Image Enhancement
(Histogram Processing)

Subhadip Basu, Ph.D.
Department of Computer Science and Engineering,
Jadavpur University

Contents

Over the next few lectures we will look at image enhancement techniques working in the spatial domain:

- – What is image enhancement?
- – Different kinds of image enhancement
- – Histogram processing
- – Point processing
- – Neighbourhood operations

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

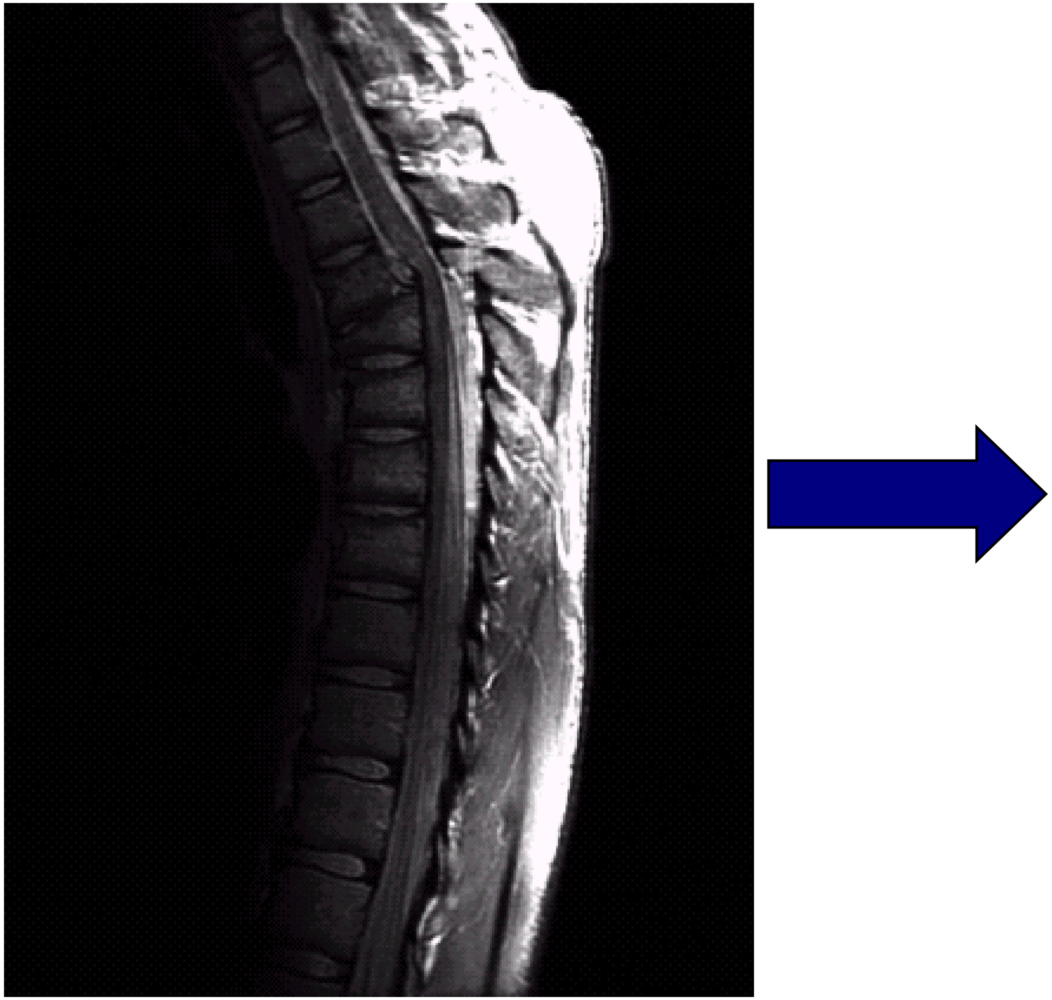
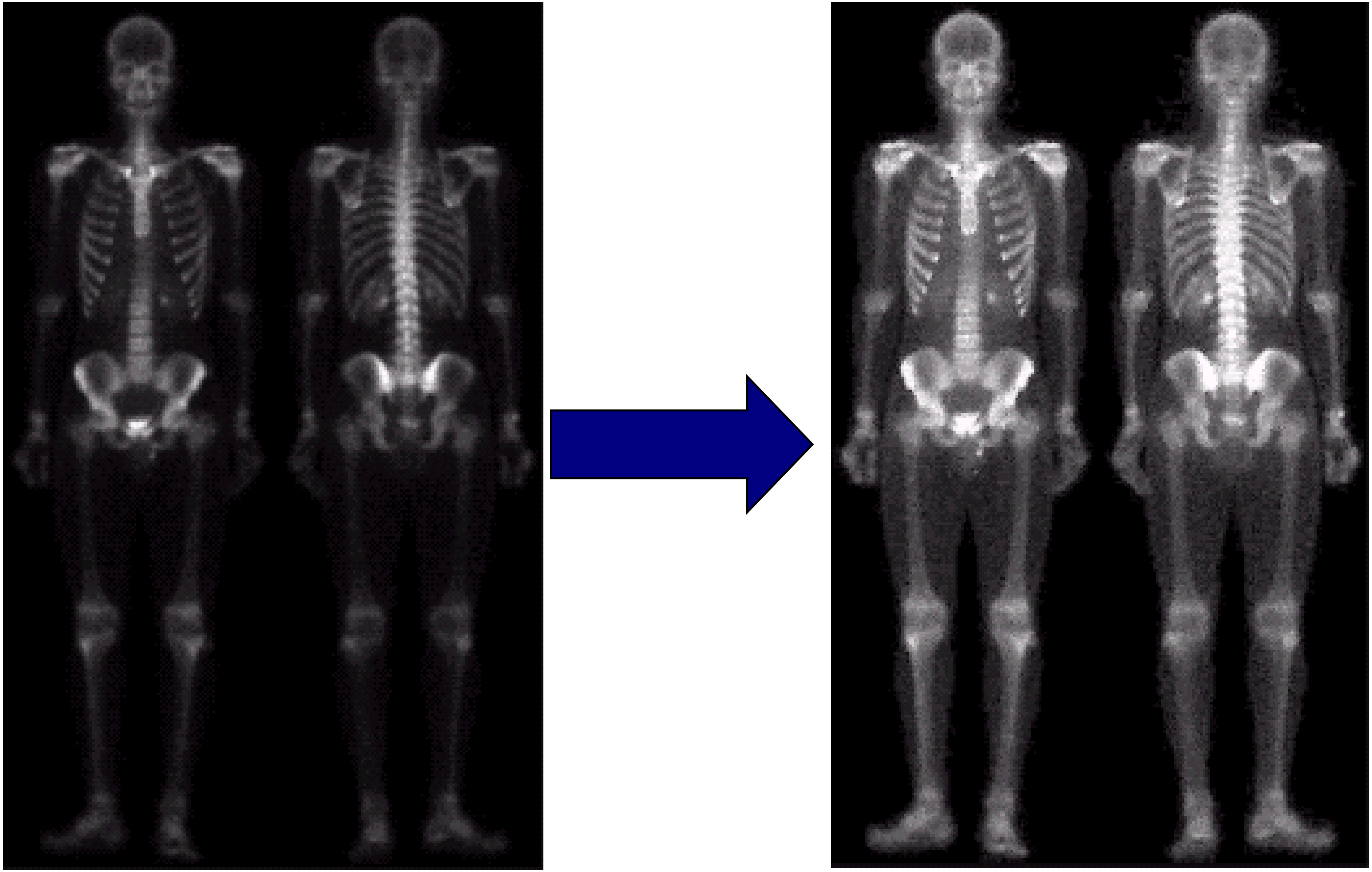


Image Enhancement Examples (cont…)



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

Image Enhancement Examples (cont…)

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

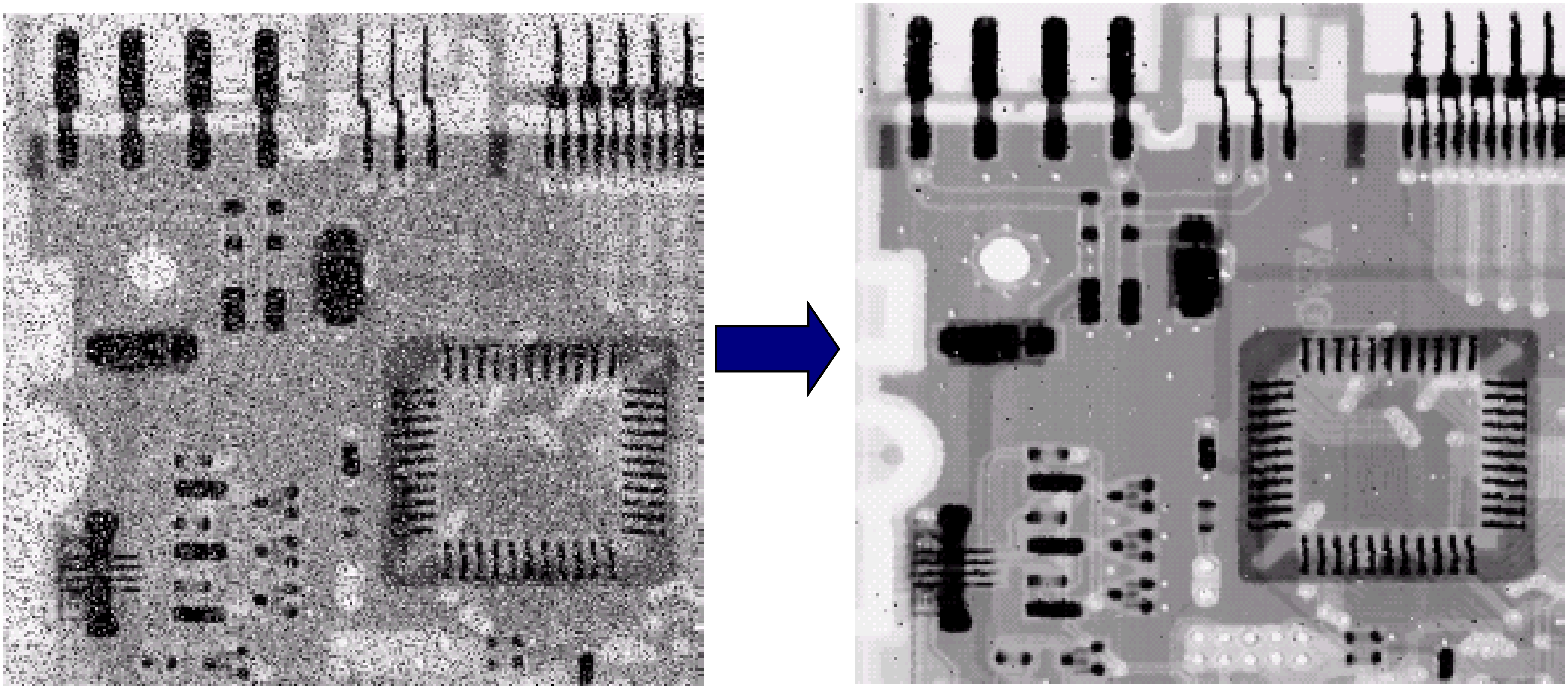
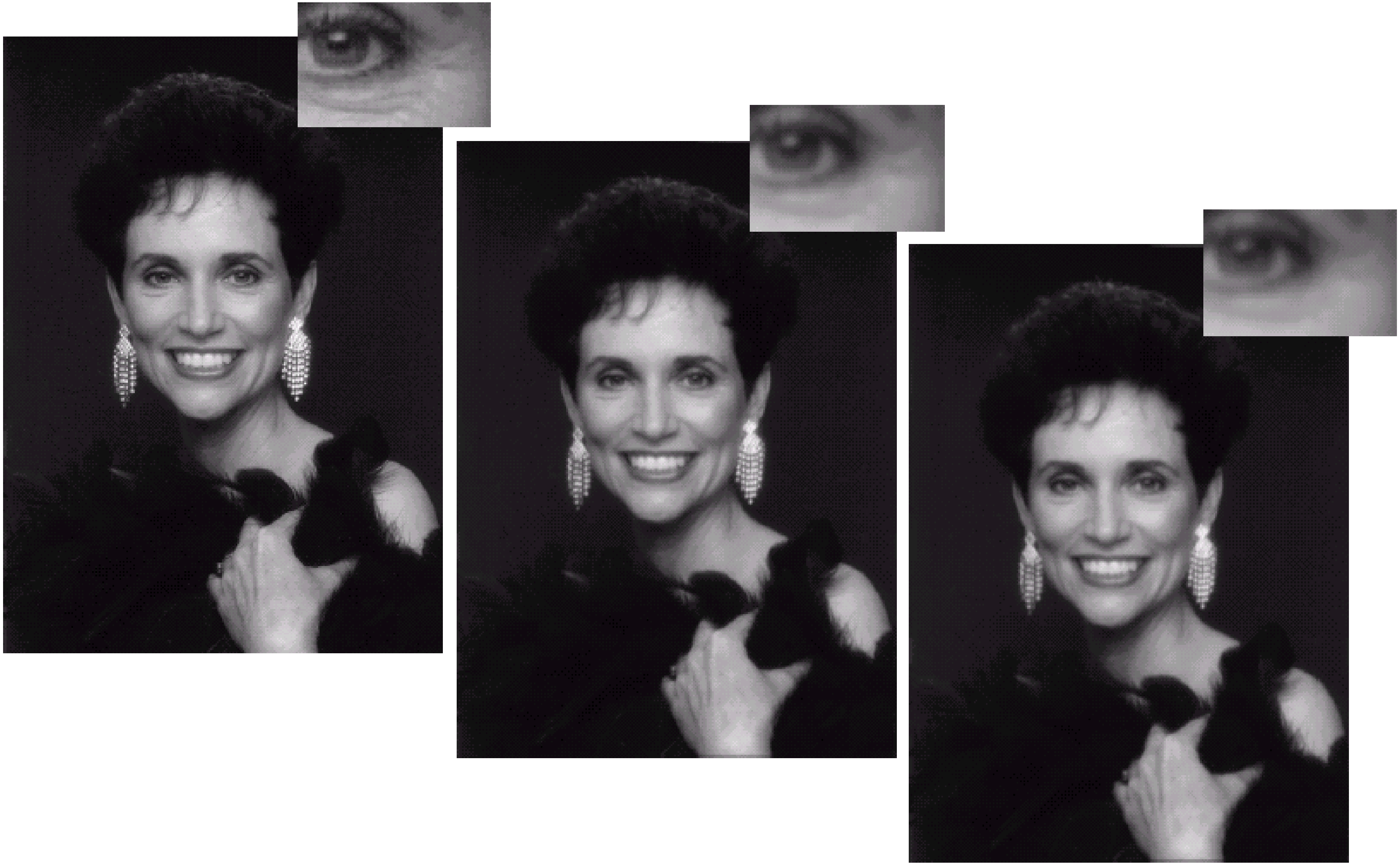


Image Enhancement Examples (cont…)



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

Spatial & Frequency Domains

There are two broad categories of image enhancement techniques

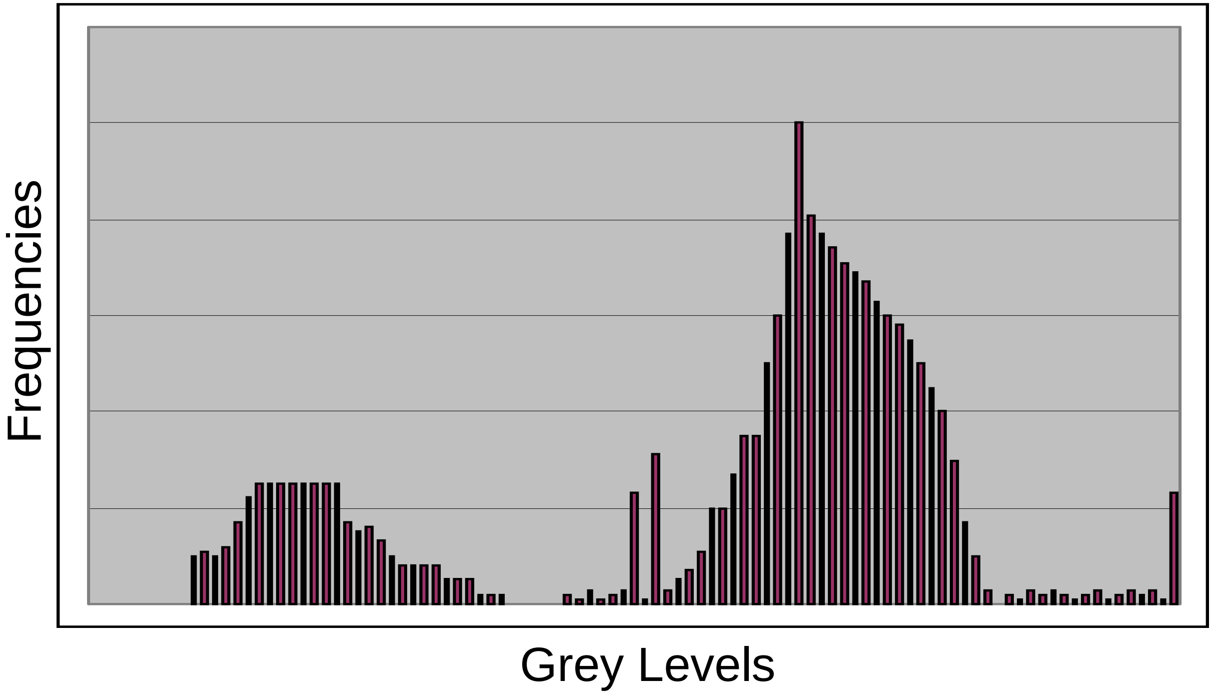
- Spatial domain techniques
  - Direct manipulation of image pixels
- Frequency domain techniques
  - Manipulation of Fourier transform or wavelet transform of an image

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

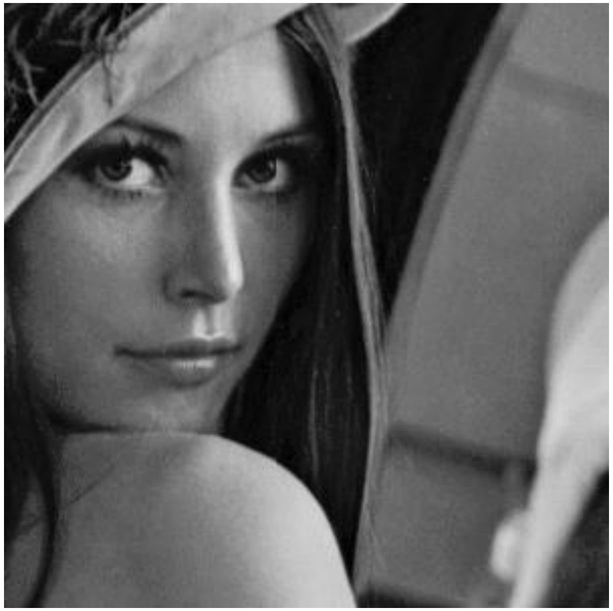


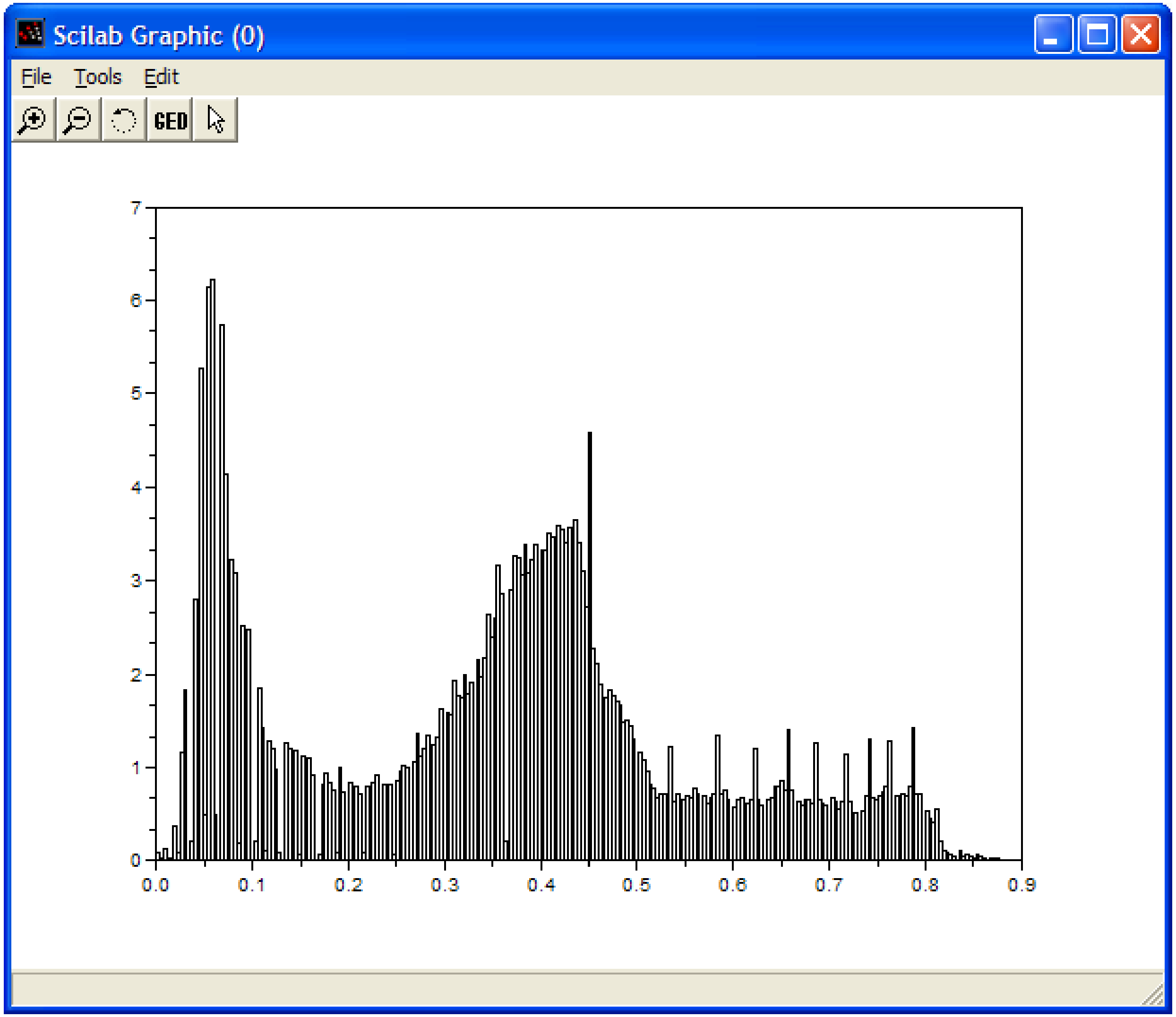
Histogram Examples

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

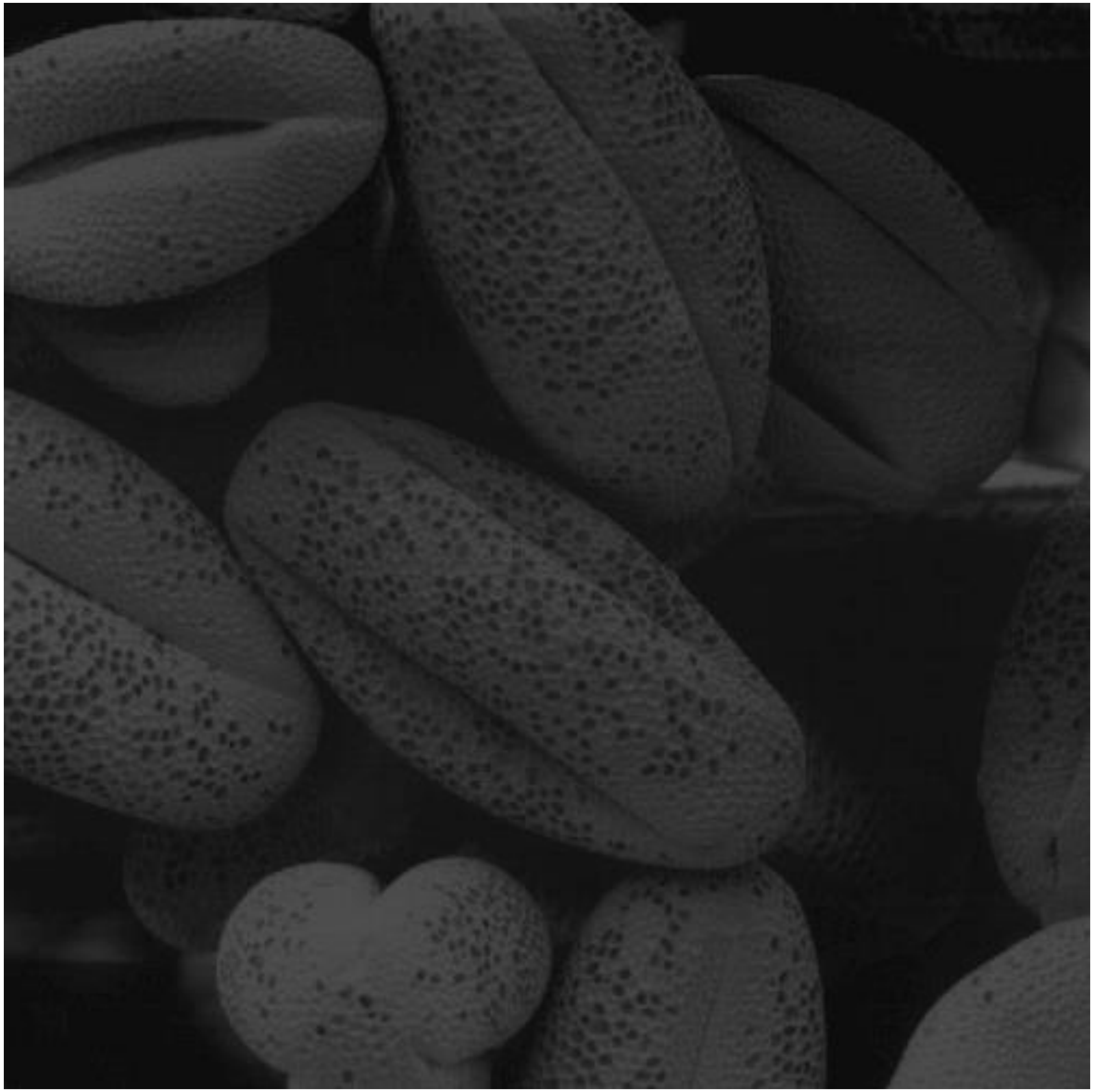
Histogram Examples (cont…)

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





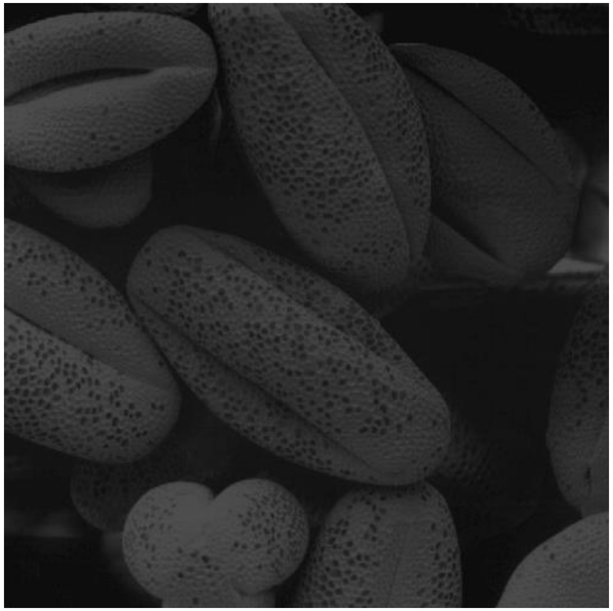
Histogram Examples (cont…)

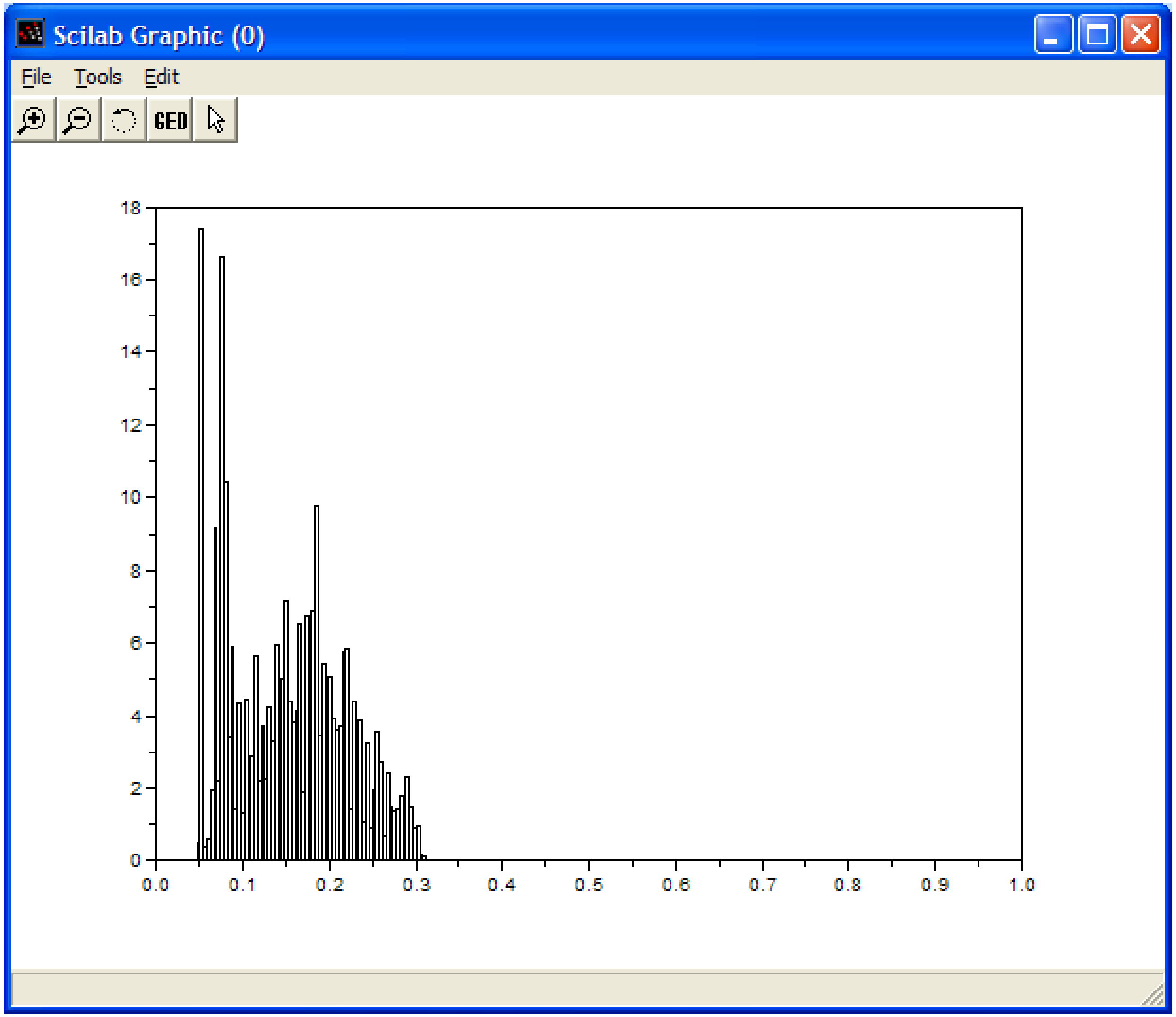


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

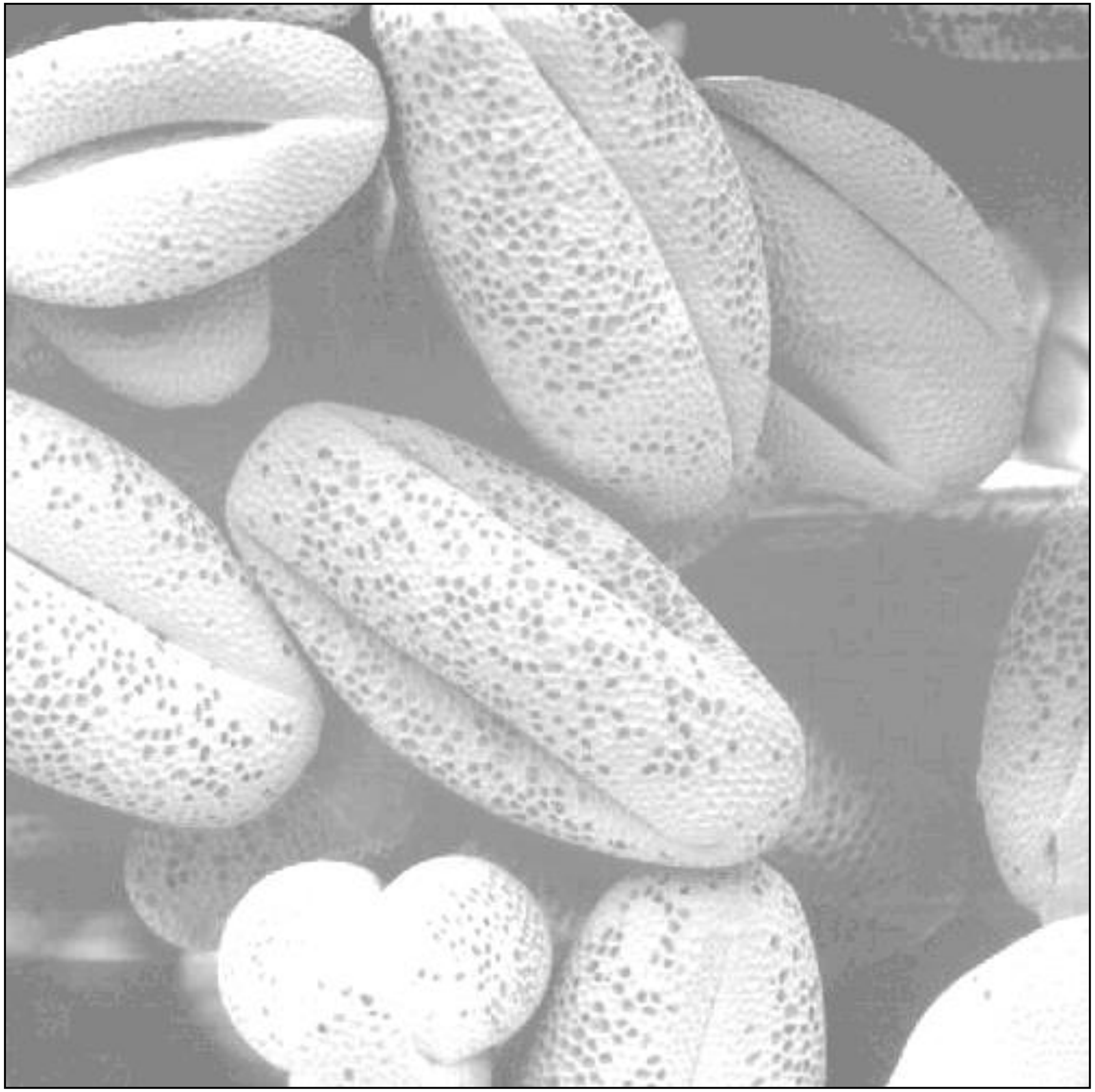
Histogram Examples (cont…)

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





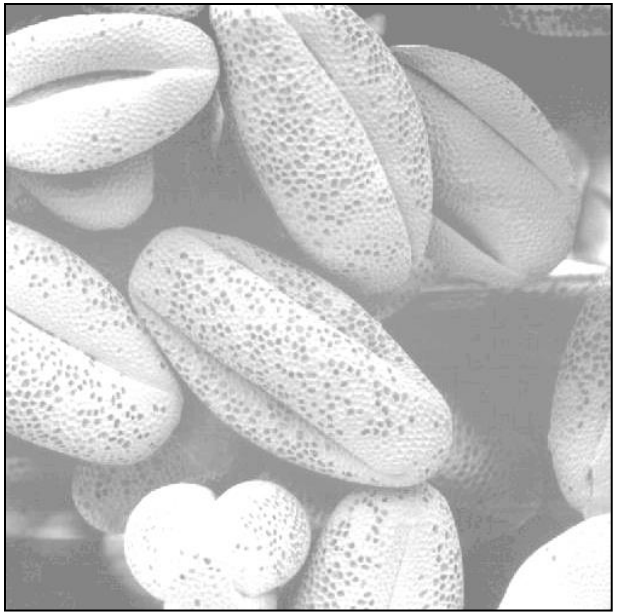
Histogram Examples (cont…)

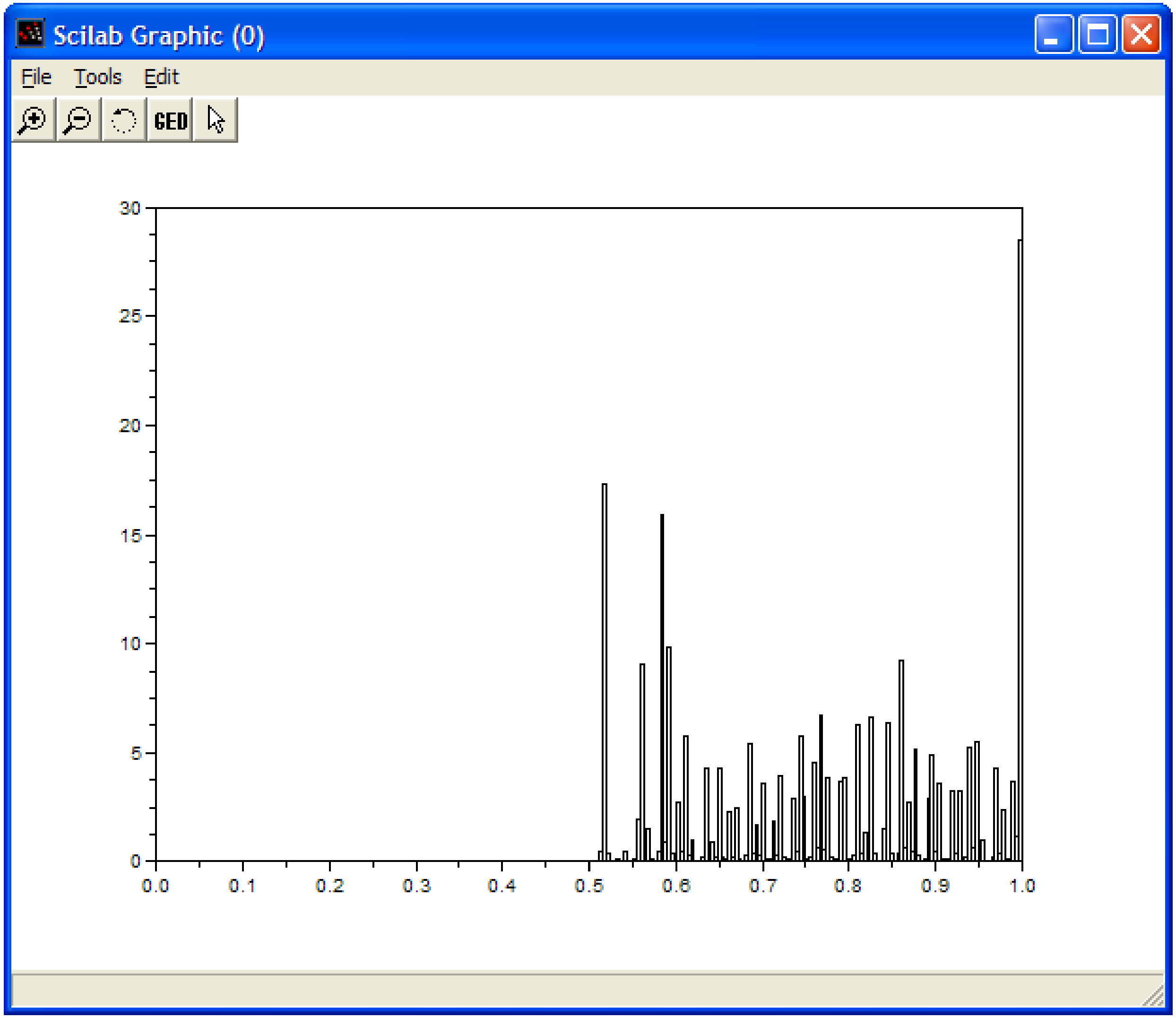


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

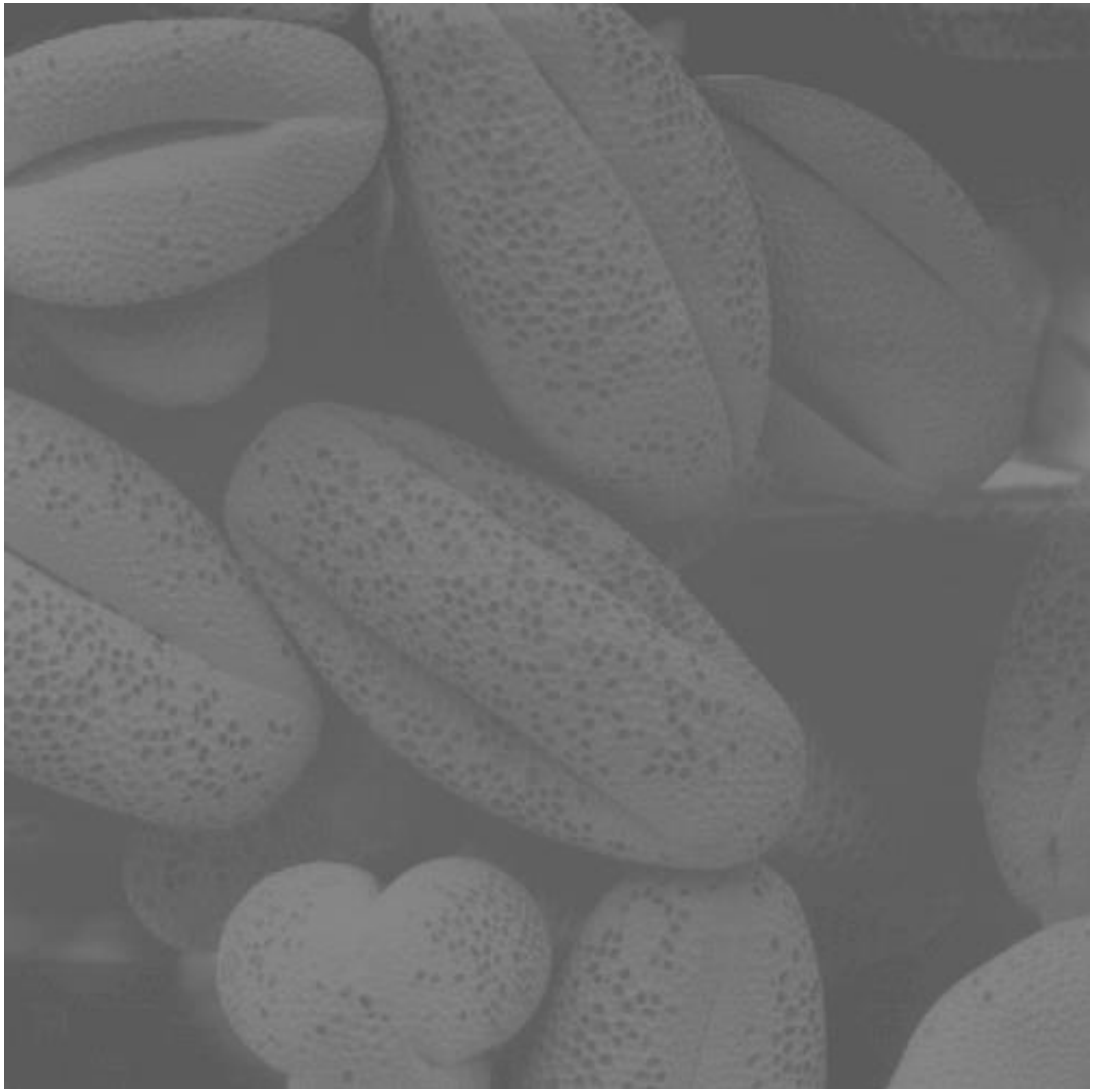
Histogram Examples (cont…)

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



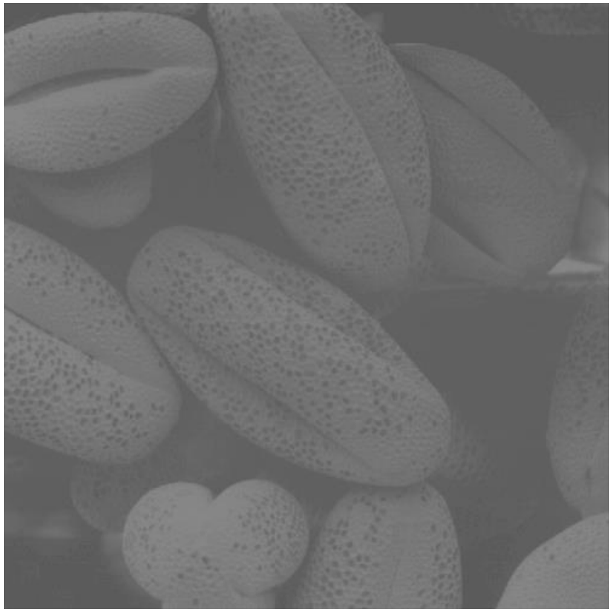


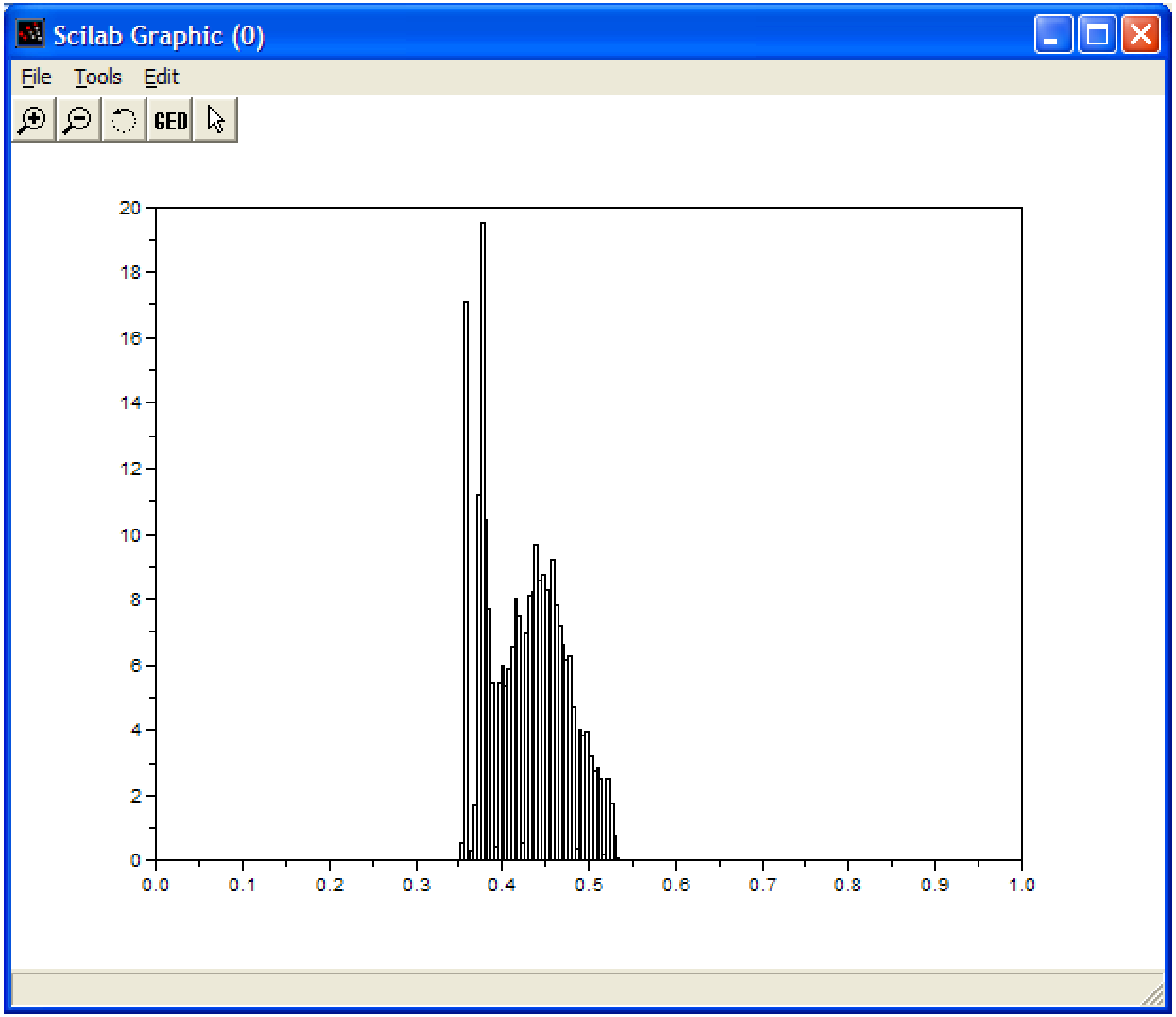
Histogram Examples (cont…)



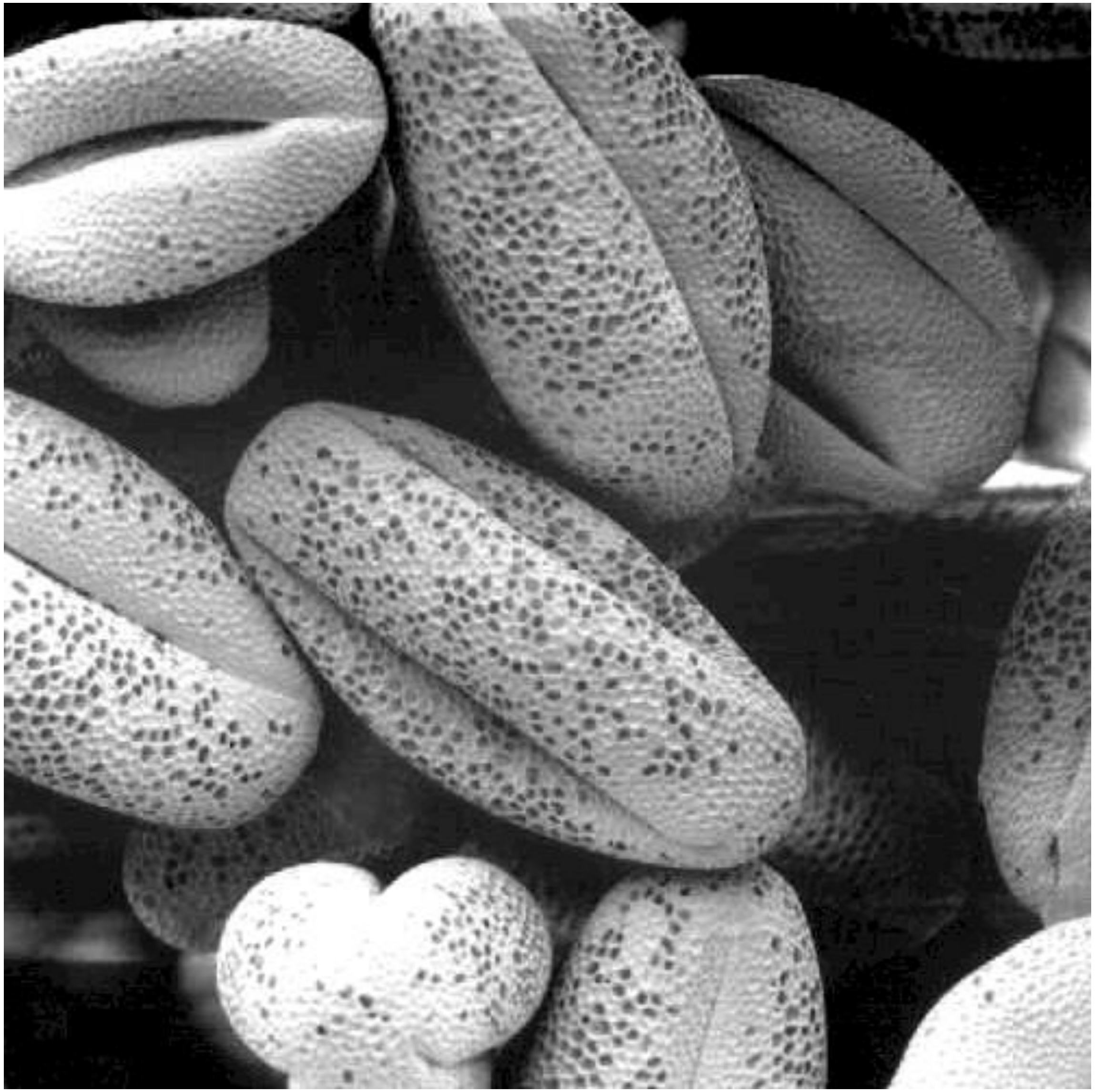
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Histogram Examples (cont…)





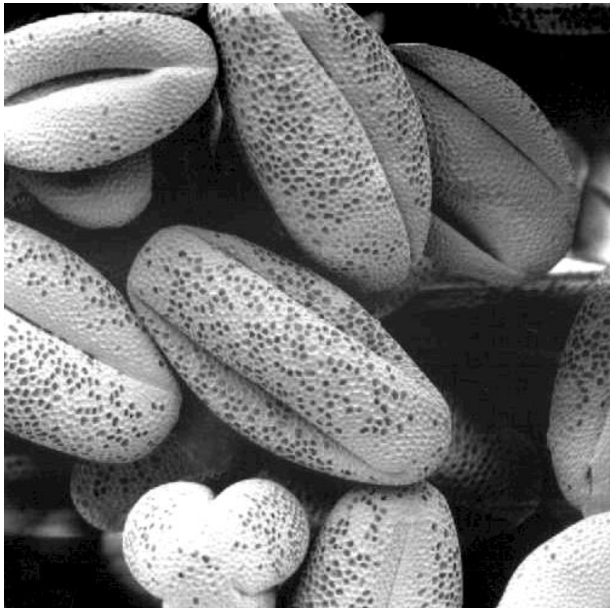
Histogram Examples (cont…)

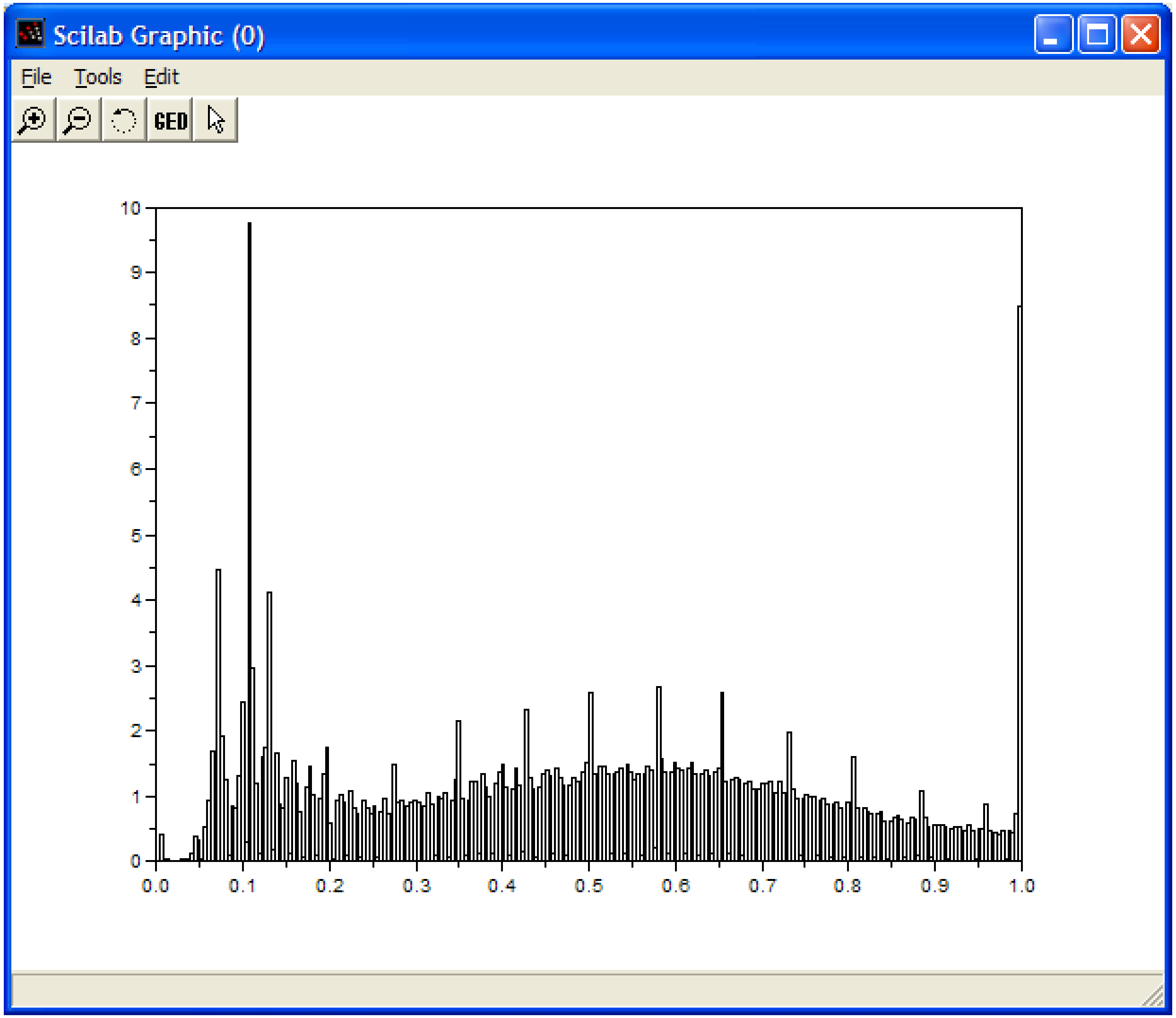


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Histogram Examples (cont…)

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



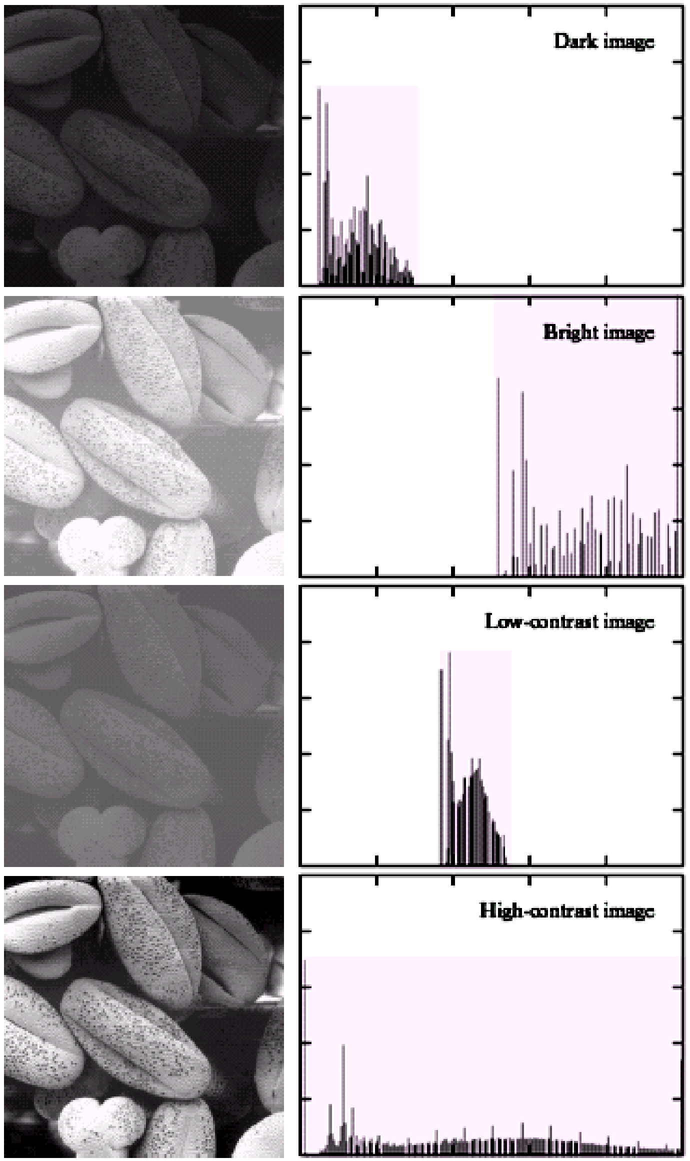


Histogram Examples (cont…)

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

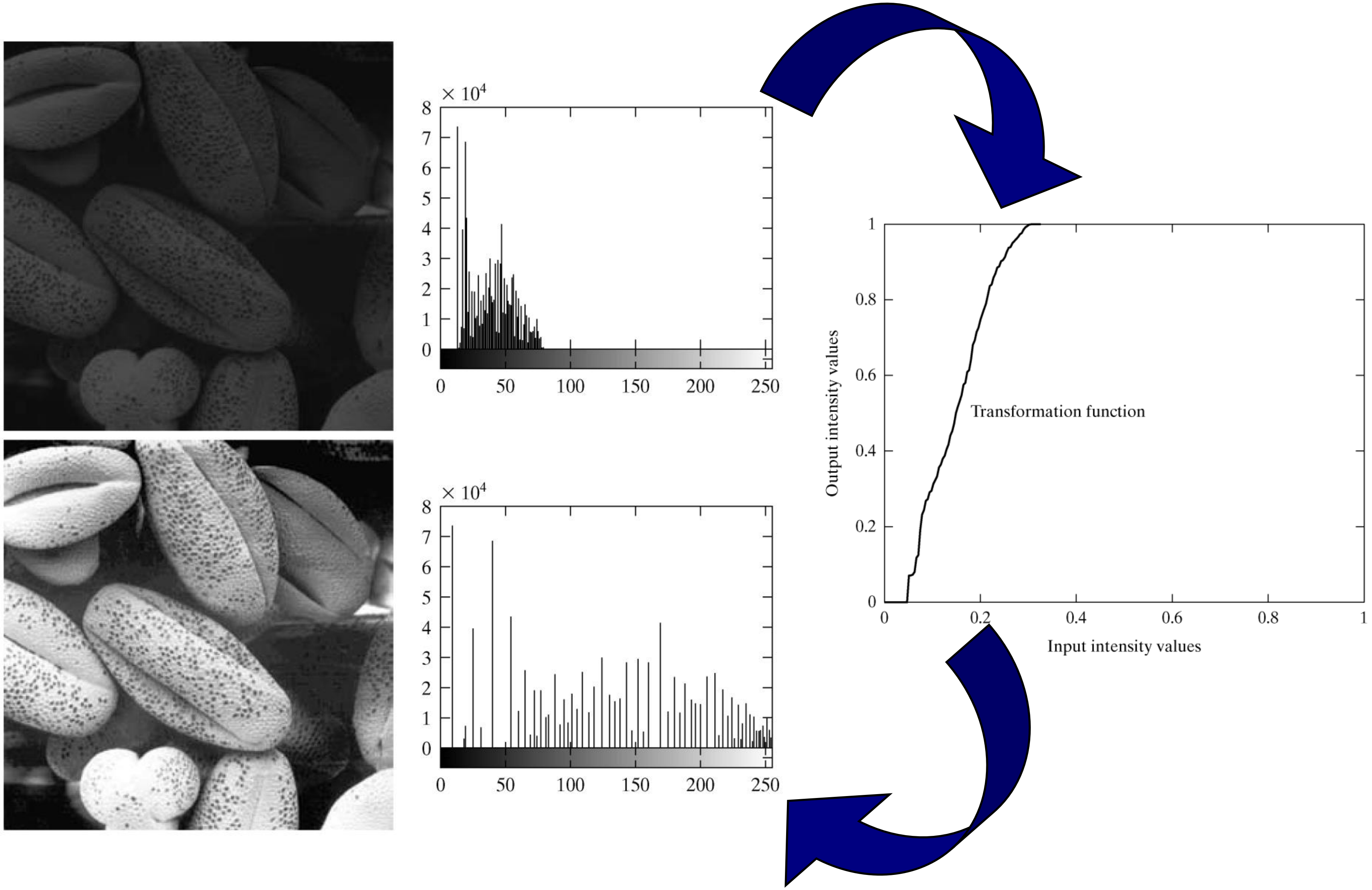
The formula for histogram equalisation is given where

- 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

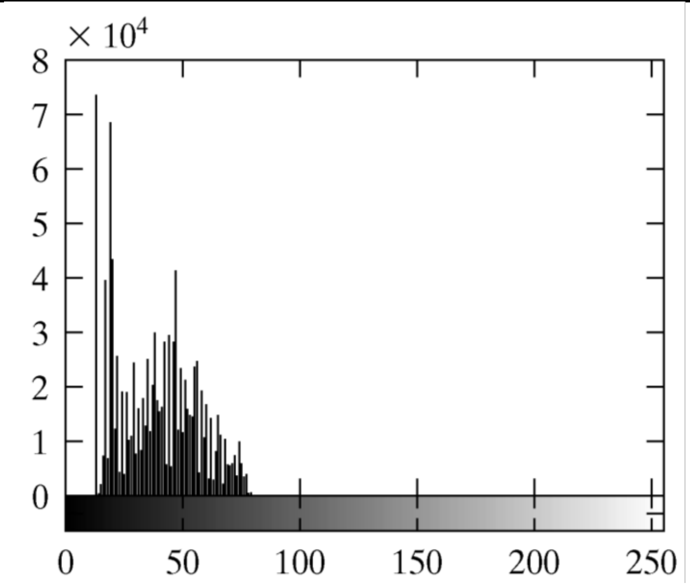
$$\begin{aligned} s_k &= T(r_k) \\ &= \sum_{j=1}^{k} p_r(r_j) \\ &= \sum_{j=1}^{k} \frac{n_j}{n} \end{aligned}$$

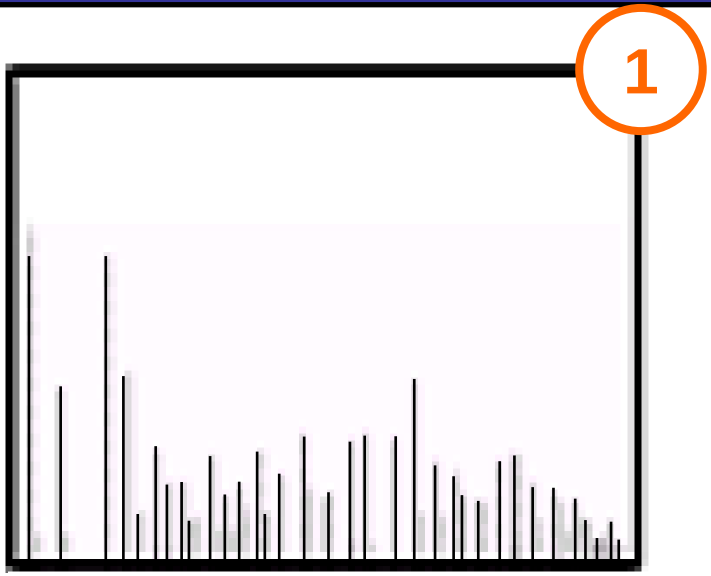
Equalisation Transformation Function

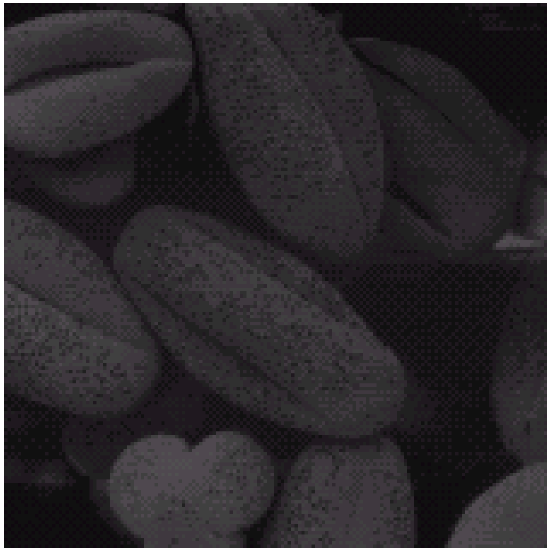
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

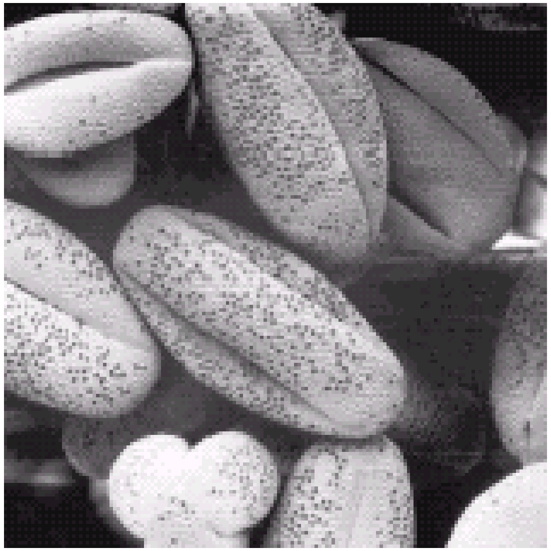


Equalisation Examples







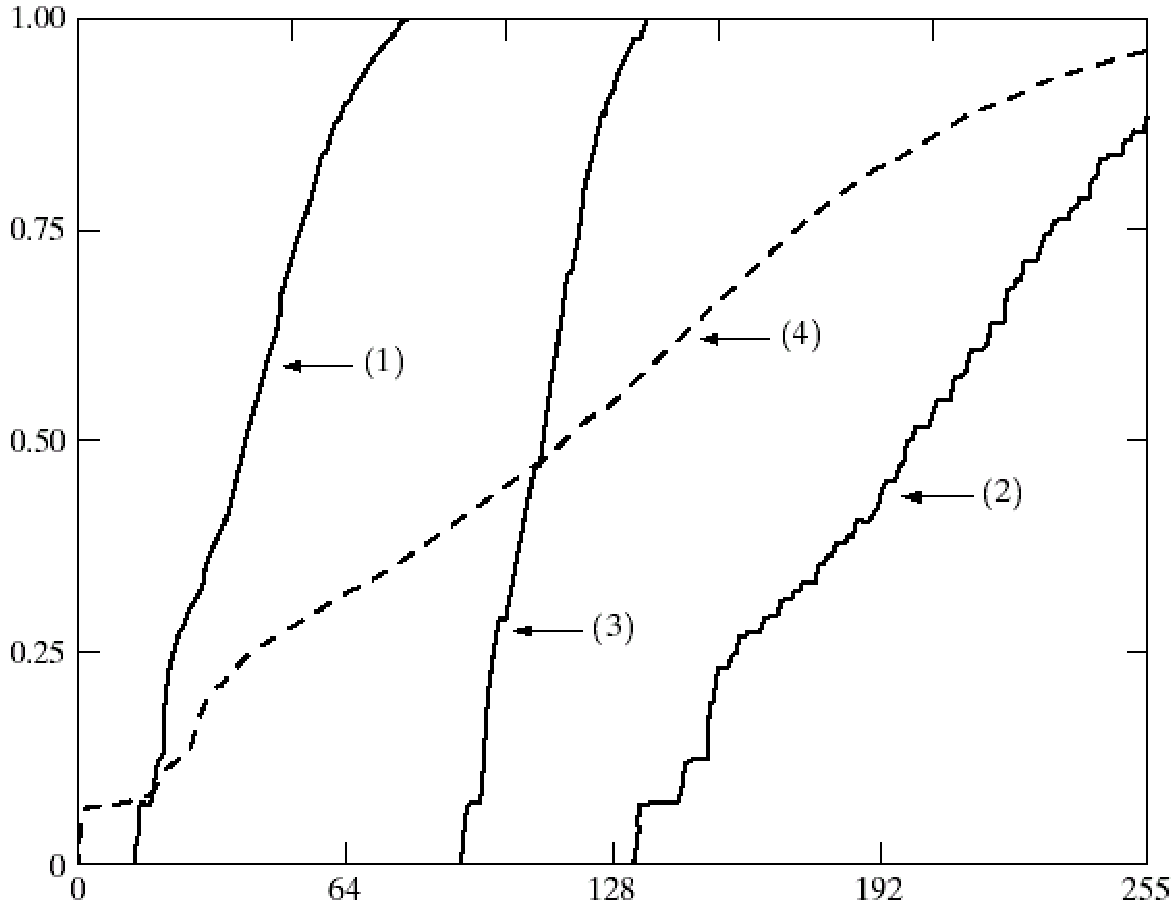


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

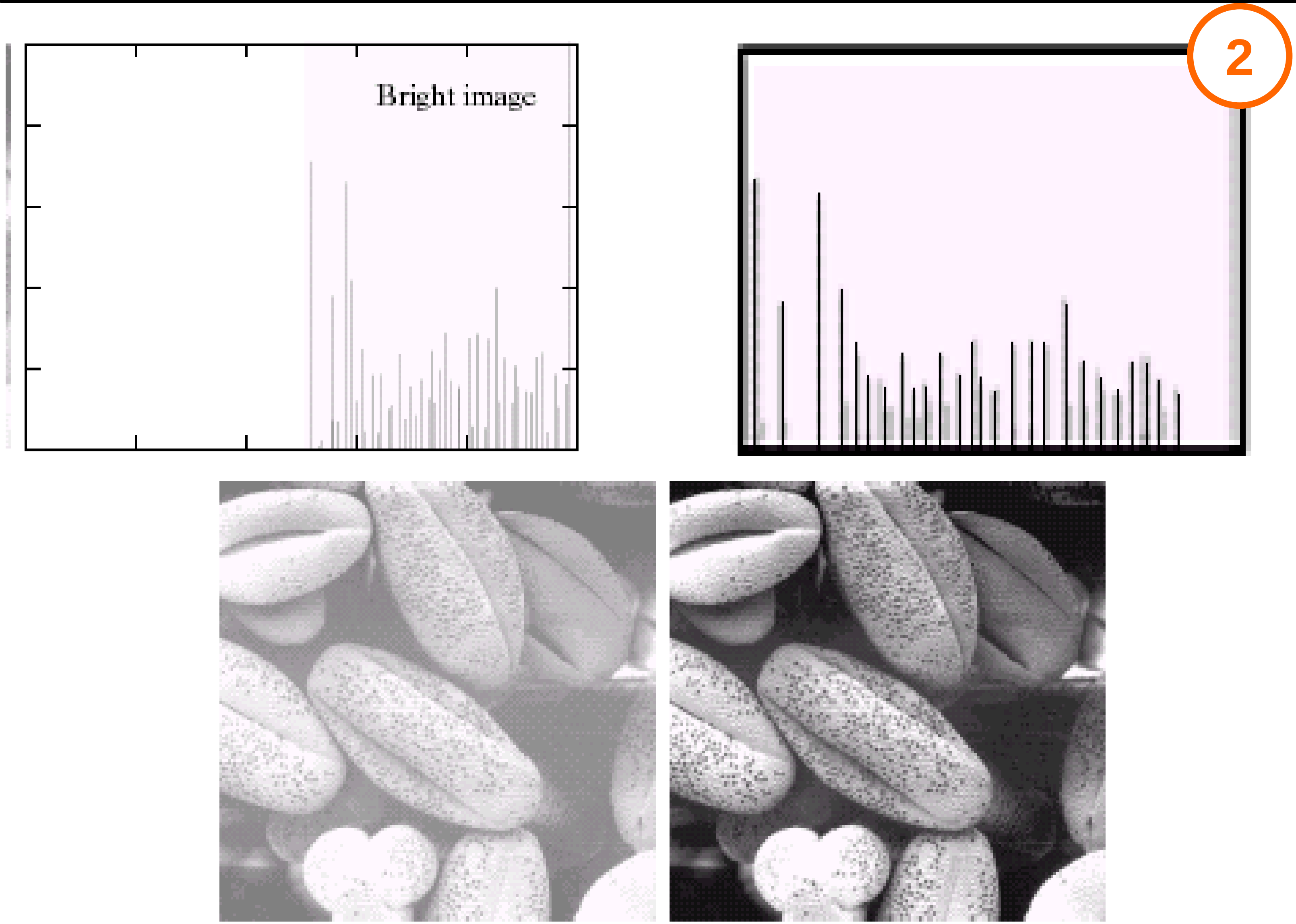
Equalisation Transformation Functions

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

The functions used to equalise the images in the previous example



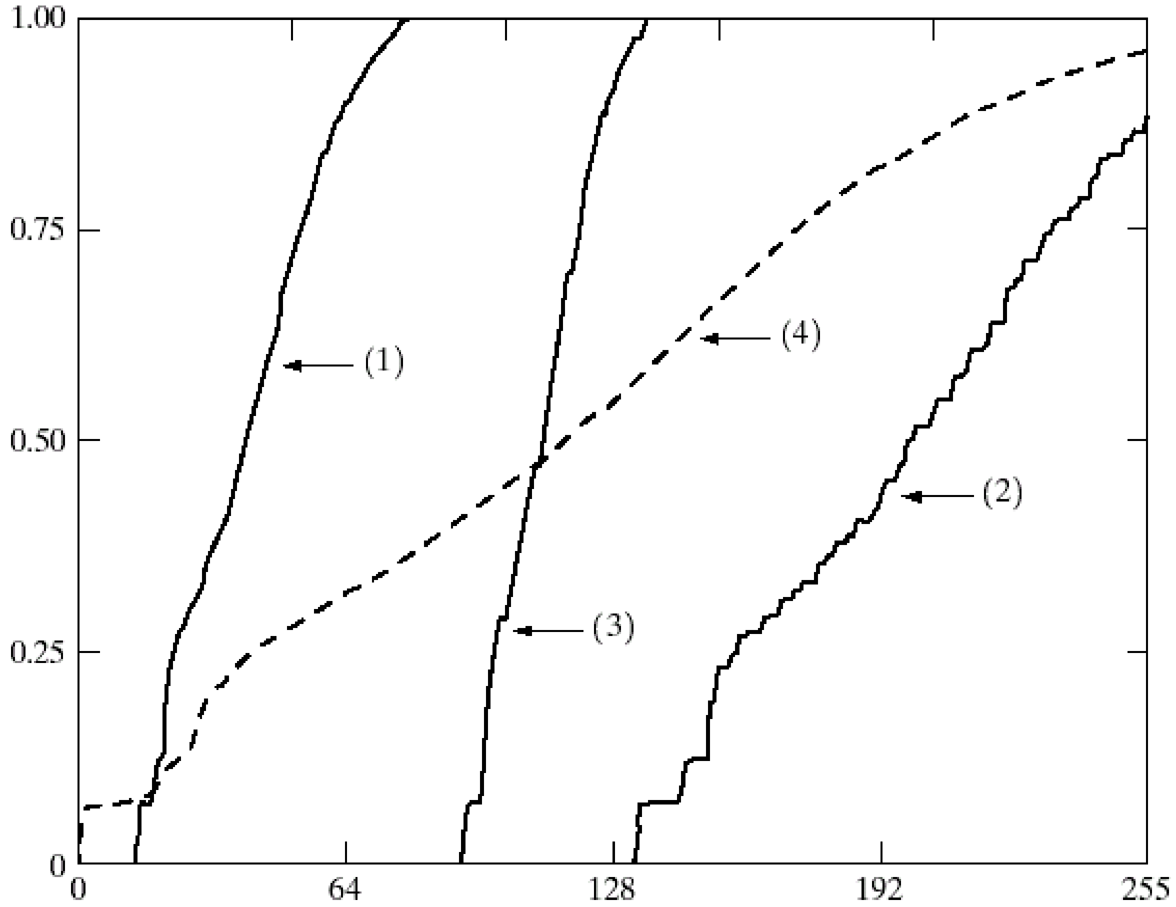
Equalisation Examples



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

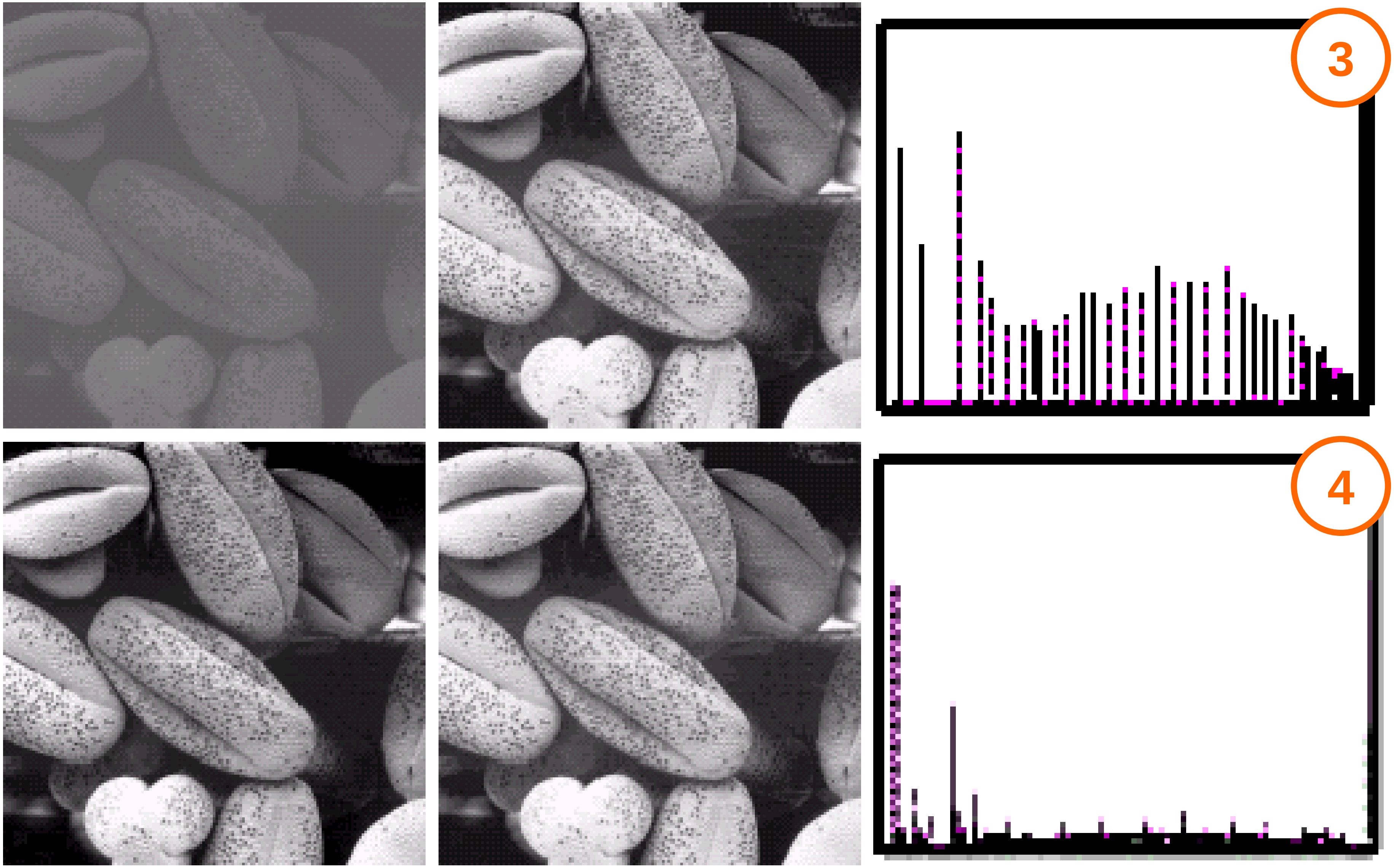
Equalisation Transformation Functions

The functions used to equalise the images in the previous example



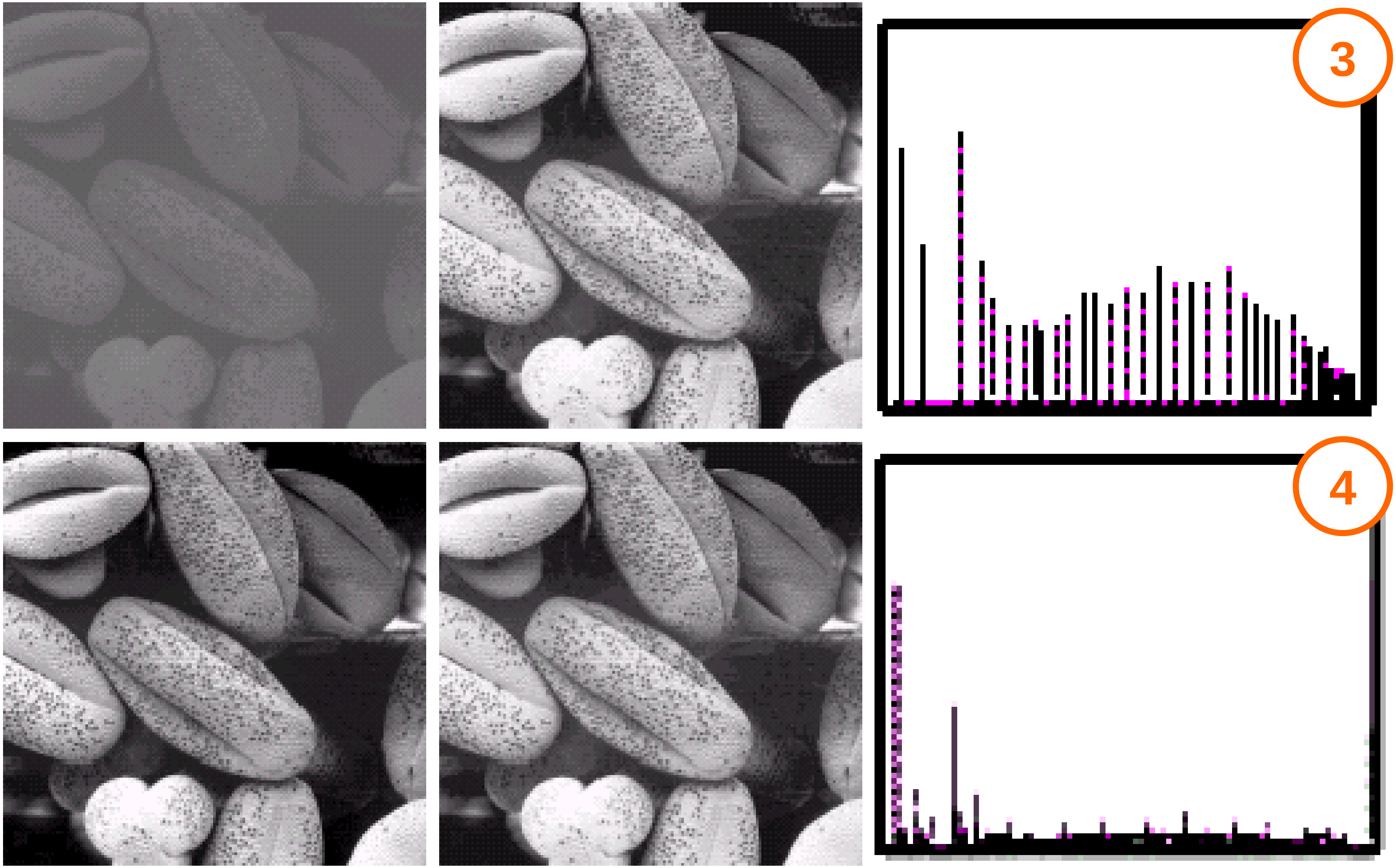
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Equalisation Examples (cont…)



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

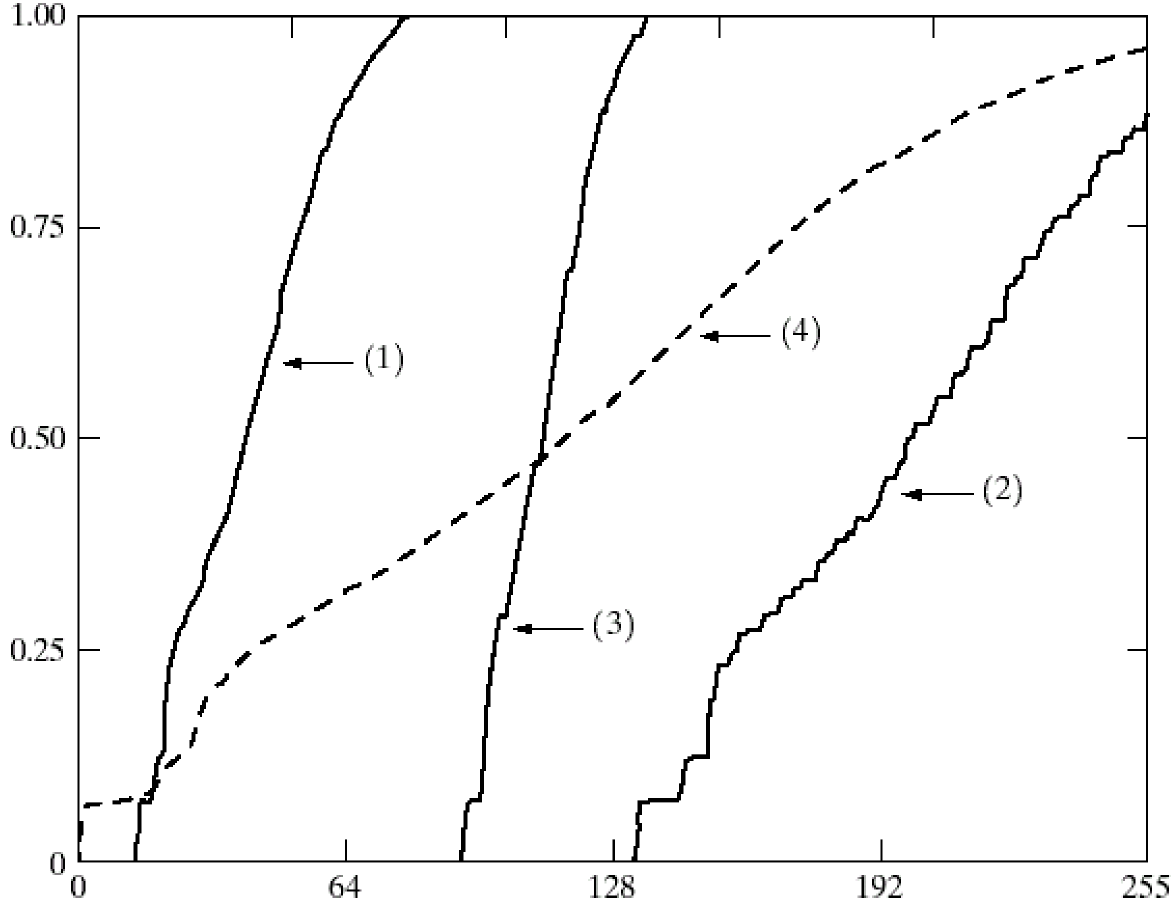
Equalisation Examples (cont…)



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

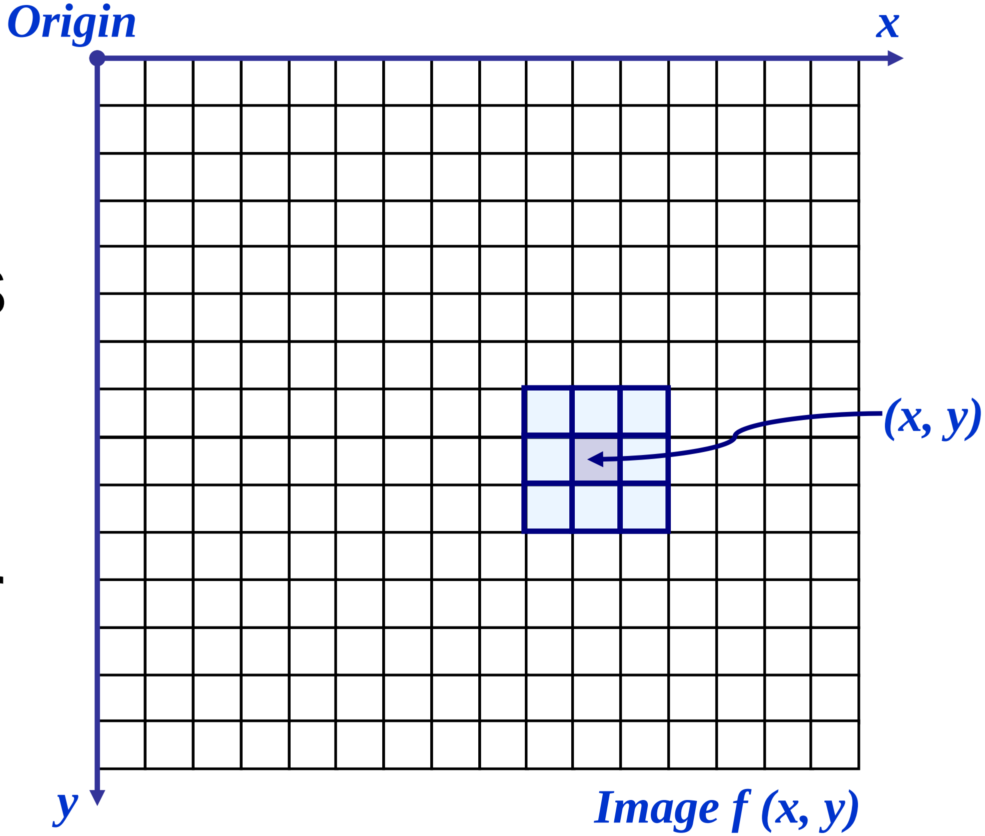
Image Enhancement (Point Processing)

Contents

- In this lecture we will look at image enhancement point processing techniques:
  - What is point processing?
  - Negative images
  - Thresholding
  - Logarithmic transformation
  - Power law transforms
  - Grey level slicing
  - Bit plane slicing

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)

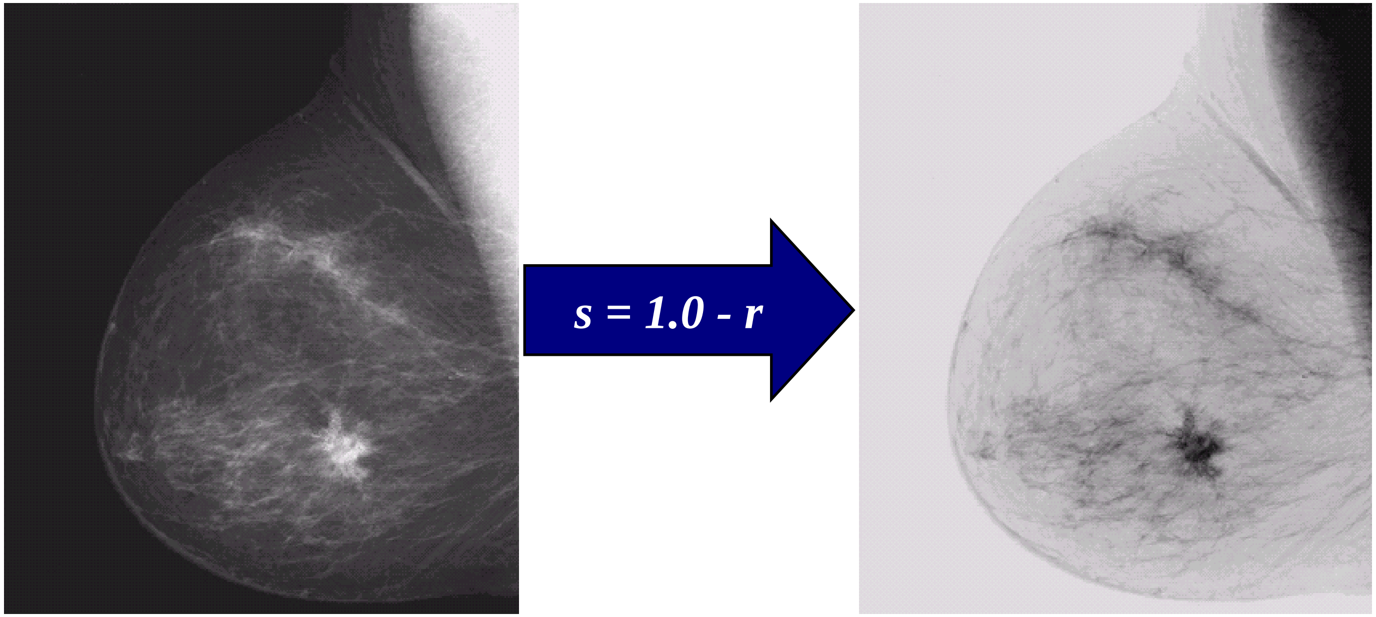


Point Processing

- 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

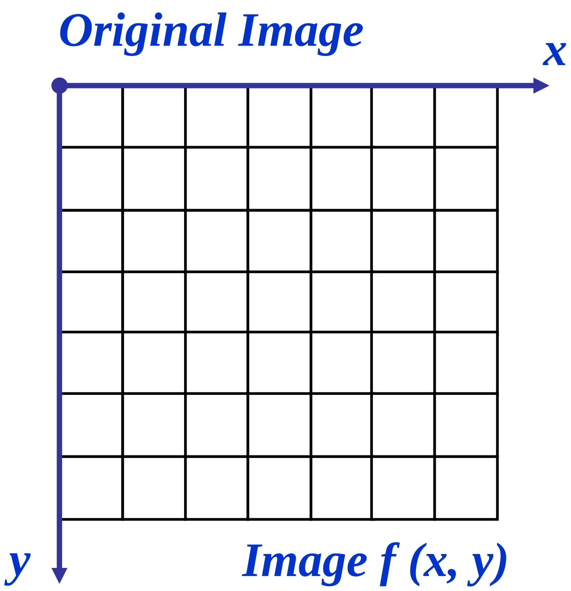
- Negative images are useful for enhancing white or grey detail embedded in dark regions of an image
  - – Note how much clearer the tissue is in the negative image of the mammogram below

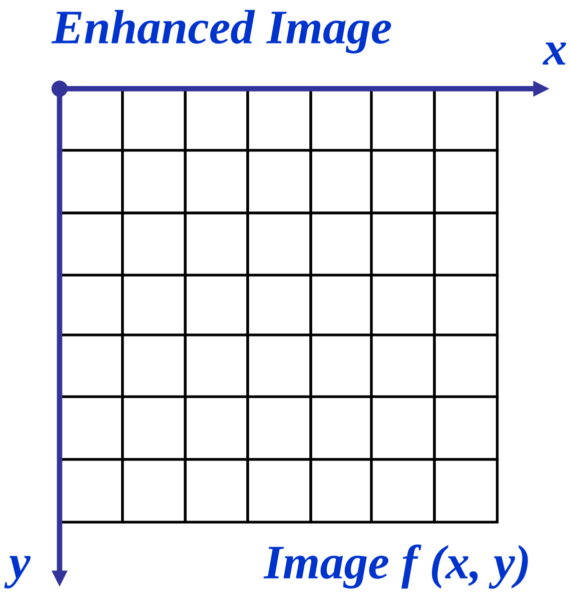


$s = 1.0 - r$

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

Point Processing Example: Negative Images (cont…)

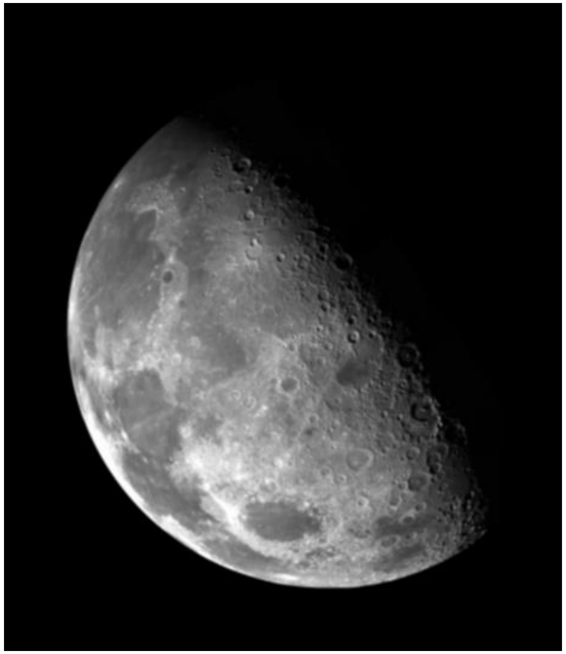




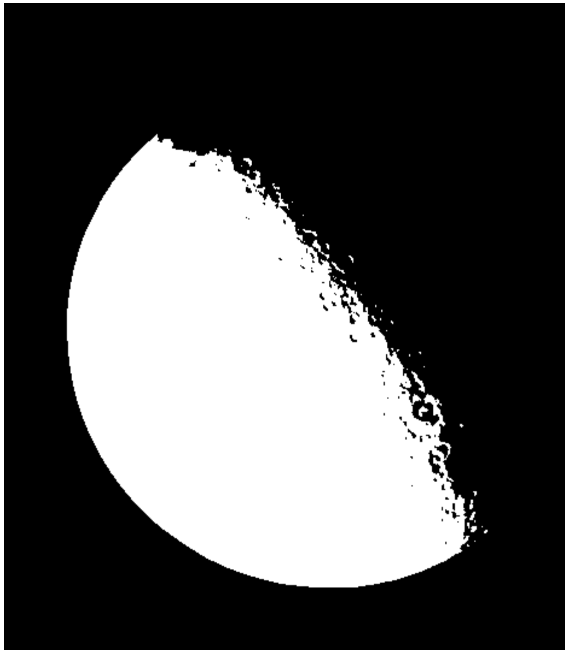
$$s = intensity_{max} - r$$

Point Processing Example: Thresholding

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

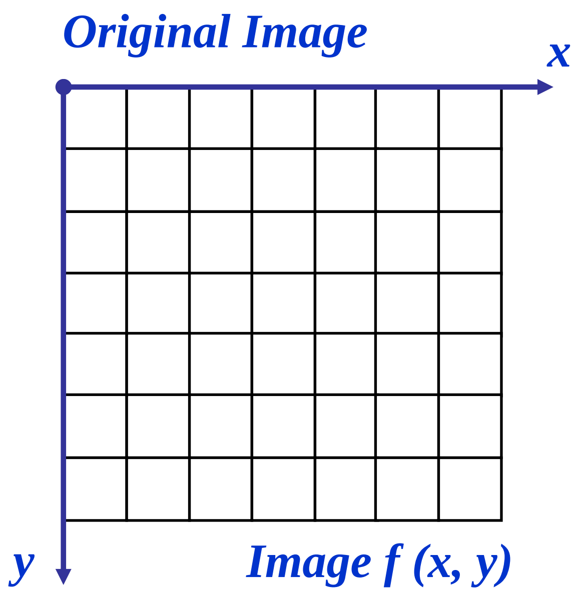


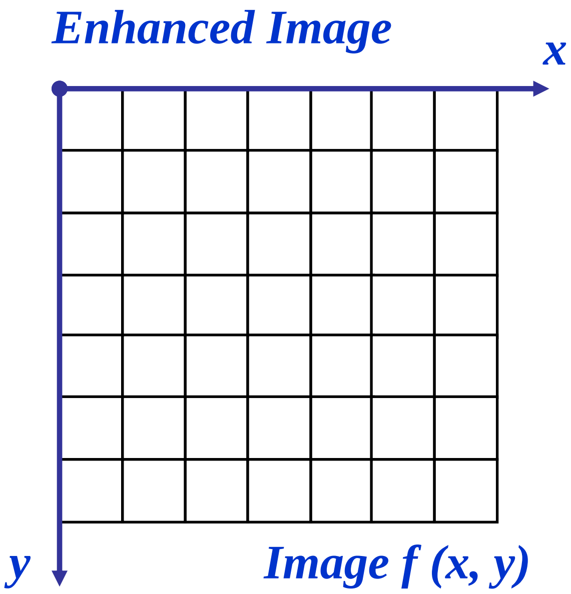
$$s = \begin{cases} 1.0 & r > \text{threshold} \\ 0.0 & r <= \text{threshold} \end{cases}$$



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

Point Processing Example: Thresholding (cont…)

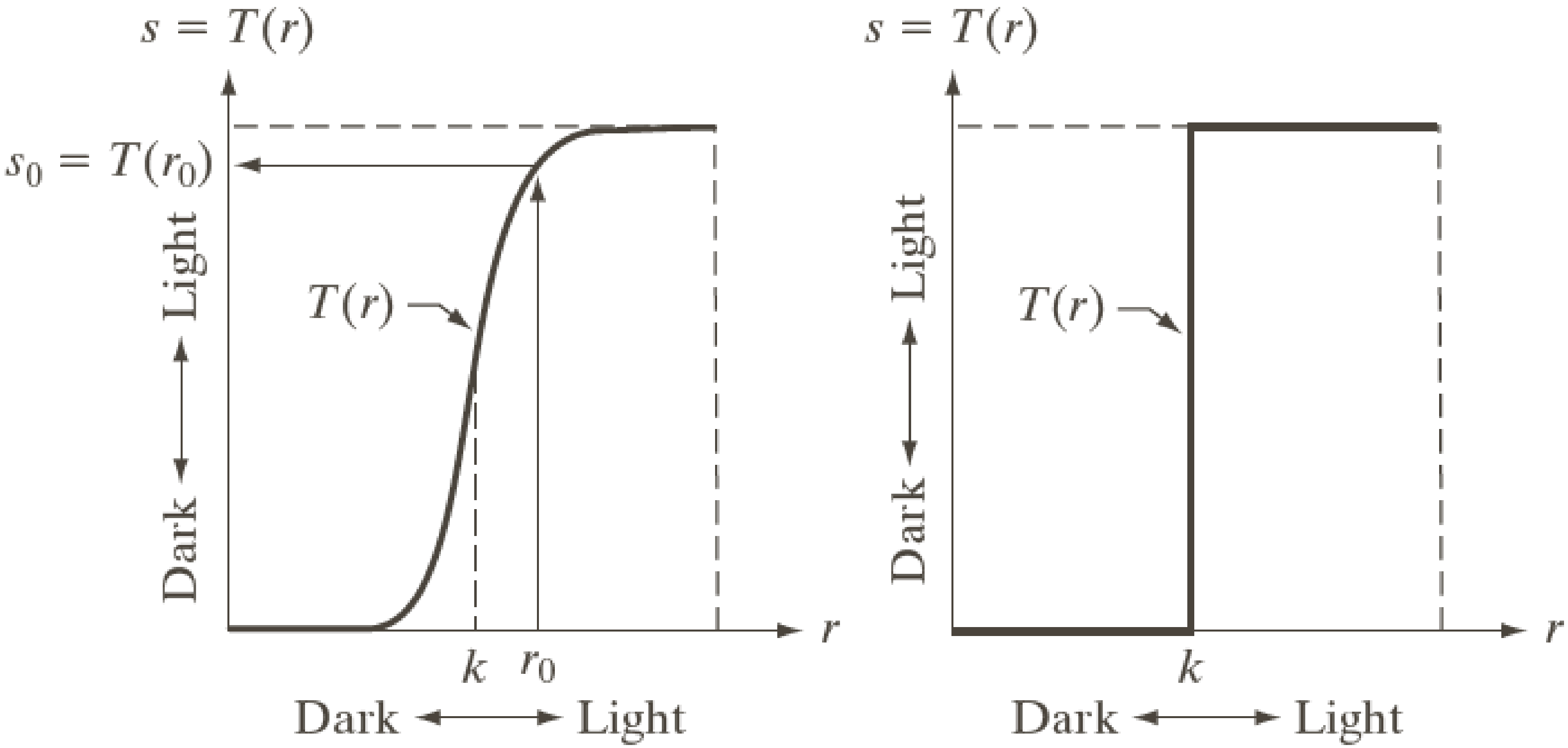




$$s = \begin{cases} 1.0 & r > threshold \\ 0.0 & r <= threshold \end{cases}$$

Intensity Transformations

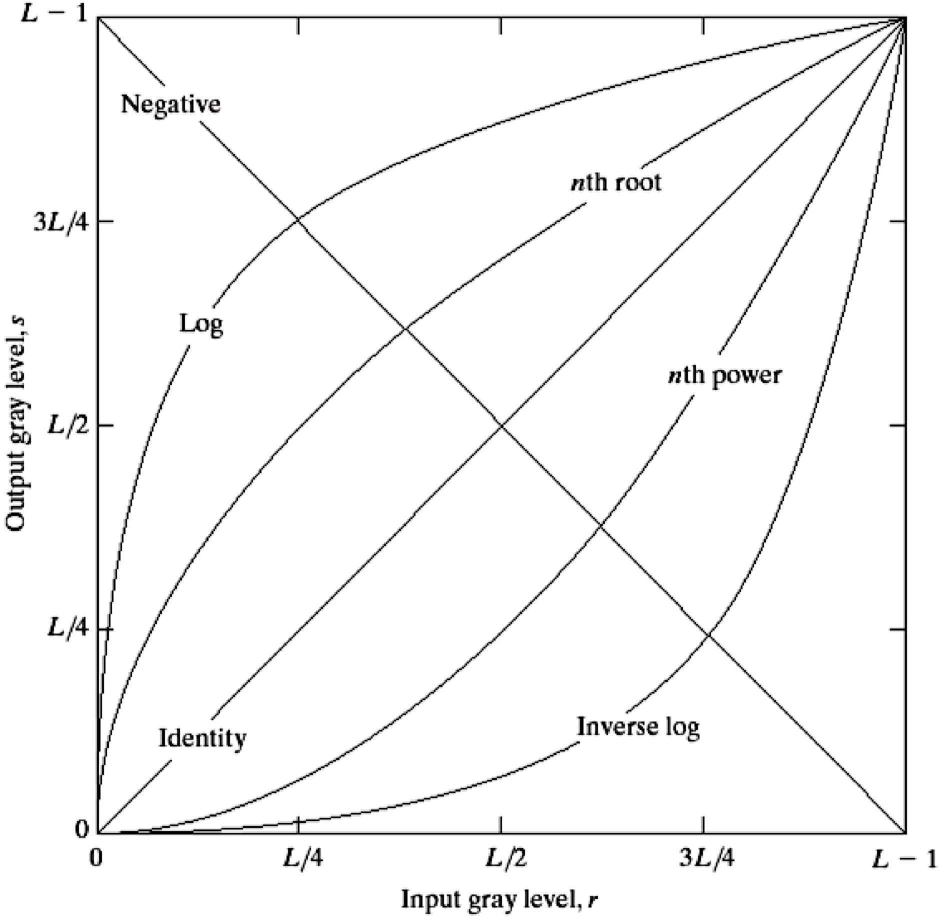
Images taken from Gonzalez & Woods, Digital Image Processing (2002)



Basic Grey Level Transformations

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

- 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
    - nth power/nth root

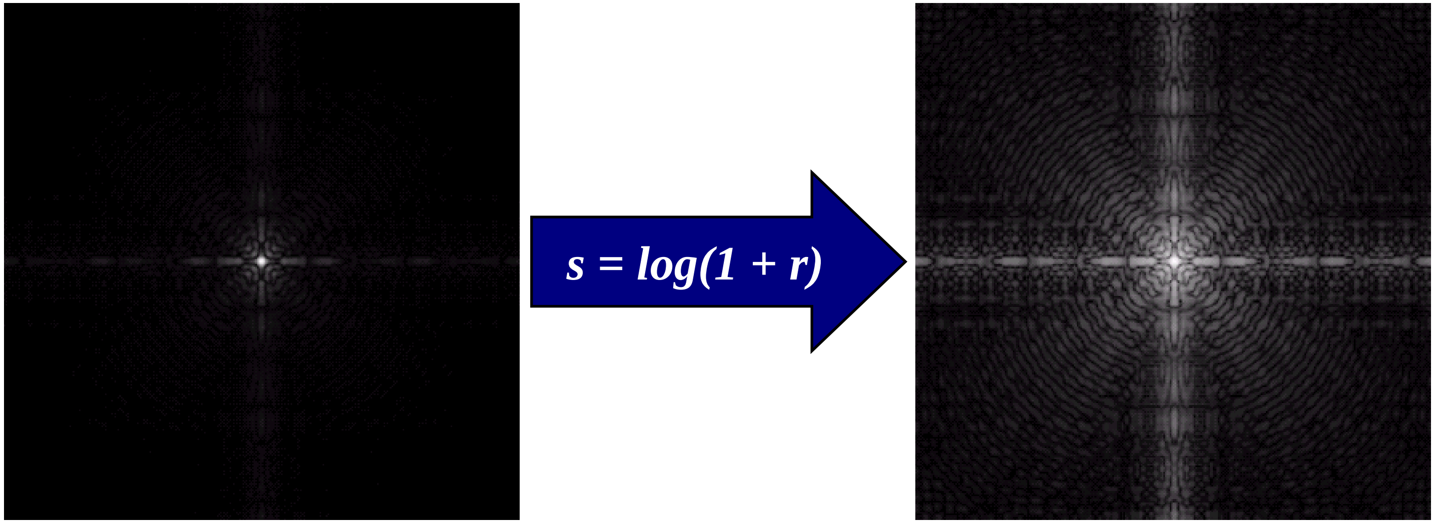


Logarithmic Transformations

- 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…)

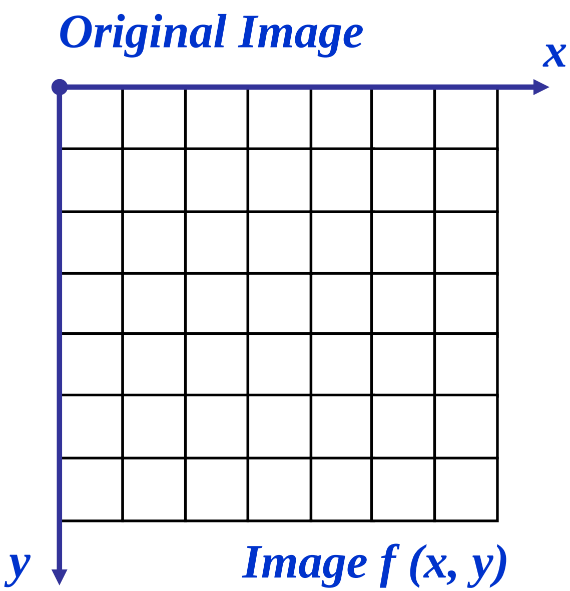
- Log functions are particularly useful when the input grey level values may have an extremely large range of values
- In the following example the Fourier transform of an image is put through a log transform to reveal more detail

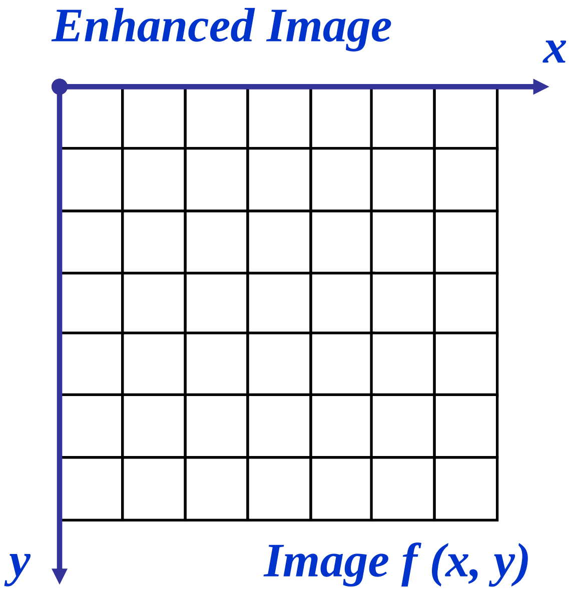


$s = \log(1 + r)$

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

Logarithmic Transformations (cont…)





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

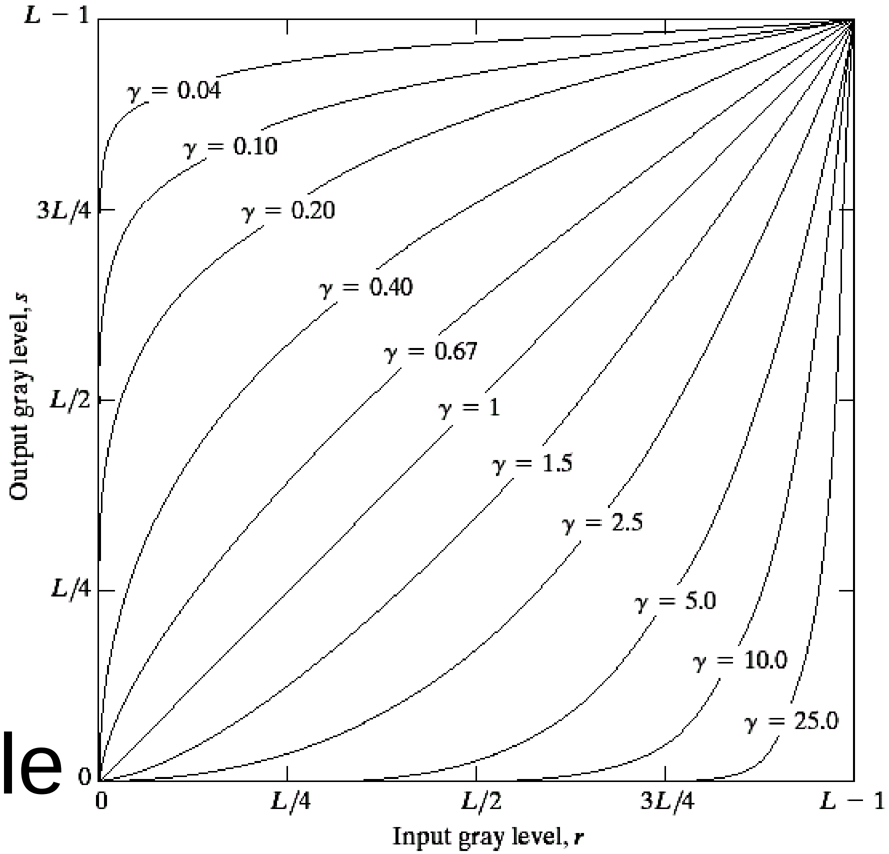
We usually set c to 1

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

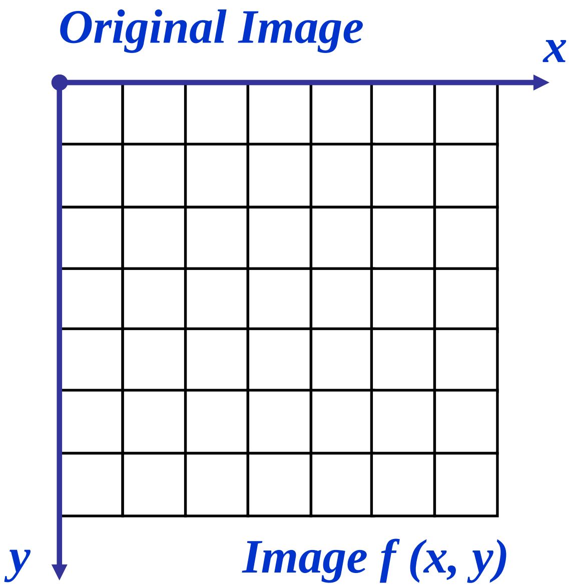
Power Law Transformations

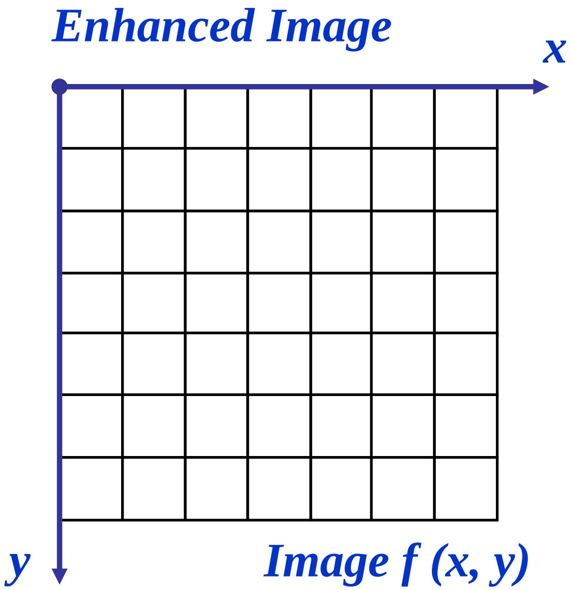
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

- 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



Power Law Transformations (cont…)

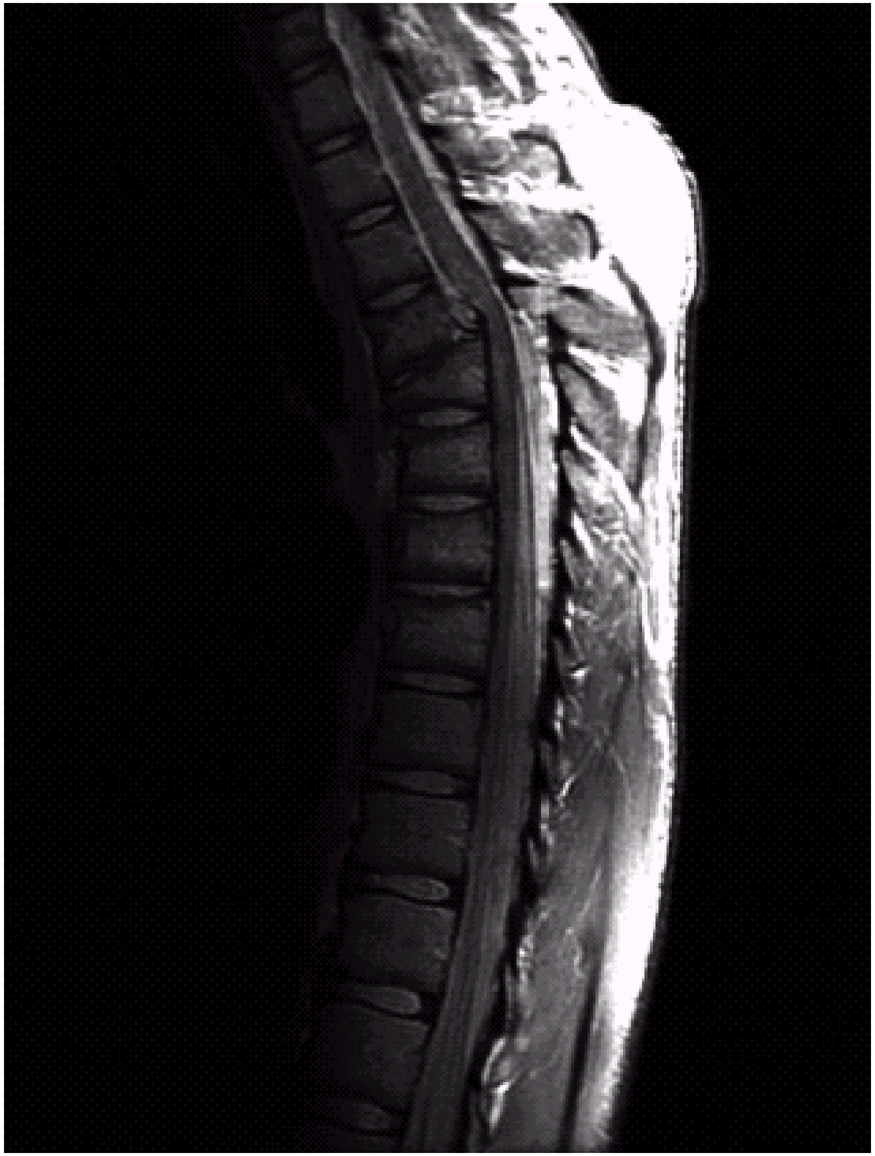




$$s = r^{\gamma}$$

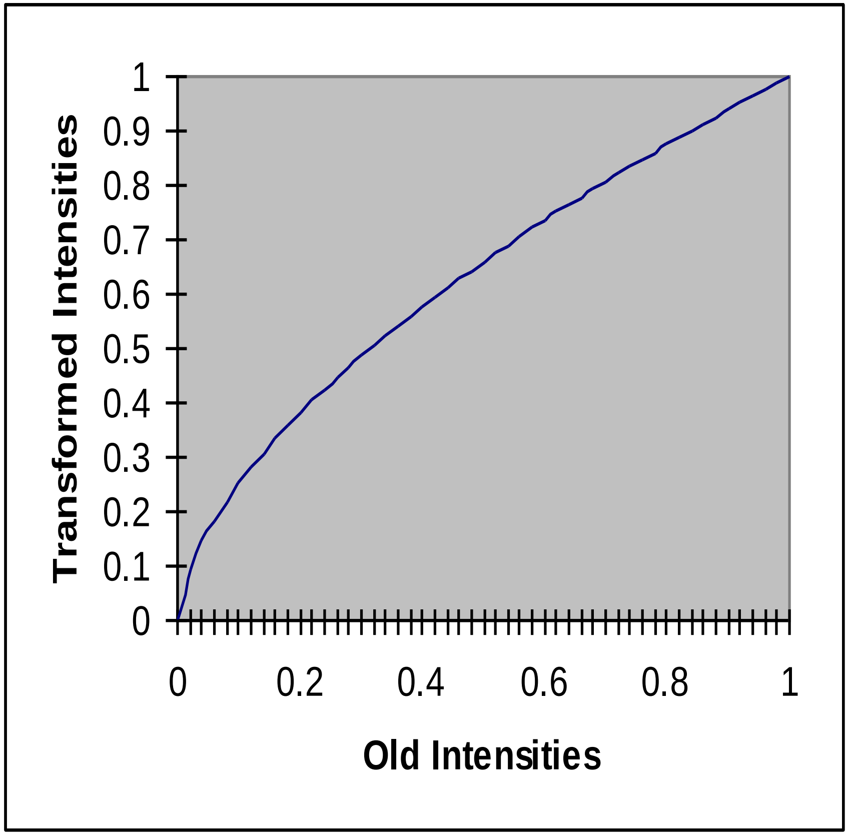
- We usually set c to 1
- Grey levels must be in the range [0.0, 1.0]

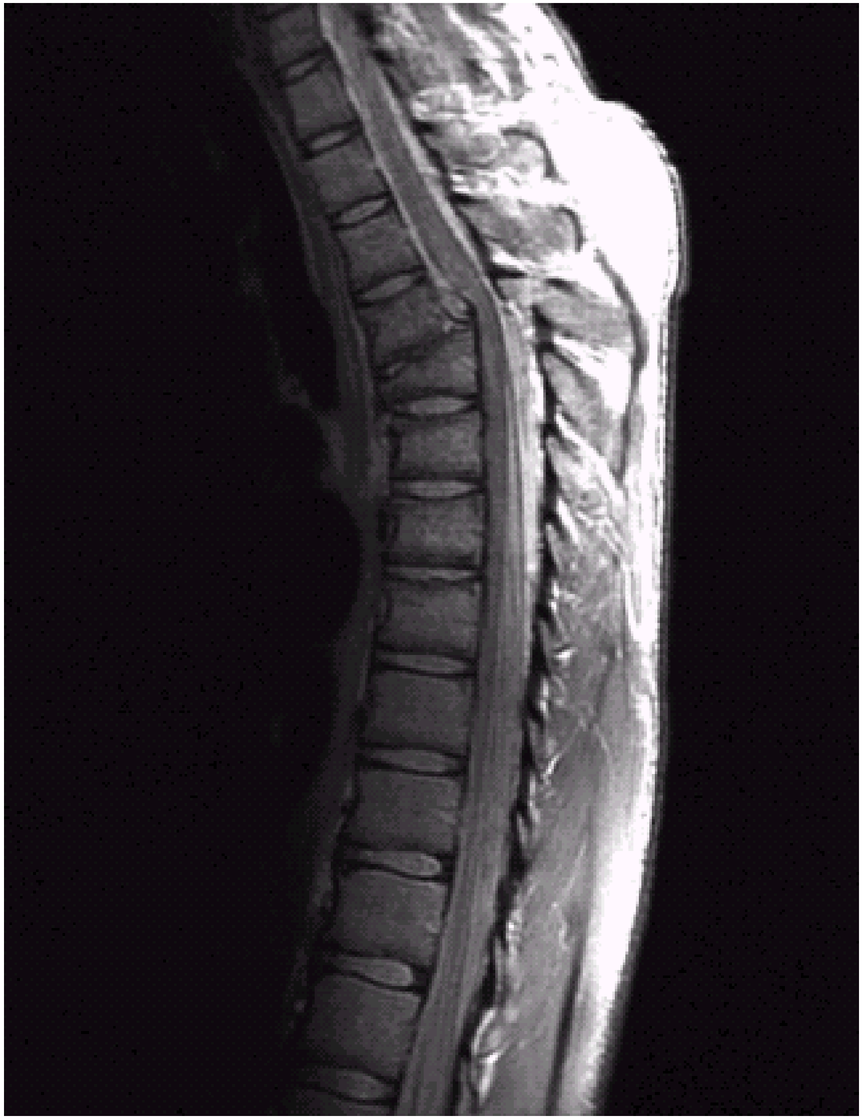
Power Law Example



Power Law Example (cont…)

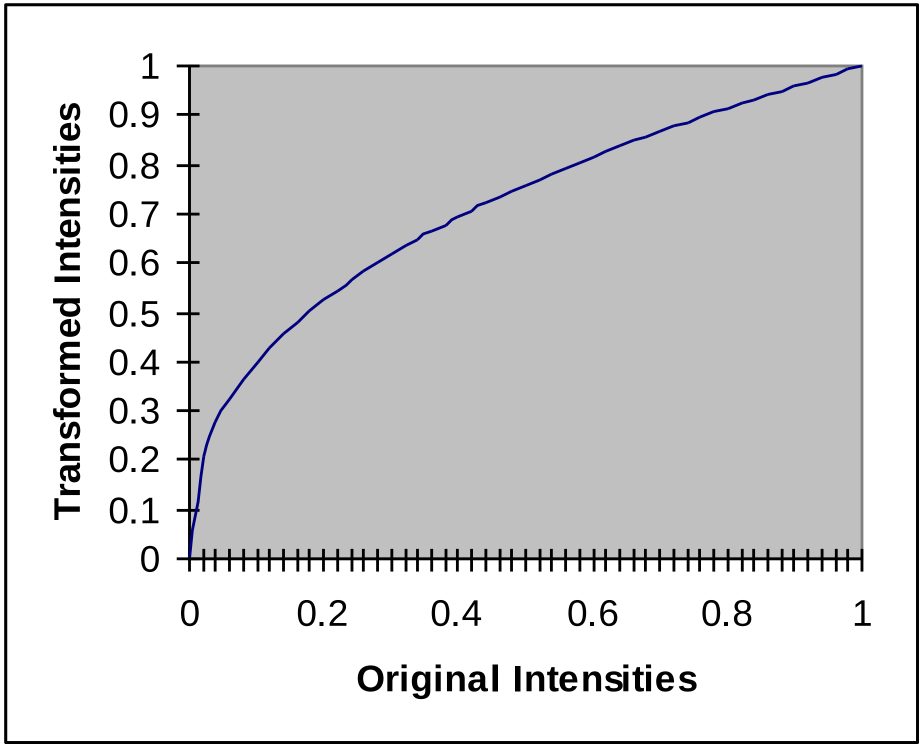
$\gamma = 0.6$

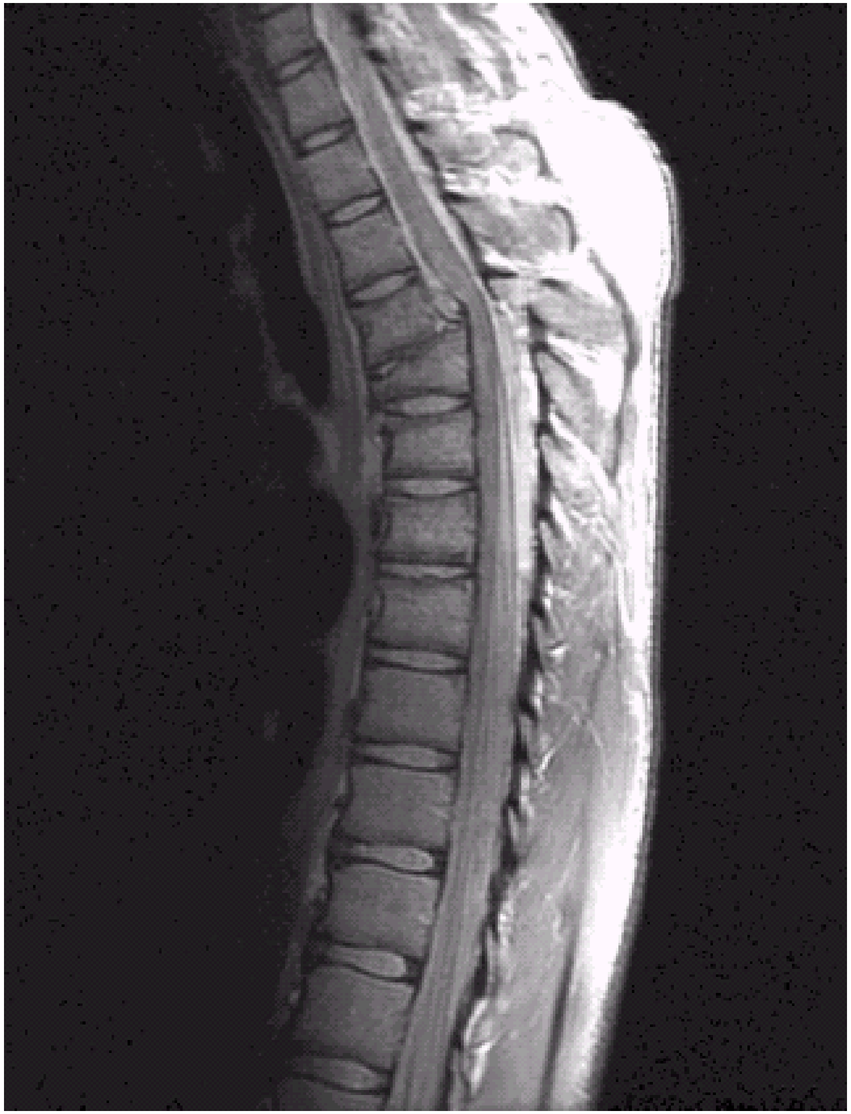




Power Law Example (cont…)

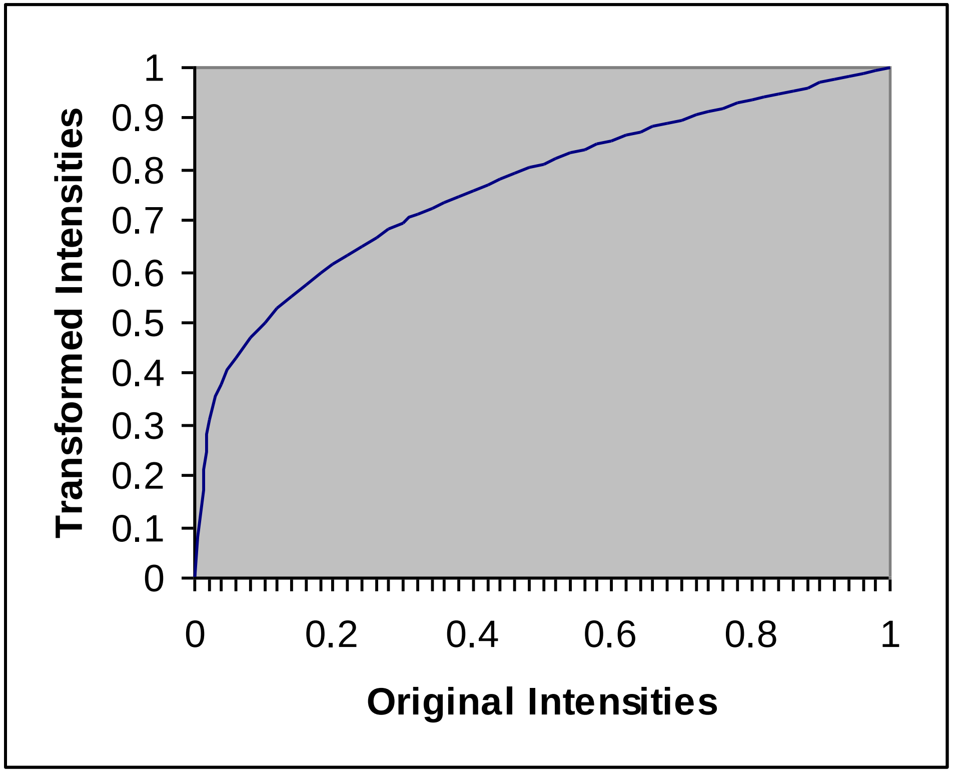
$$\gamma = 0.4$$

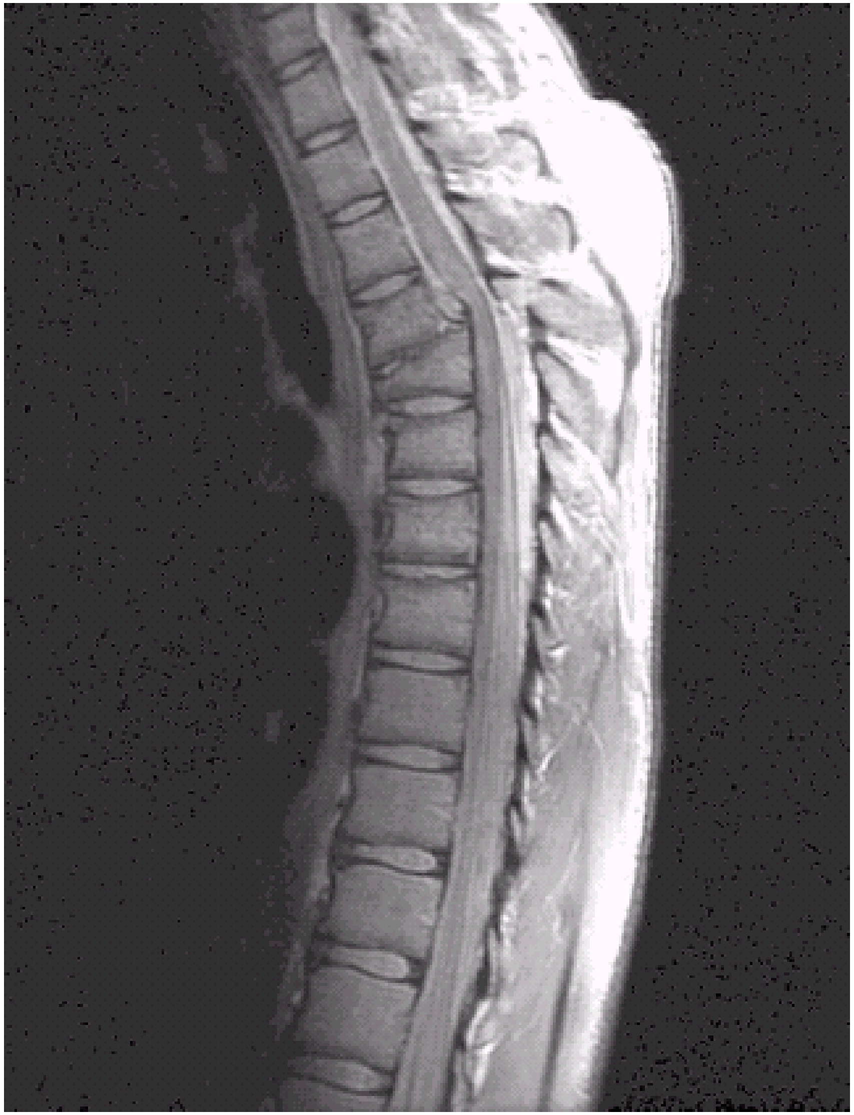




Power Law Example (cont…)

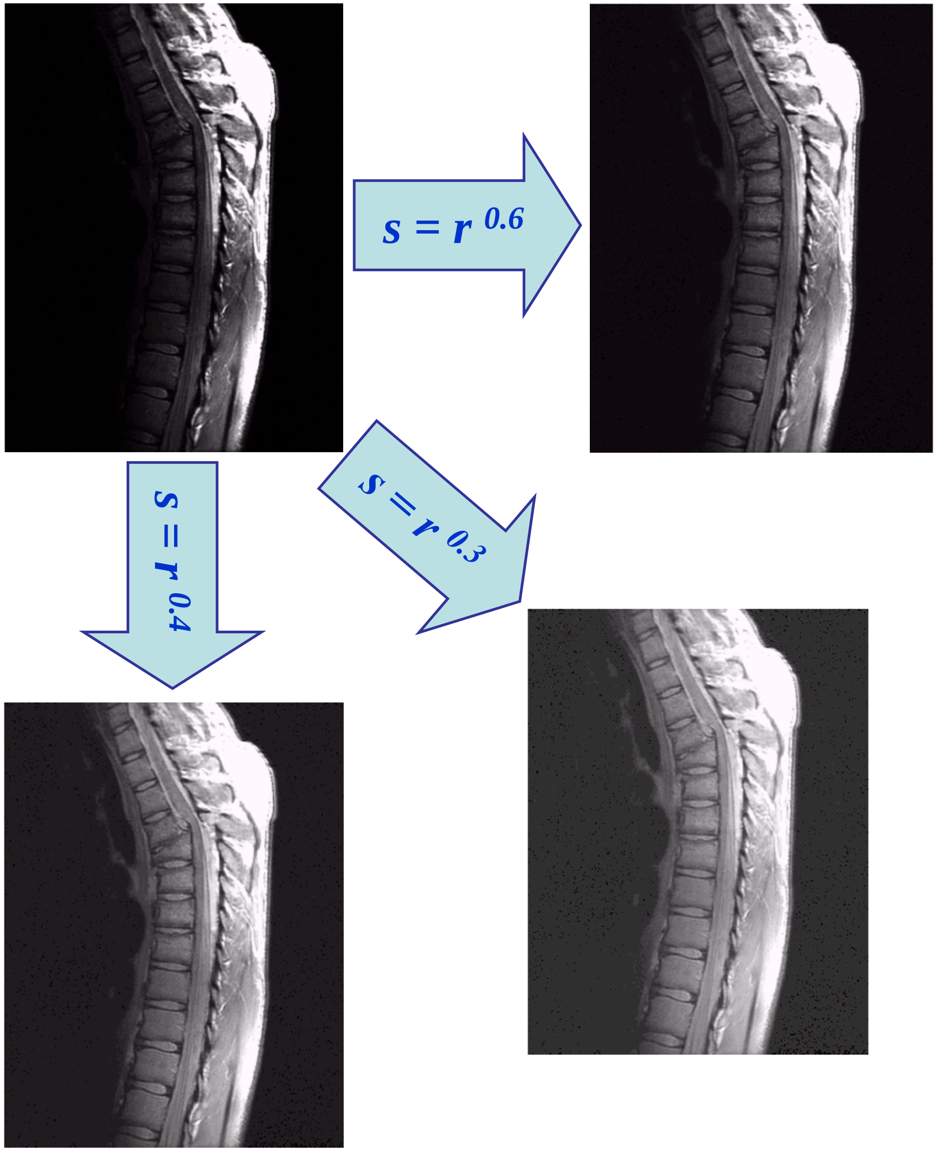
$\gamma = 0.3$





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

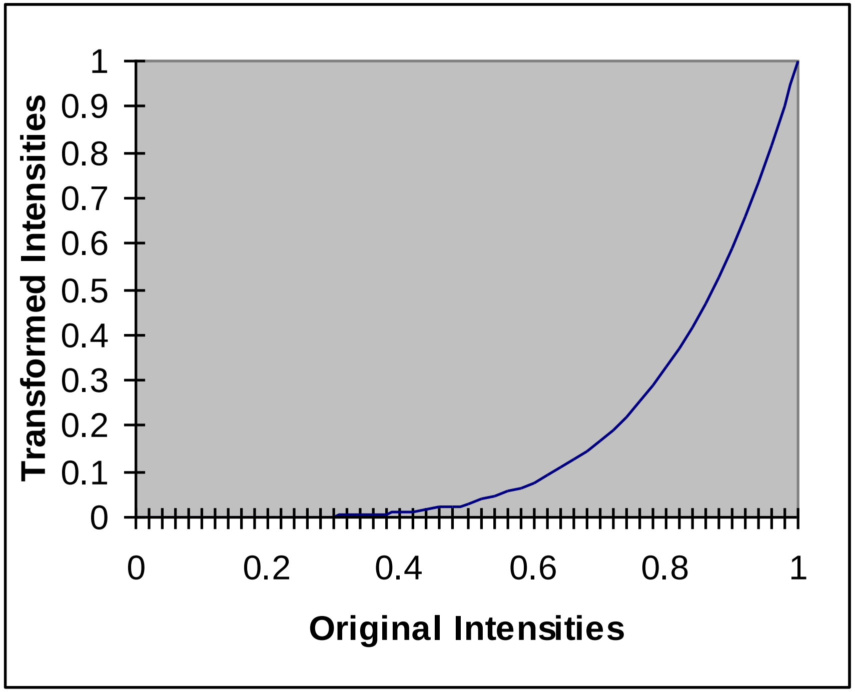


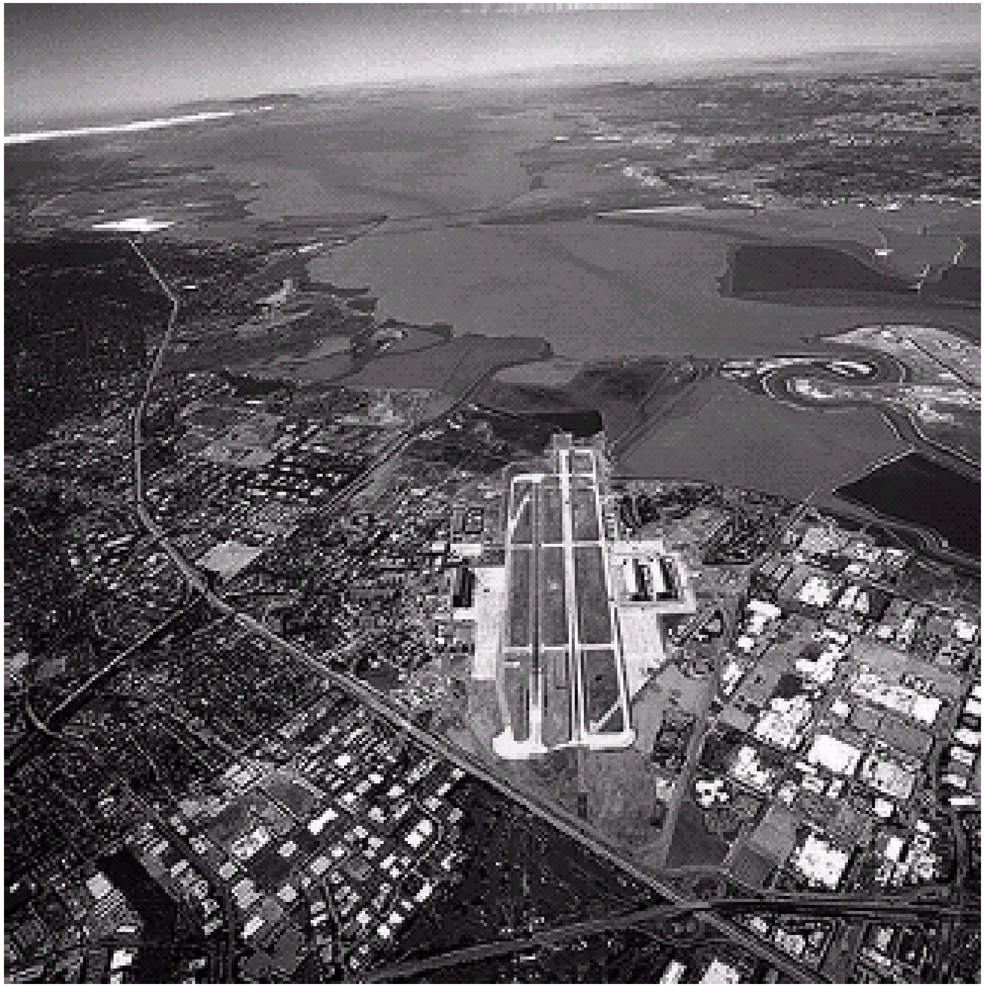
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Power Law Example

Power Law Example (cont…)

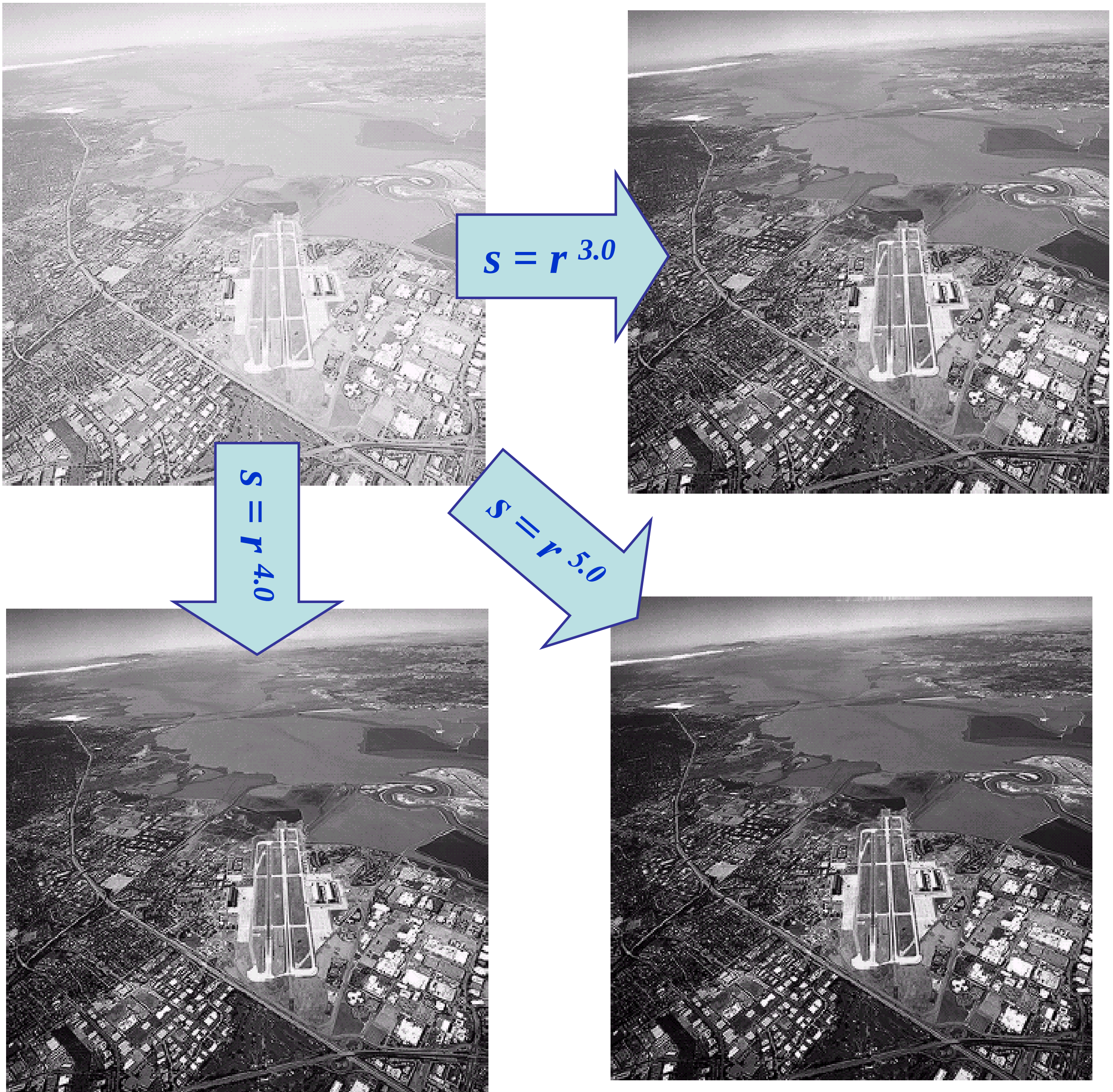
$$\gamma = 5.0$$





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

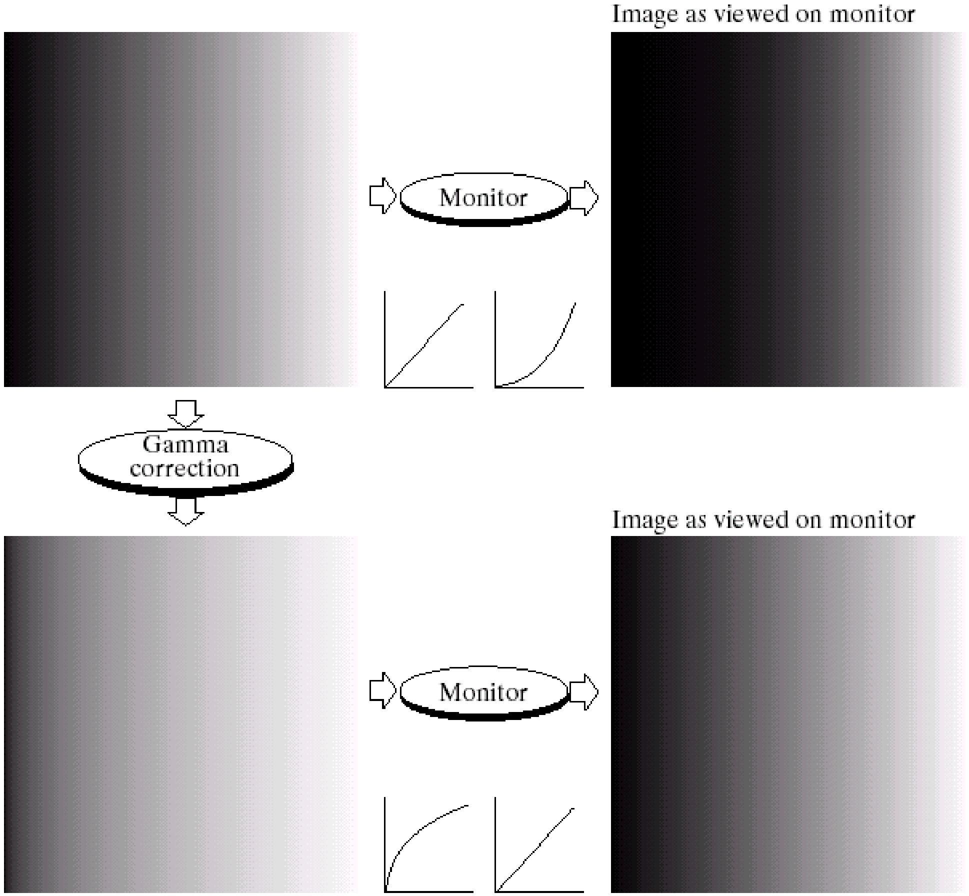


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

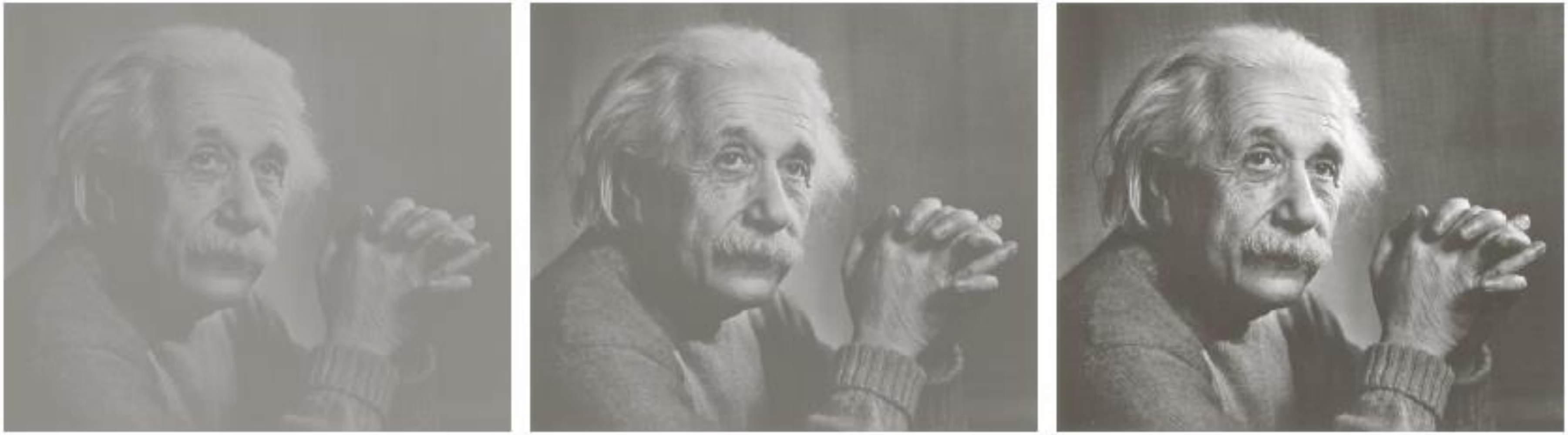
Gamma Correction

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

- 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

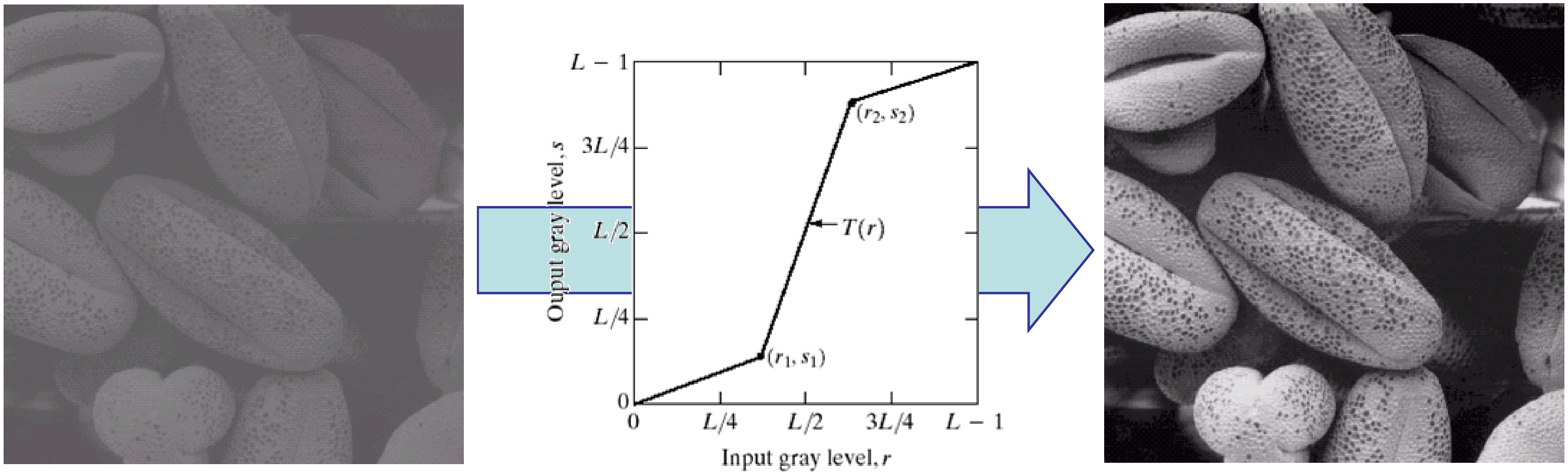


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Piecewise Linear Transformation Functions

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

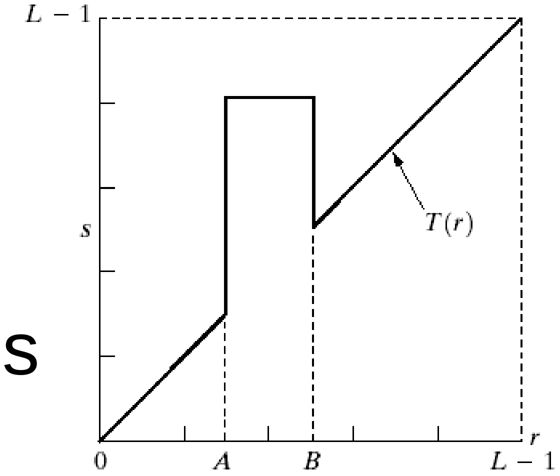
- Rather than using a well defined mathematical function we can use arbitrary user-defined transforms
- The images below show a contrast stretching linear transform to add contrast to a poor quality image

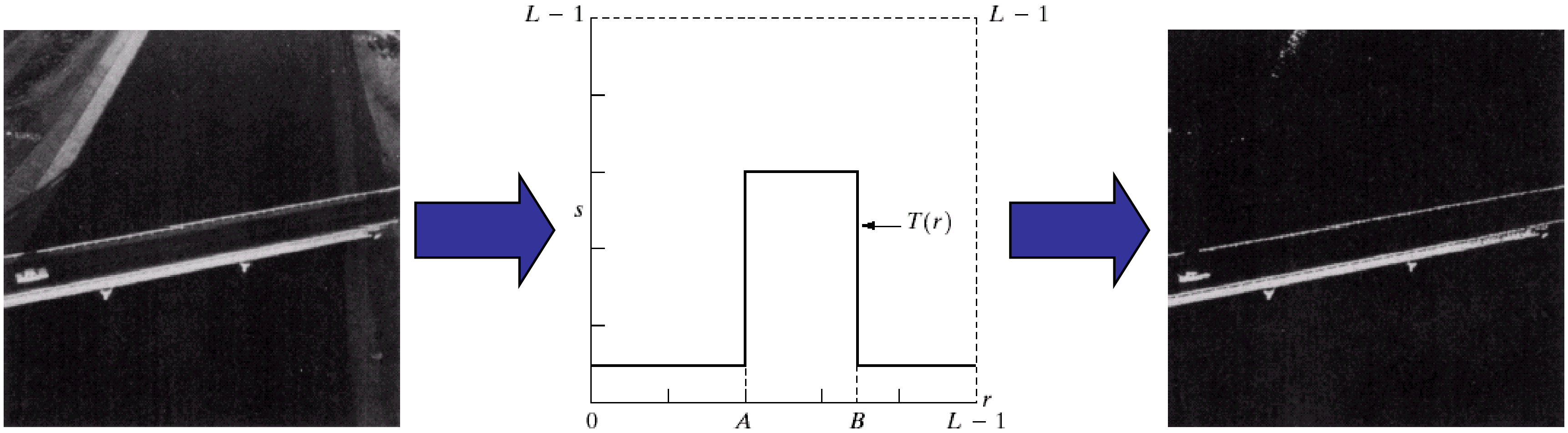


Gray Level Slicing

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

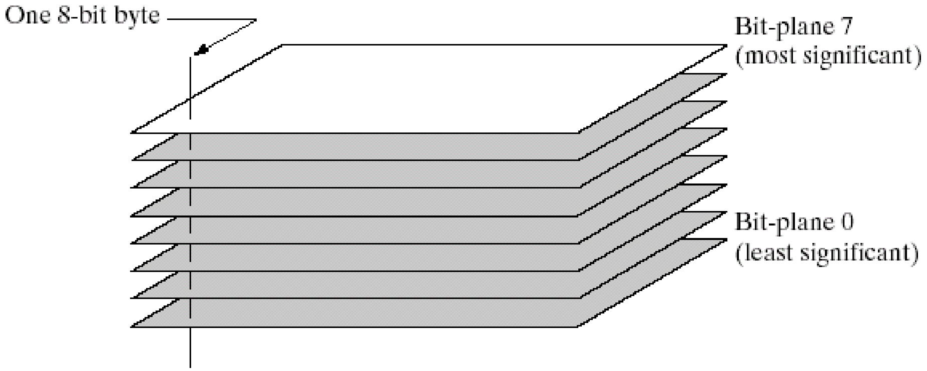
- Highlights a specific range of grey levels
  - Similar to thresholding
  - Other levels can be suppressed or maintained
  - Useful for highlighting features in an image

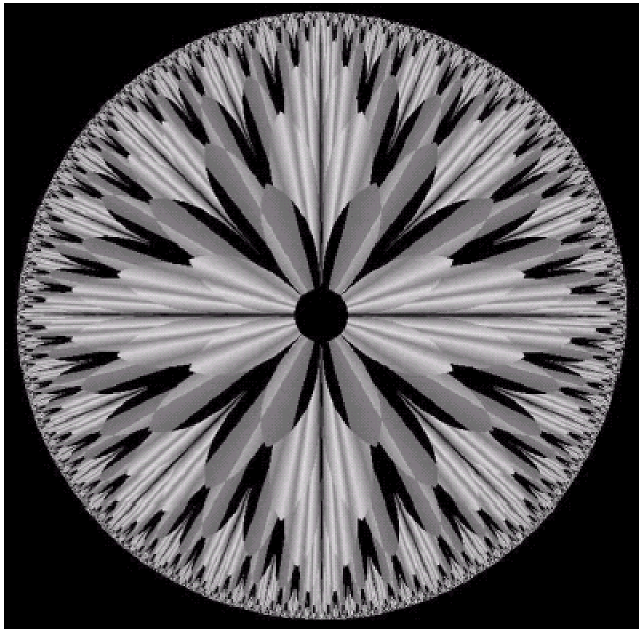




Bit Plane Slicing

- 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



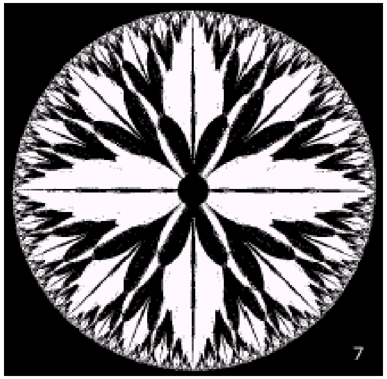


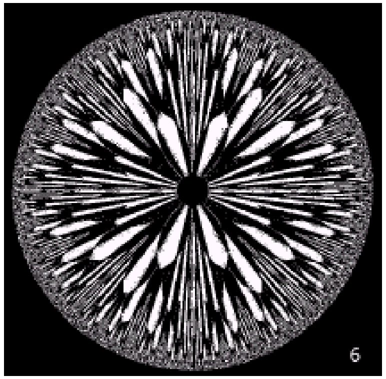
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)

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

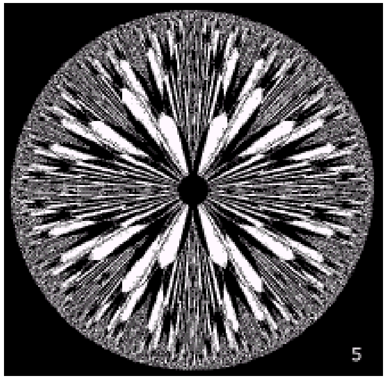
[10000000]

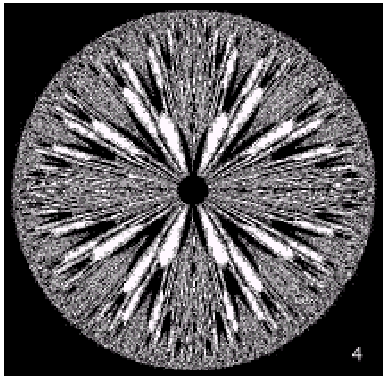


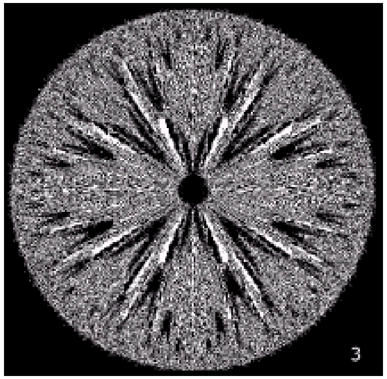


[01000000]

[00100000]

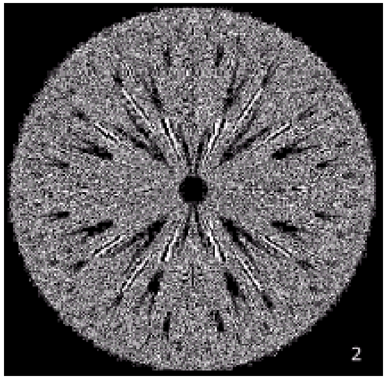


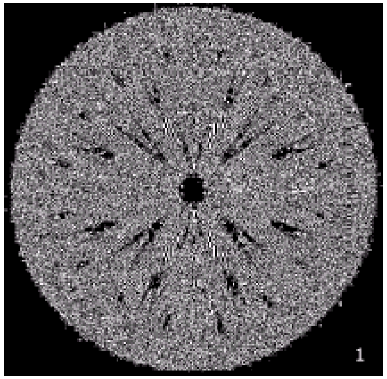


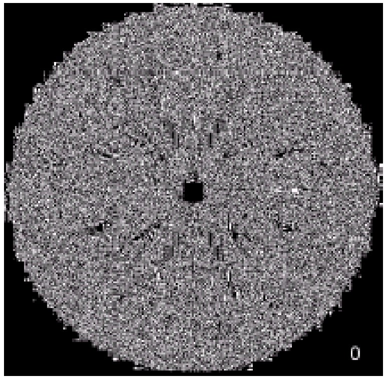


[00001000]

[00000100]



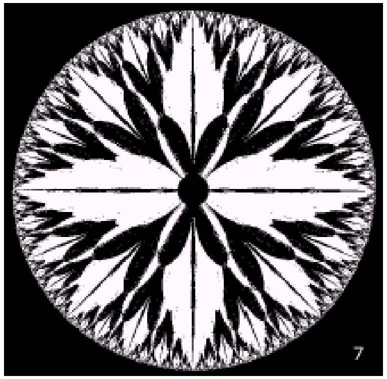


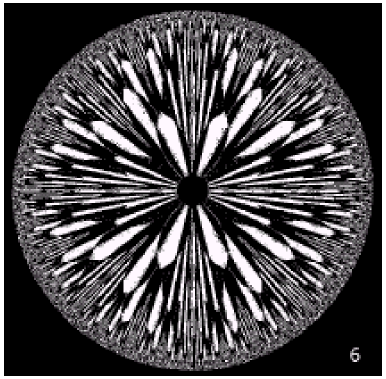


[00000001]

Bit Plane Slicing (cont…)

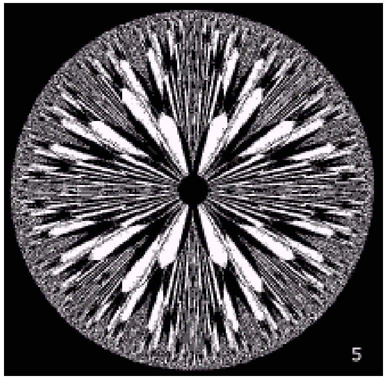
[10000000]

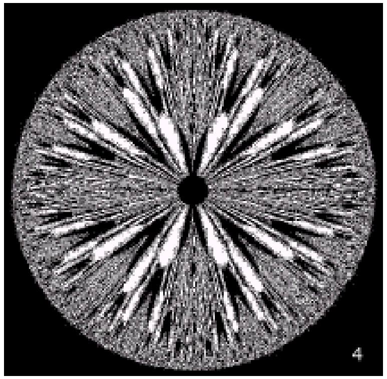


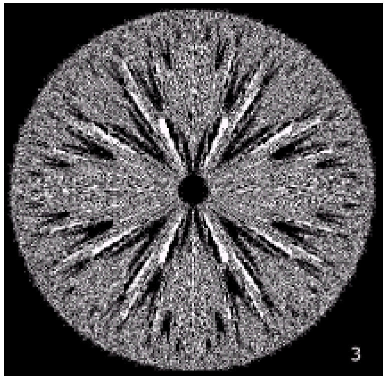


[01000000]

[00100000]

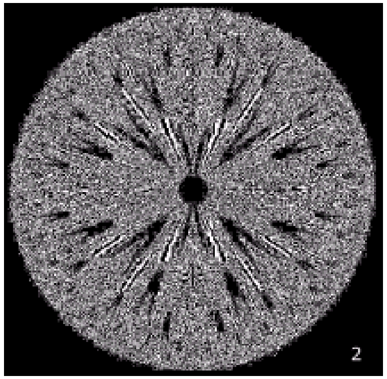


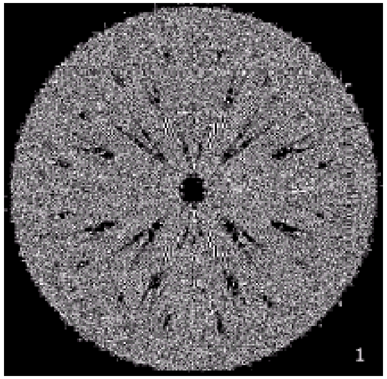


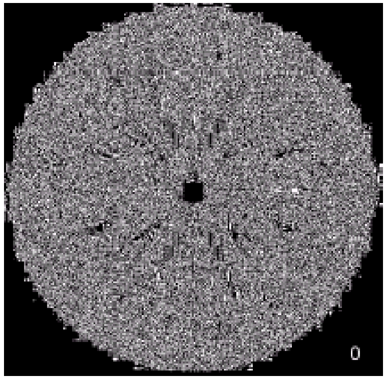


[00001000]

[00000100]



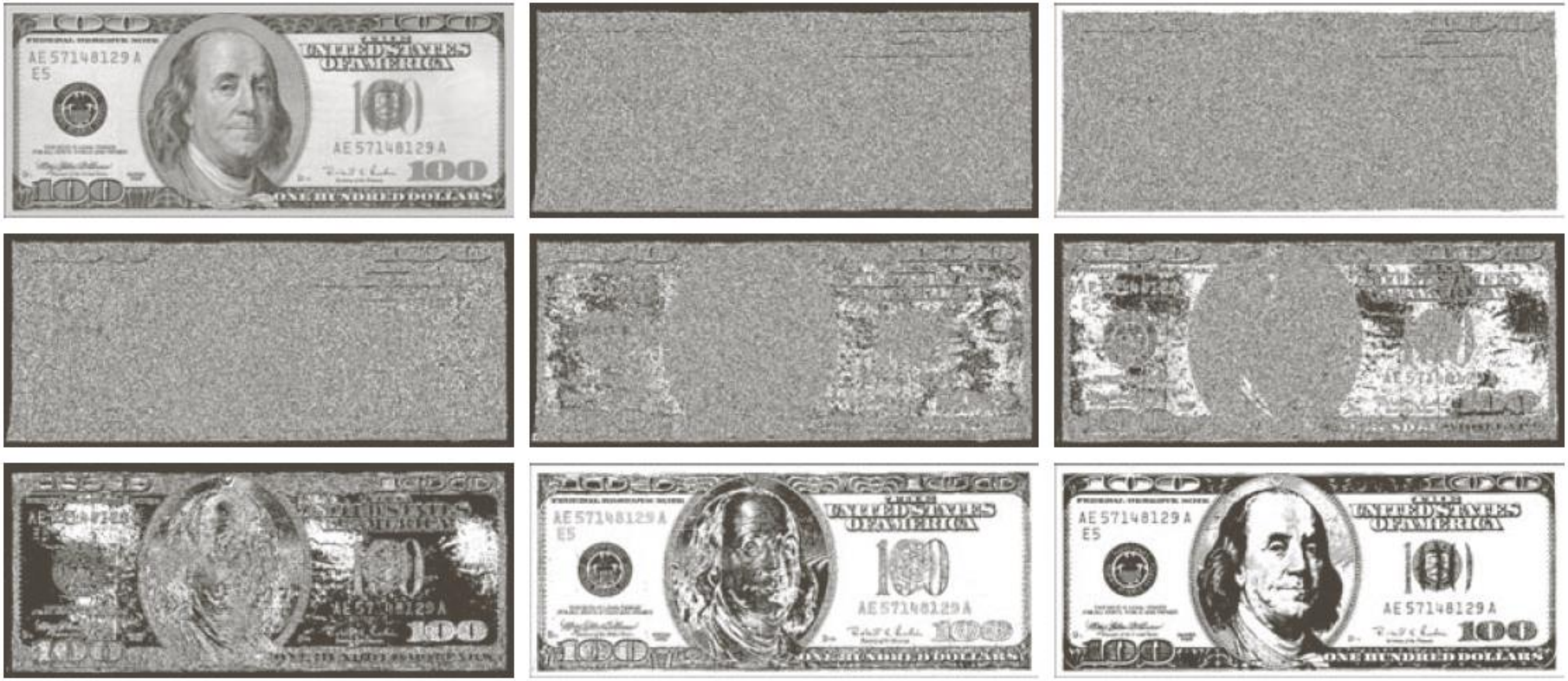




[00000001]

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…)

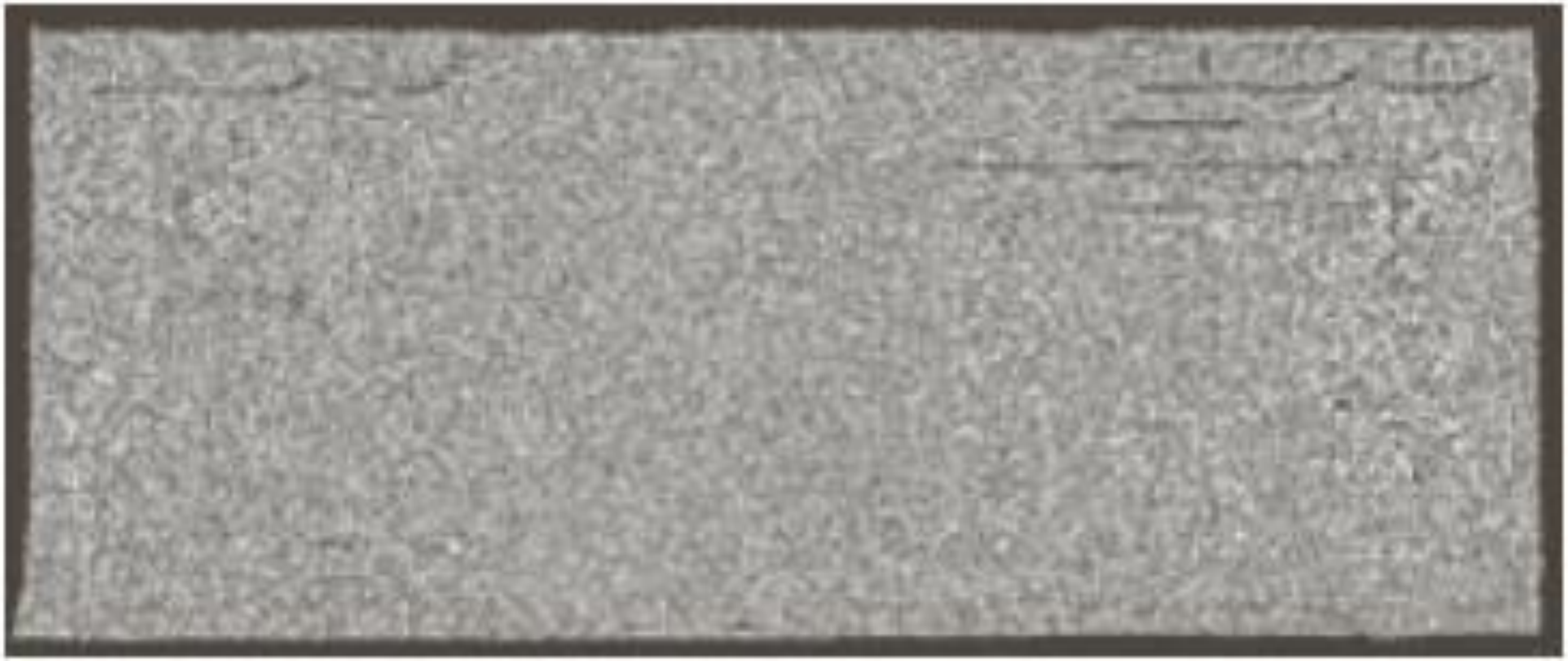
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)

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

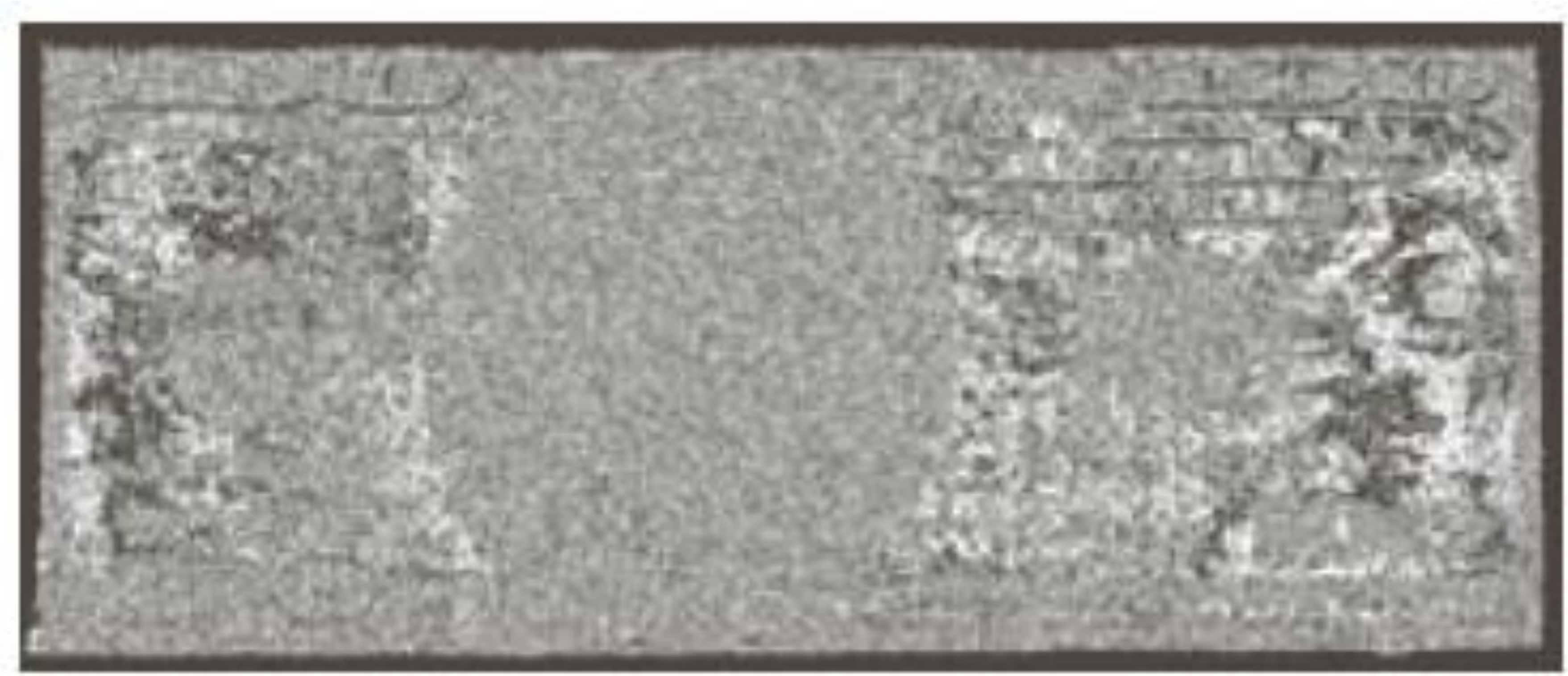
Bit Plane Slicing (cont…)

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



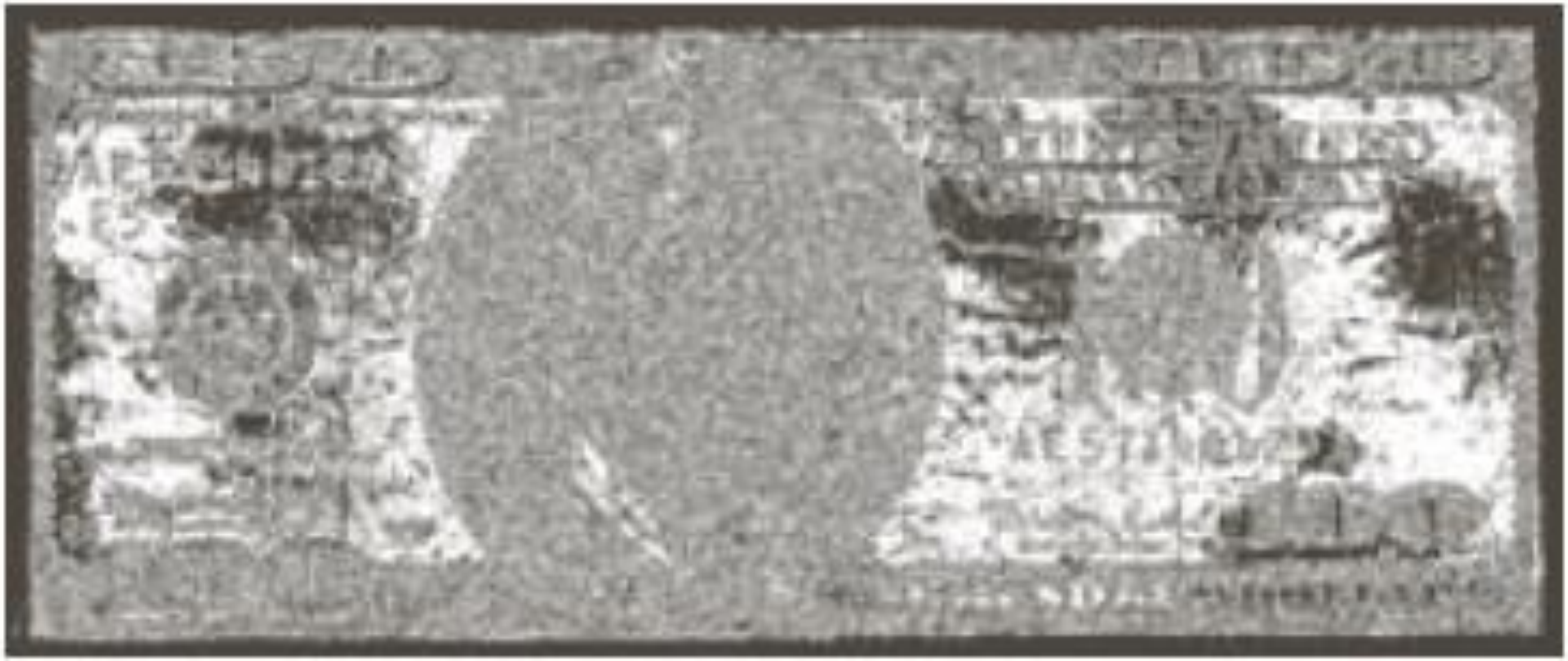
Bit Plane Slicing (cont…)

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



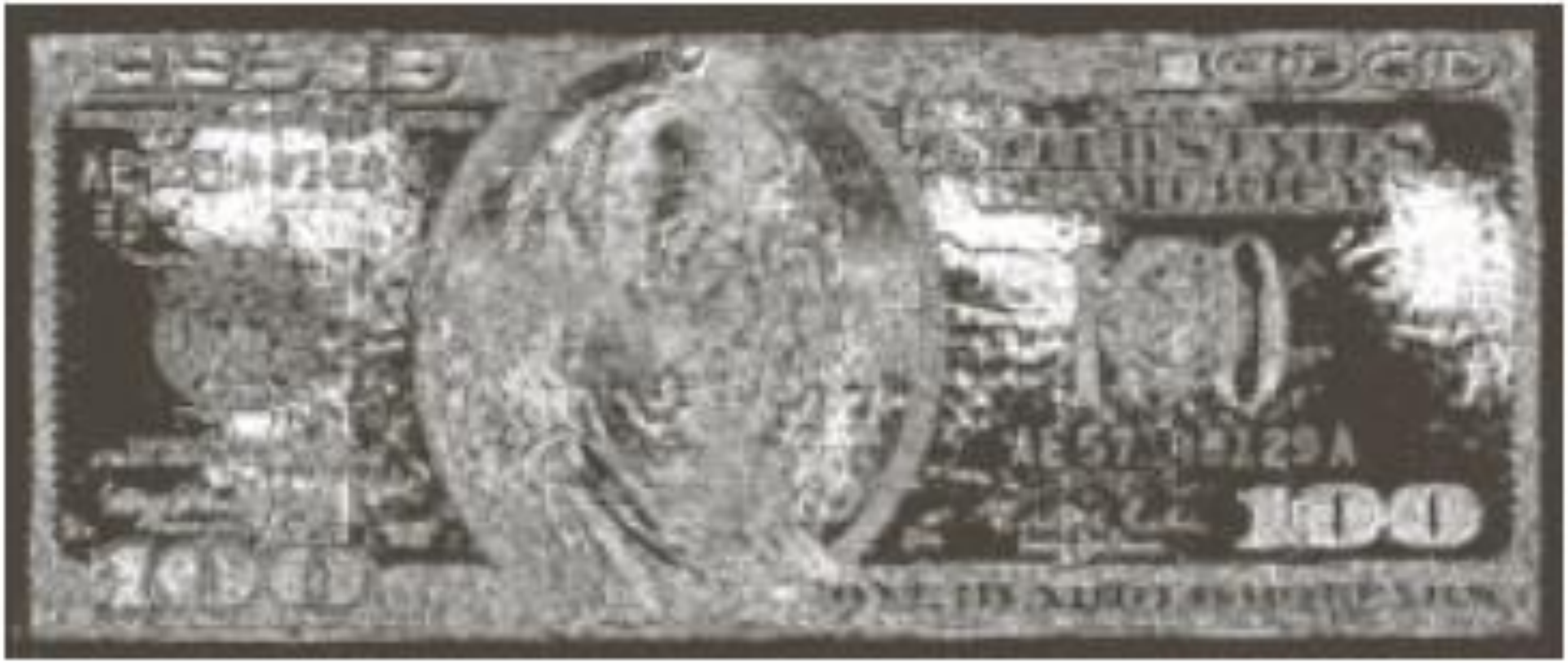
Bit Plane Slicing (cont…)

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



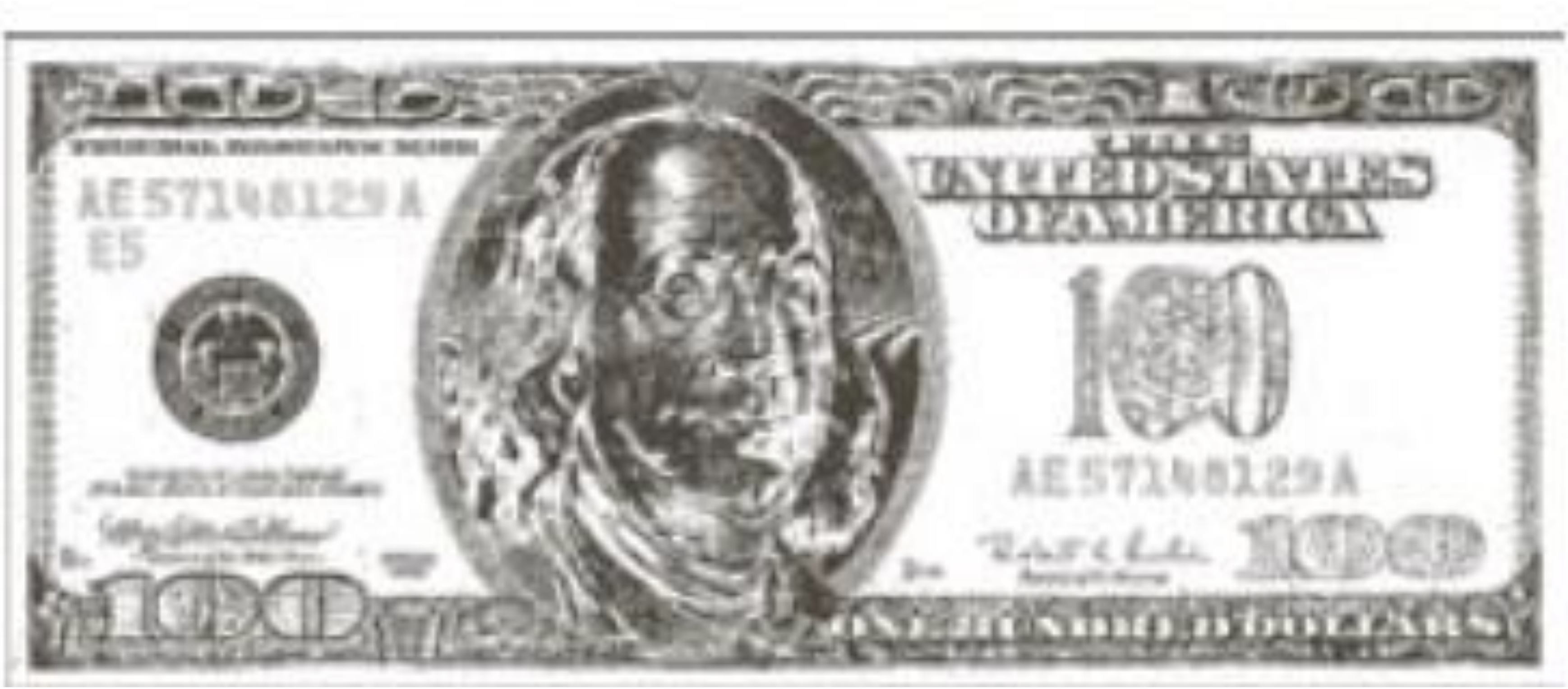
Bit Plane Slicing (cont…)

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



Bit Plane Slicing (cont…)

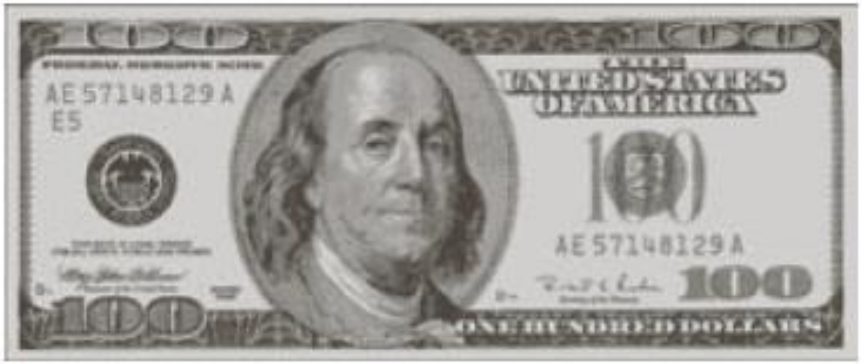
Images taken from Gonzalez & Woods, Digital Image Processing (2002)



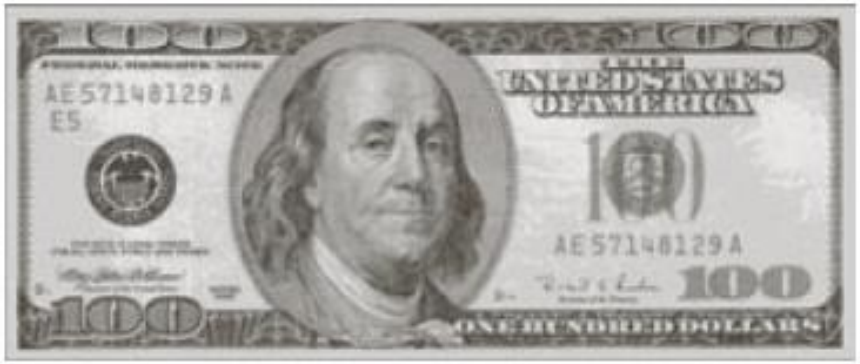
Bit Plane Slicing (cont…)

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

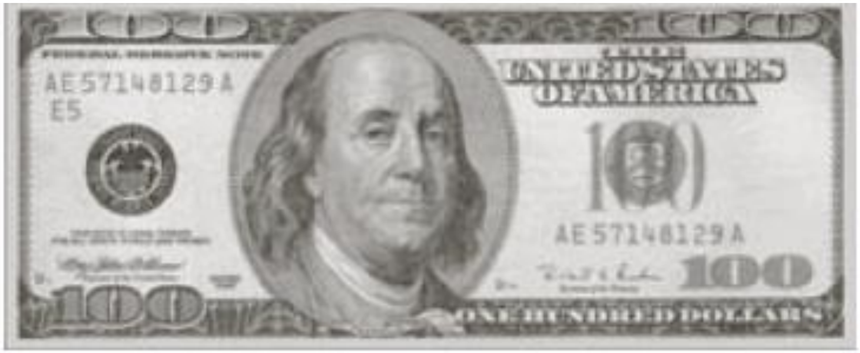
Bit Plane Slicing (cont…)



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)

Summary

- We have looked at different kinds of point processing image enhancement
- Next time we will start to look at neighbourhood operations – in particular *filtering* and *convolution*