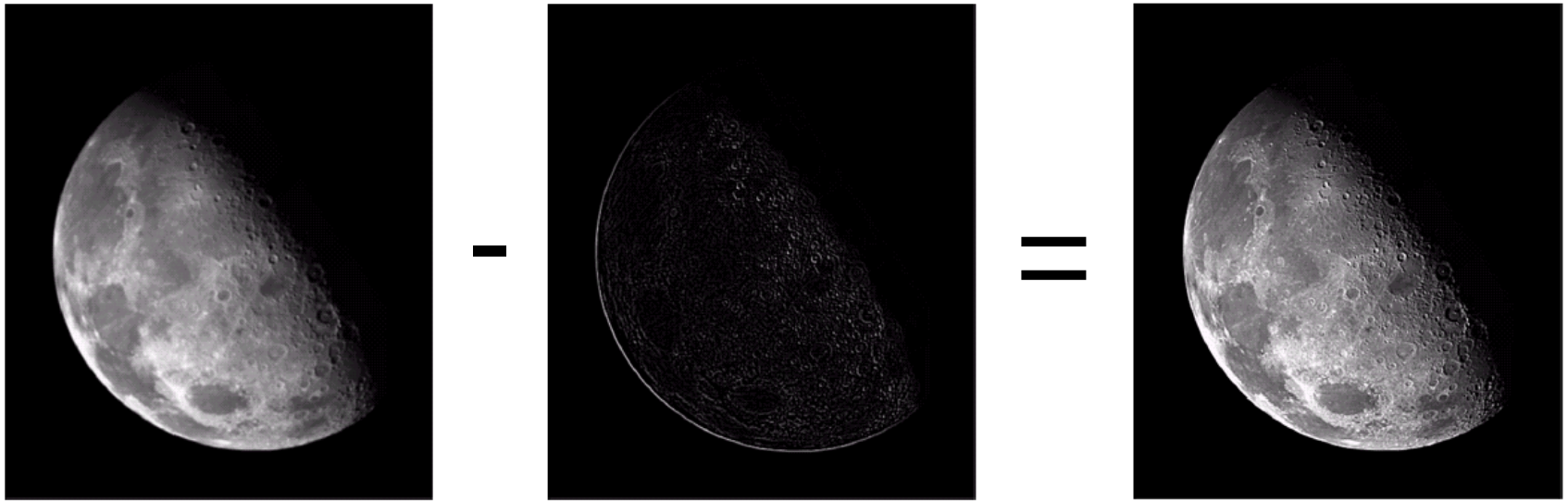
$$g(x,y) = f(x,y) - \nabla^2 f$$



Laplacian
Filtered Image
Scaled for Display

This is called
Unsharp Masking

Laplacian Image Enhancement



Original Image

Laplacian Filtered Image

Sharpened Image

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

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

Laplacian Image Enhancement

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

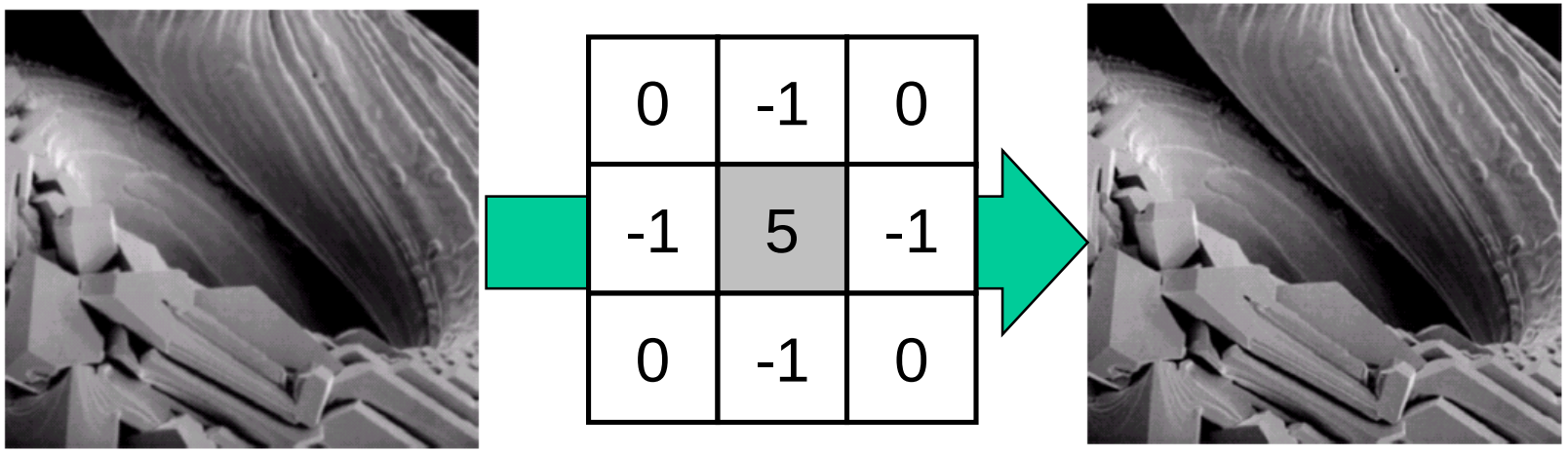
Simplified Image Enhancement

The entire enhancement can be combined into a single filtering operation

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

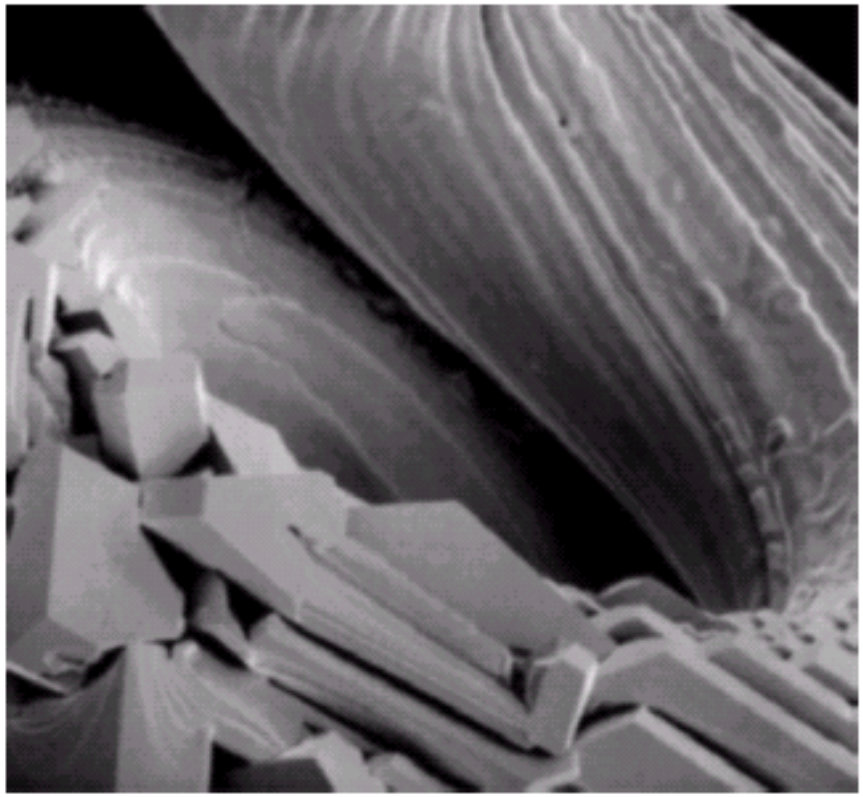
Simplified Image Enhancement

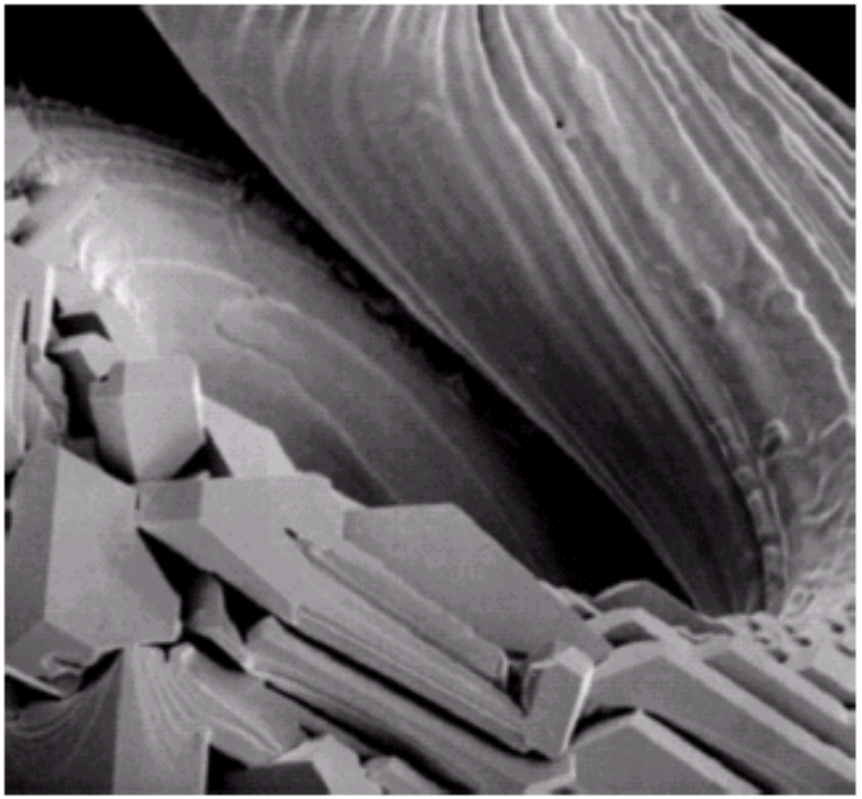
This gives us a new filter which does the whole job in one step.



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

Simplified Image Enhancement





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

Variants On The Simple Laplacian

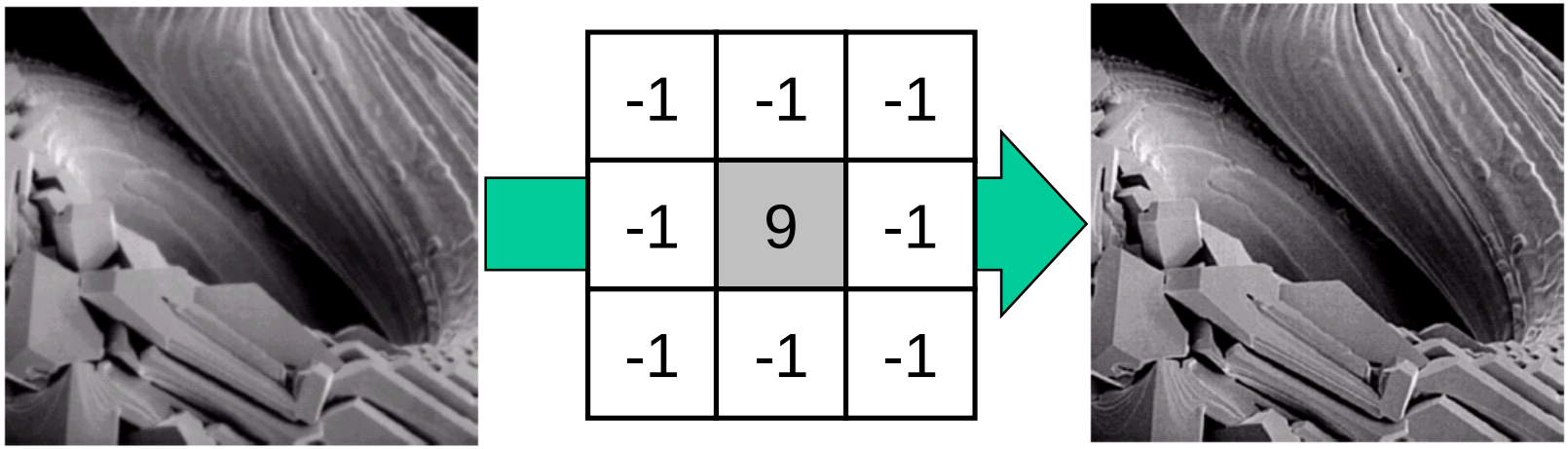
There are lots of slightly different versions of the Laplacian that can be used:

0	1	0
1	-4	1
0	1	0

Simple Laplacian

1	1	1
1	-8	1
1	1	1

Variant of Laplacian



Comparison of 1st Order and 2nd Order Derivatives

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

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)

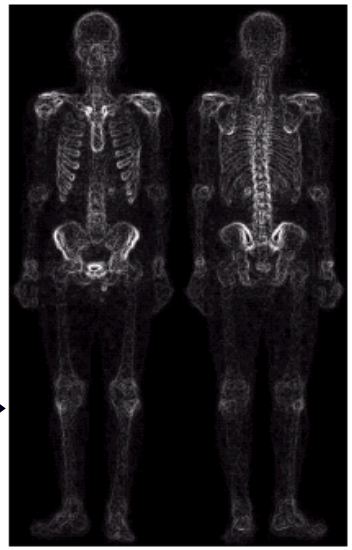
Laplacian filter of bone scan (a)

(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

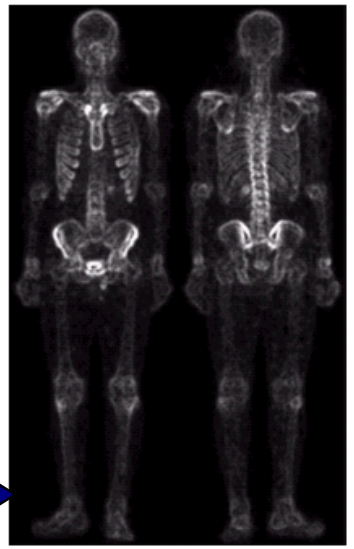
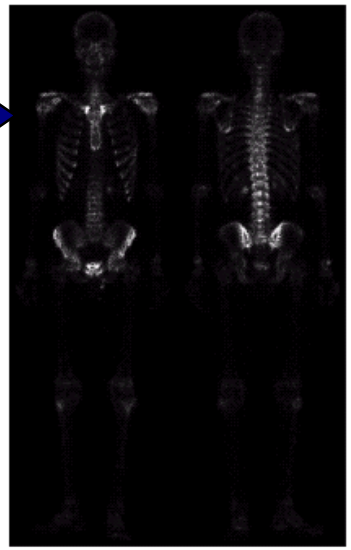


Image (d) smoothed with a 5 × 5 averaging filter

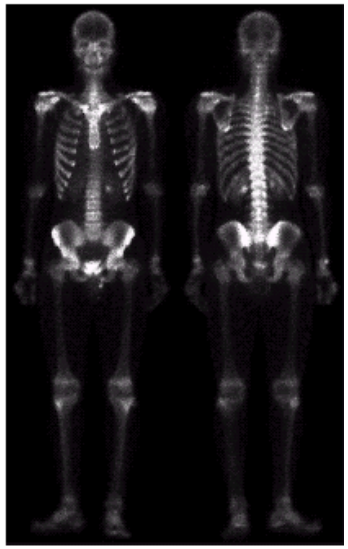
The product of (c) and (e) which will be used as a mask

(e)



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

(f)



Note the dominance of the strong edges and the relative lack of visible noises

Result of applying a power-law transformation to (g)

(g)

(h)

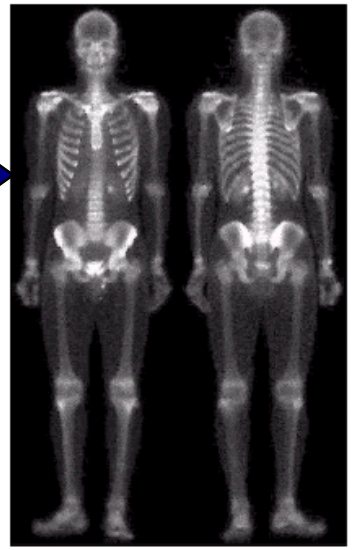
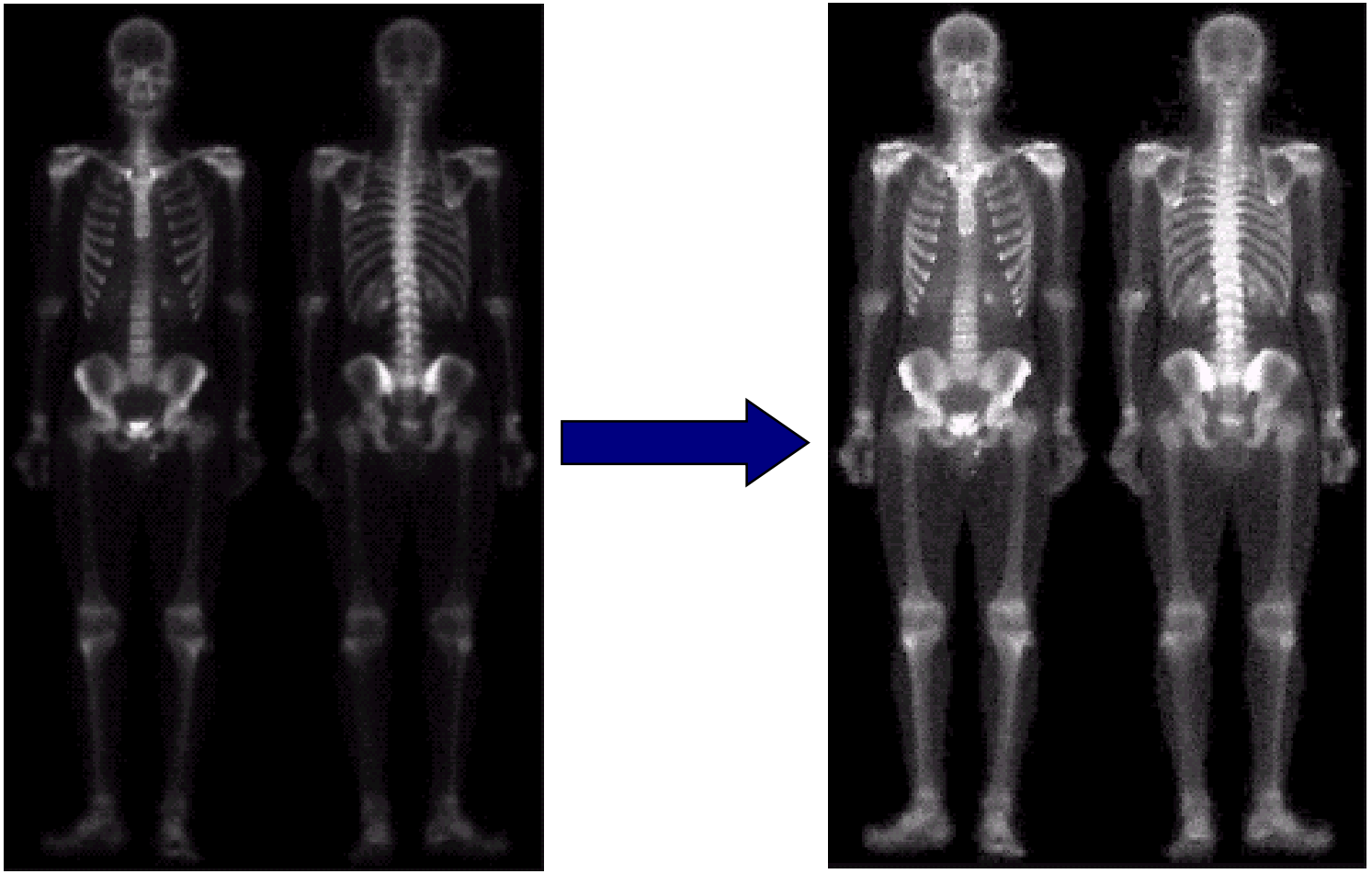


Image taken from Gonzalez & Woods, Digital Image Processing (2002)

Combining Spatial Enhancement Methods

Compare the original and final images:



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