CMPUT 206: DIGITAL IMAGE PROCESSING

POINT OPERATIONS

Nilanjan Ray

Contents

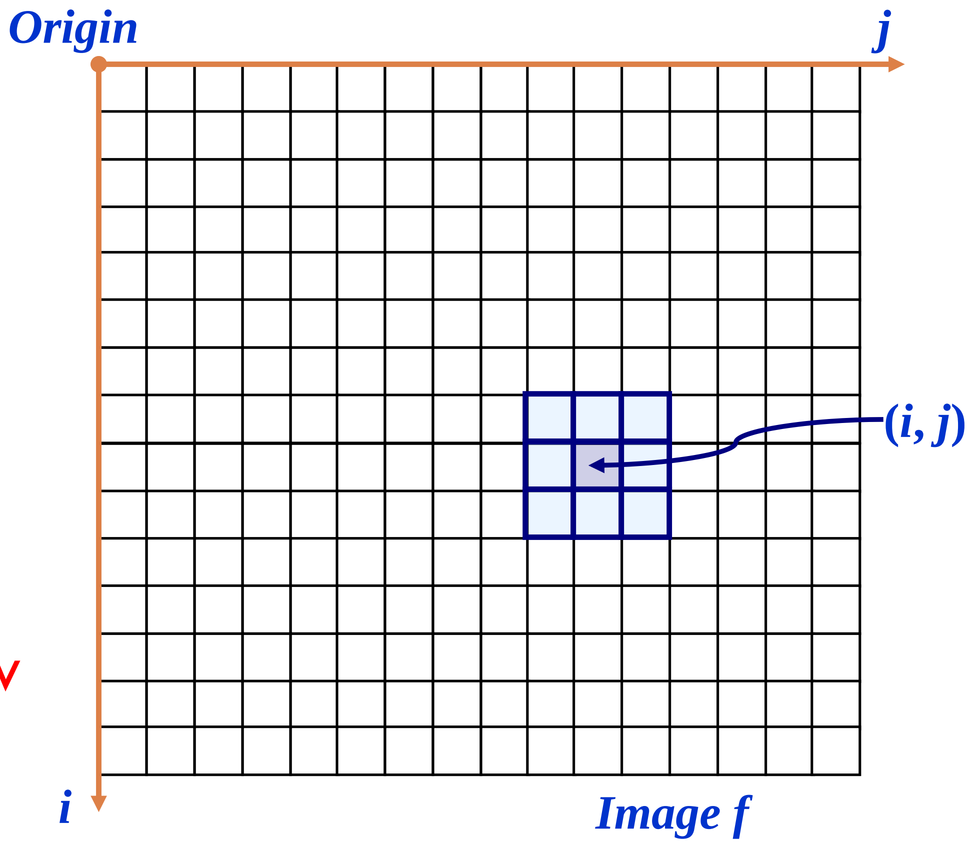
- In this lecture we will look at image point processing techniques, such as:
  - Negative images
  - Global thresholding
  - Logarithmic transformation
  - Power law transforms
  - Contrast stretching
  - Grey level slicing
  - Bit plane slicing

We have seen filtering before…

- It is a moving window operation
- It can be denoted by

$$g(i, j) = T[\, f\,(i, j)]$$

- where $f\,(i, j)$ is the input image, $g(i, j)$ is the processed/output image and T is some operator defined over some neighbourhood/window of (i, j)

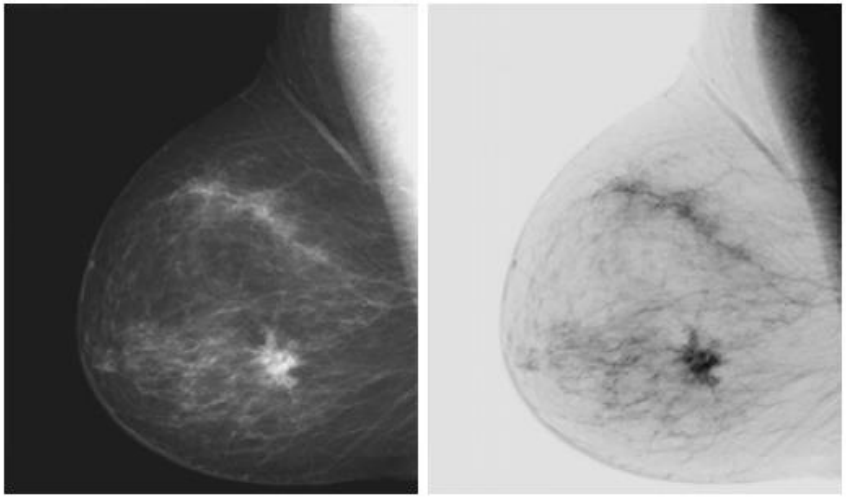


Point processing / Point operation

- The simplest form of T occur when the neighbourhood is of size 1×1 $i.e.$ simply the pixel itself
- This technique often 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: Negative images

- Denote [0, L-1] intensity levels of the image.
- Image negative is obtained by $s = L-1-r$
- 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



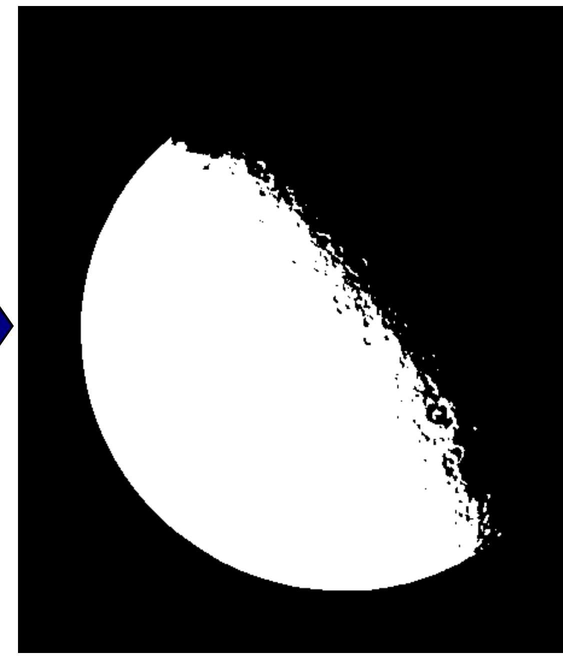
a b

FIGURE 3.4 (a) Original digital mammogram. (b) Negative image obtained using the negative transformation in Eq. (3.2-1). (Courtesy of G.E. Medical Systems.)

Point Processing: Global thresholding

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

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

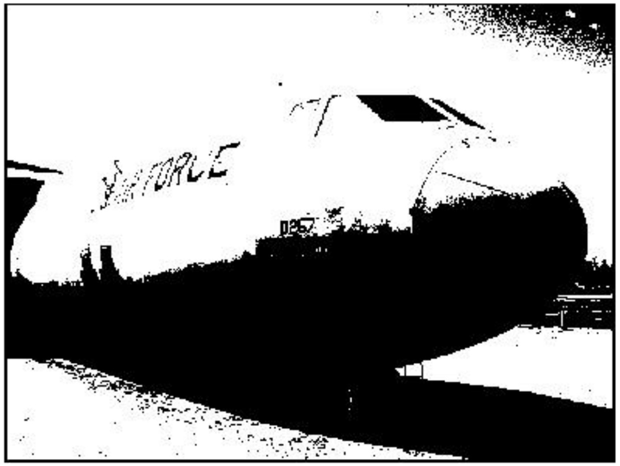


Input intensity value

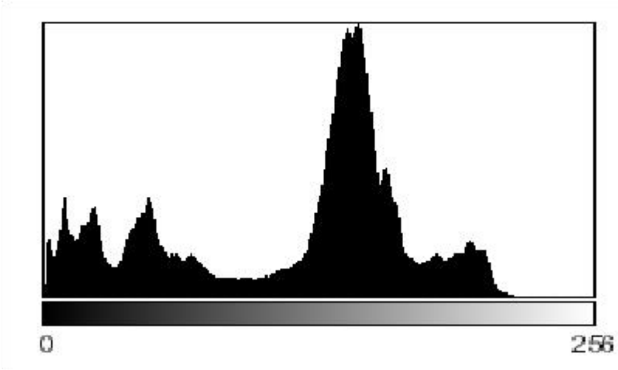
Output intensity value

Global thresholding

(a)



(b)

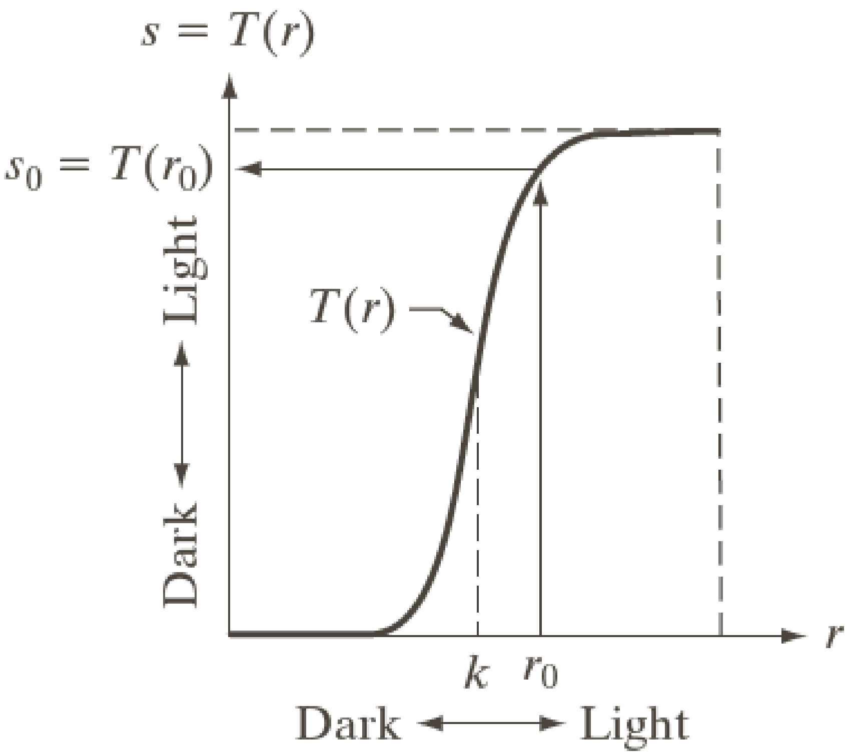


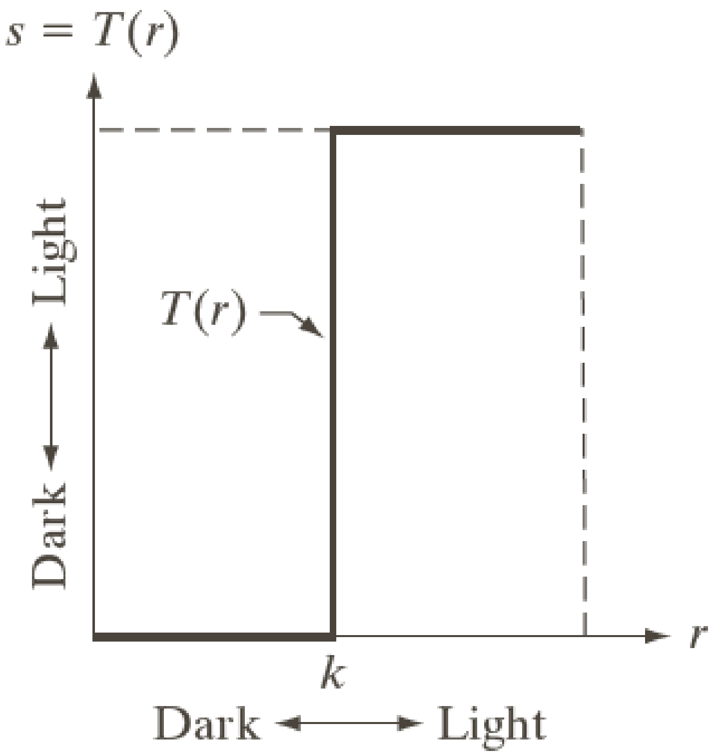
(c)

(d)

Intensity transformations

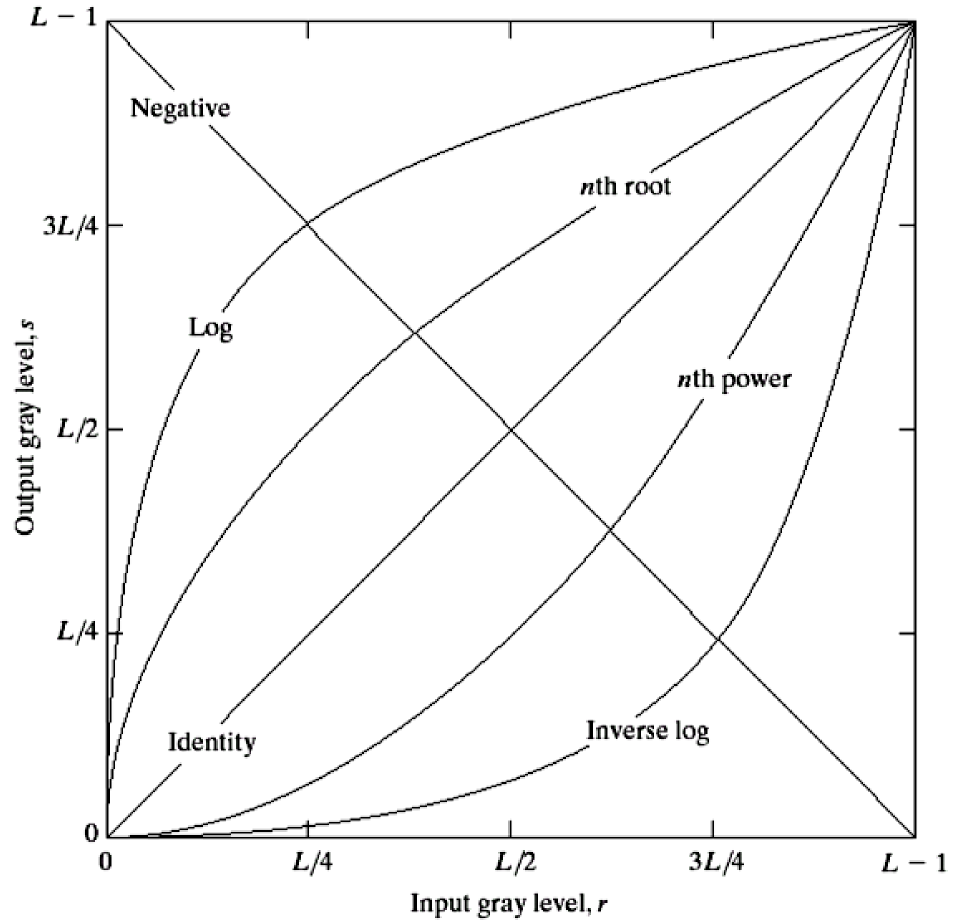
The following transformation *T* produces an image of higher contrast than the original by darkening the levels below *k* and brightening the level above *k* in input image.





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

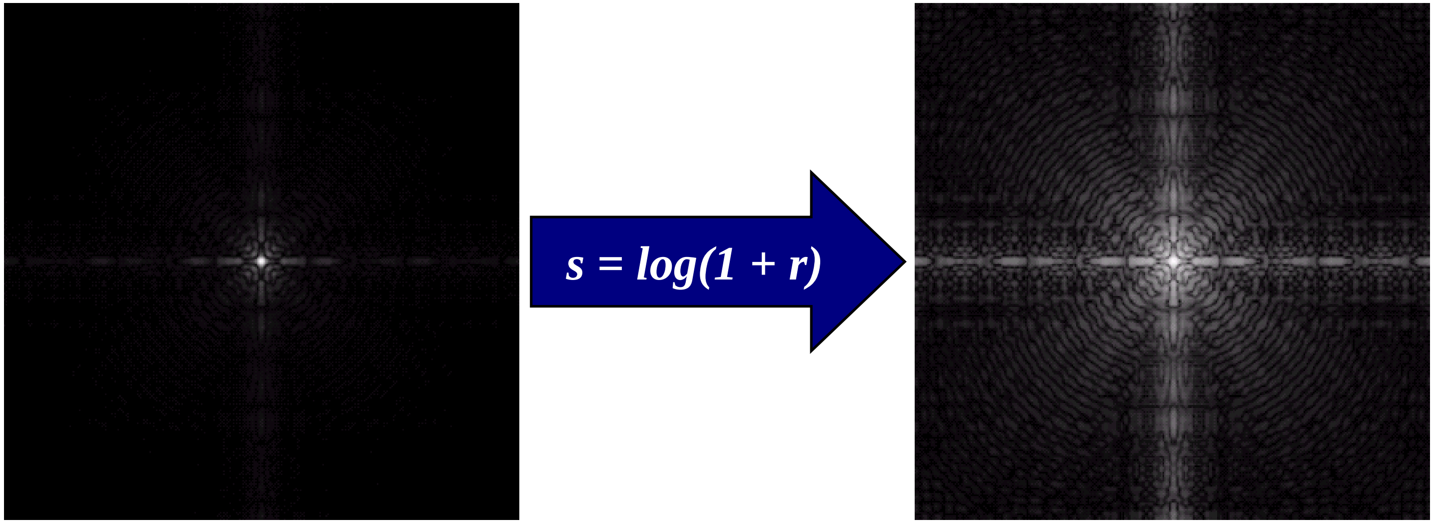


Log transformations

- 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 opposite is true of the inverse log transformation

Log transformations (cont…)

- Log transformation compresses the dynamic range of images with large variations in pixel values
- In the following example the Fourier transform of an image is put through a log transform to reveal more detail



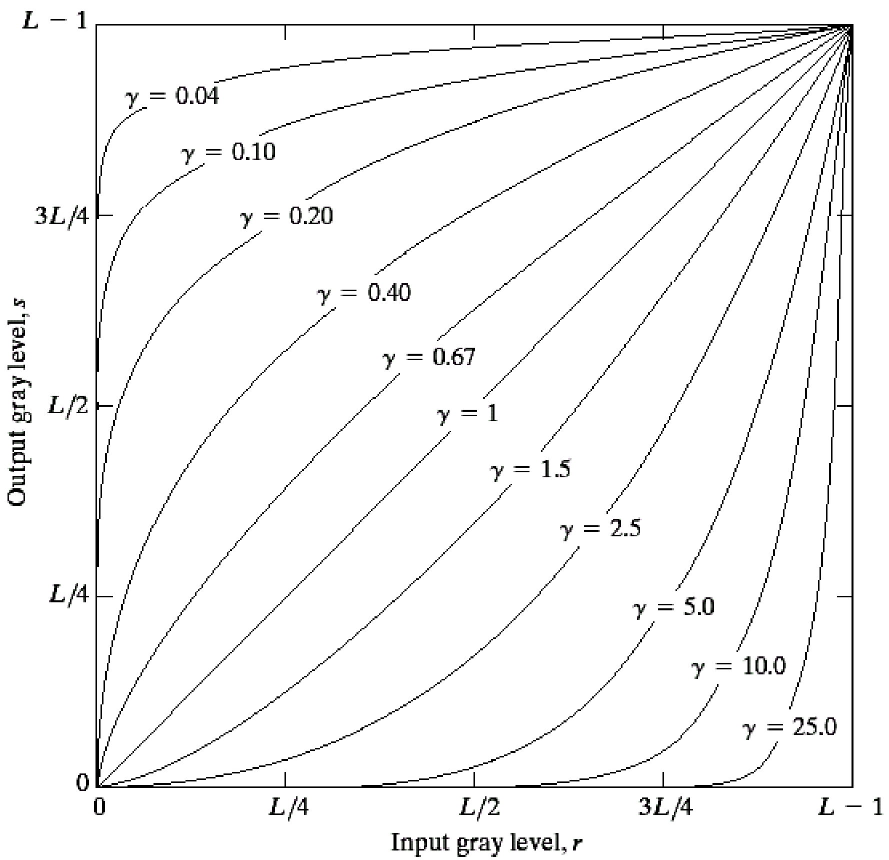
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Power law transformations

- Power law transformations have the basic form

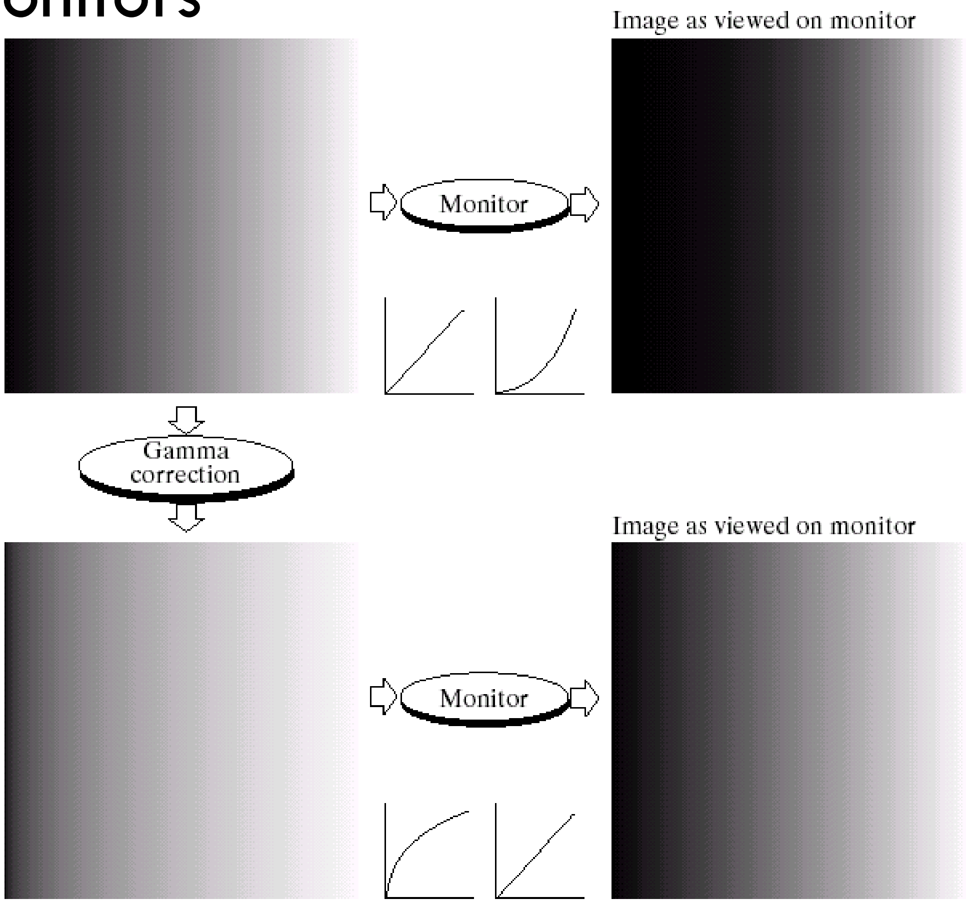
$$s = c * r^{\gamma}$$

- Map a narrow range of dark input values into a wider range of output values or vice versa
- A family of possible transformation curves are Obtained by Varying γ



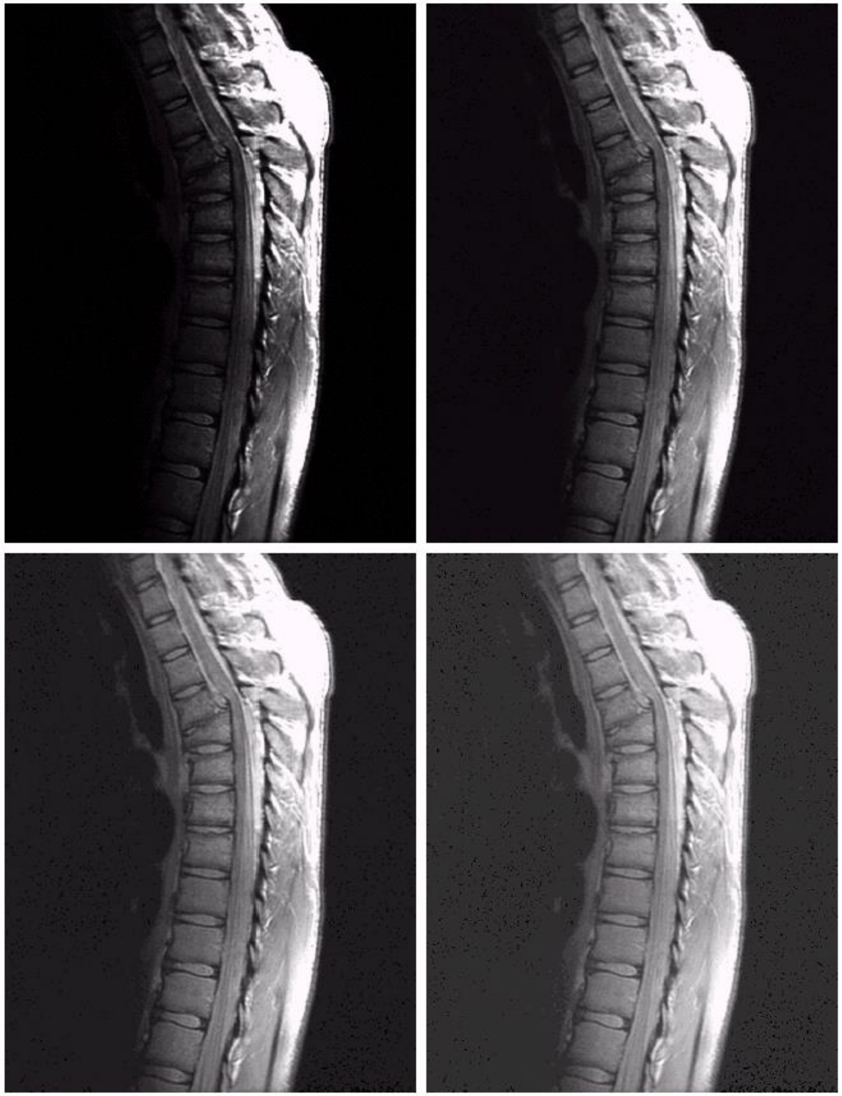
Gamma correction

- 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



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

Power law example



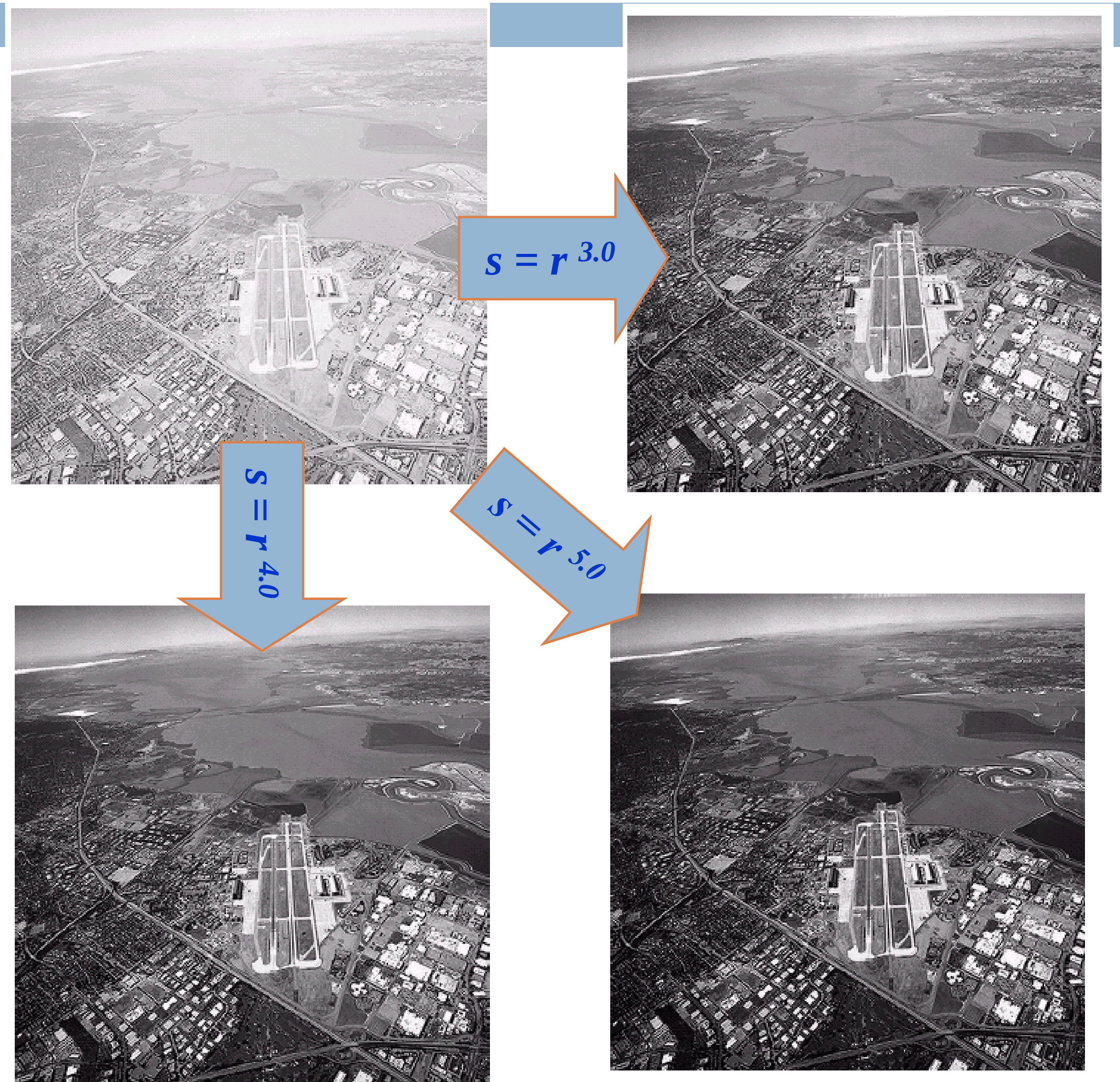
a b
c d

FIGURE 3.8
(a) Magnetic resonance (MR) image of a fractured human spine. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 0.6, 0.4,$ and 0.3, respectively. (Original image for this example courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

Power law transformations (cont…)

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

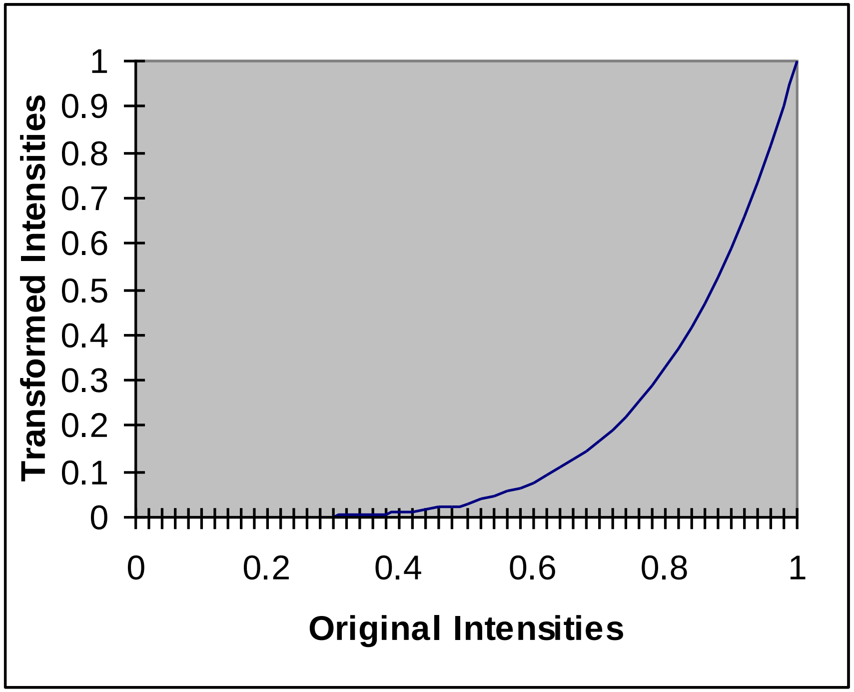
- An aerial photo of a runway is shown
- This time power law transforms are used to darken the image
- Different curves highlight different detail



Power Law Example

Power law example (cont…)

$$\gamma = 5.0$$

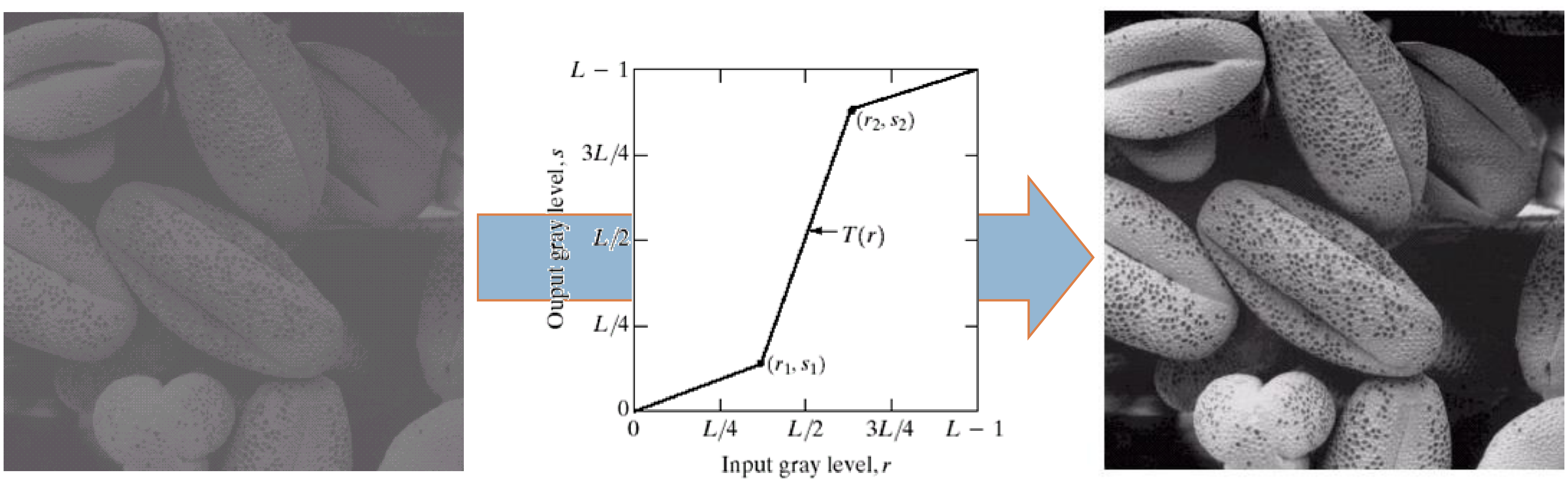


Piecewise linear transformation

- A complementary approach to the previously discussed methods.
- Principle advantage is that the piecewise linear functions can be arbitrarily complex.
- Principle disadvantage is that their specification requires considerably more user input.

Piecewise linear transformation…

- One of the simplest piecewise linear functions is a contrast-stretching transformation.
- The idea is to increase the dynamic range of the gray levels in the image being processed



Automatic contrast adjustment

Suppose a_{low} and a_{high} are the lowest and the highest pixel values in the current image.

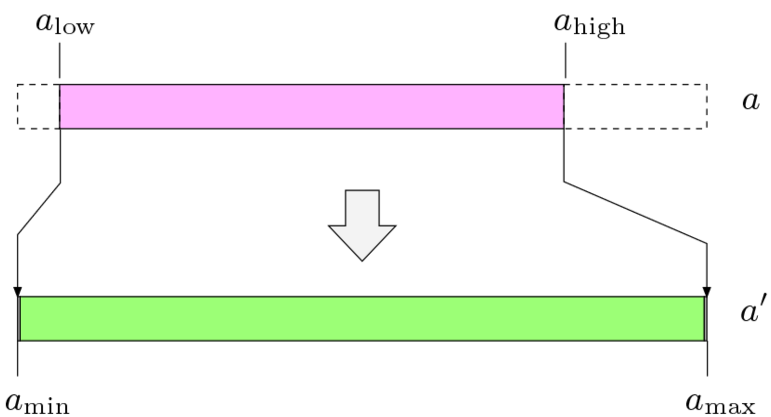
Further, suppose $[a_{min}, a_{max}]$ is the full range of intensity available. Then, automatic contrast adjustment maps a_{low} to a_{min} and a_{high} to a_{max}. All in between values are mapped linearly as follows:

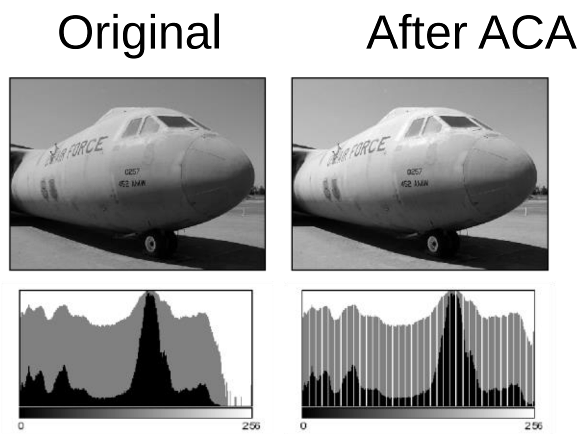
$$f_{\mathrm{ac}}(a) = a_{\mathrm{min}} + (a - a_{\mathrm{low}}) \cdot \frac{a_{\mathrm{max}} - a_{\mathrm{min}}}{a_{\mathrm{high}} - a_{\mathrm{low}}}$$

For an 8-bit image with $a_{min} = 0$, and $a_{max} = 255$, the above equation becomes:

$$f_{\mathrm{ac}}(a) = (a - a_{\mathrm{low}}) \cdot \frac{255}{a_{\mathrm{high}} - a_{\mathrm{low}}}$$

Automatic contrast adjustment….



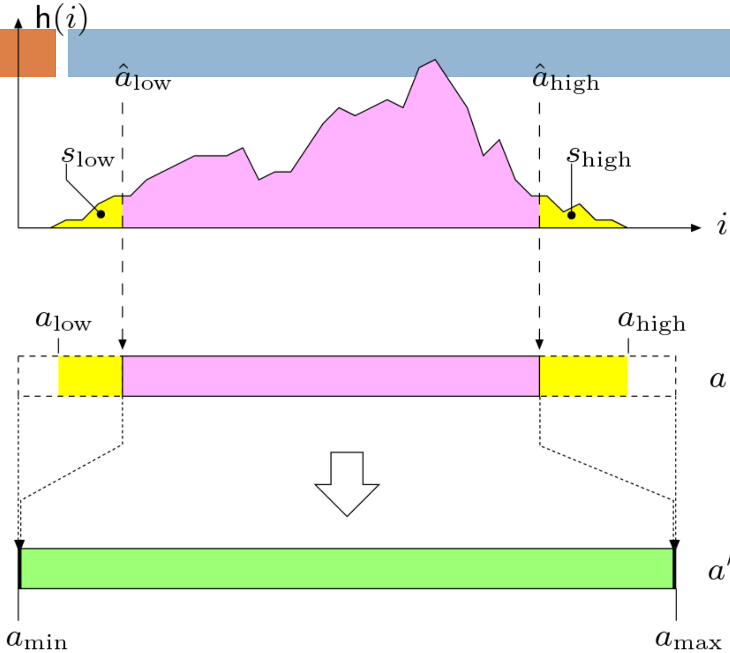


(a) (b)

Modified automatic contrast adjustment

- In practice, the linear mapping function can be affected by a few extreme pixel values (sometimes called outlier values)
- This can be avoided by saturating a fixed percentage (s_{low}, s_{high}) of pixels at the lower and upper ends of the target intensity range

Modified automatic contrast adjustment…



MAC adjustment in picture

This is how $\hat{a}_{\text{low}}$ and $\hat{a}_{\text{high}}$ are computed with the help of cumulative histogram *H*(*i*):

$$\hat{a}_{\text{low}} = \min\{\, i \mid \mathsf{H}(i) \geq M \cdot N \cdot s_{\text{low}} \}$$

$$\hat{a}_{\text{high}} = \max\{\, i \mid \mathsf{H}(i) \leq M \cdot N \cdot (1 - s_{\text{high}}) \}$$

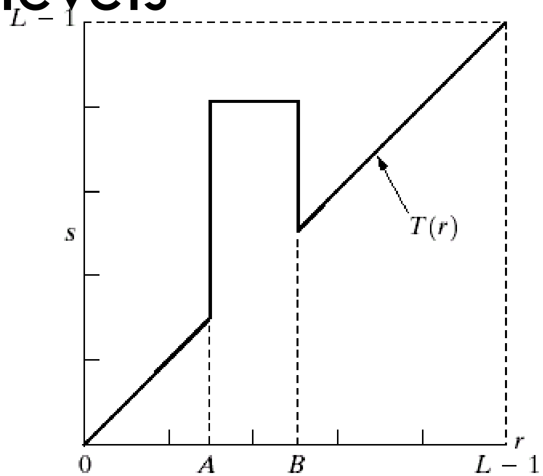
The mapping function *f* looks like:

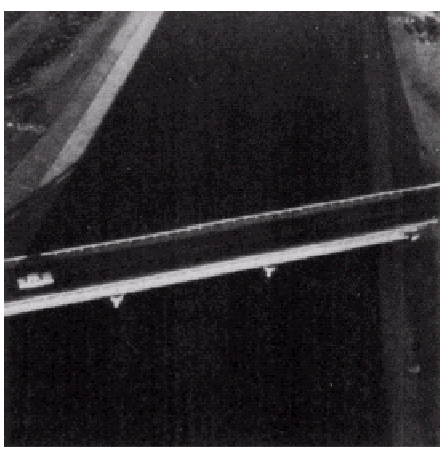
$$f_{\text{mac}}(a) = \begin{cases} a_{\min} & \text{for } a \leq \hat{a}_{\text{low}} \\ a_{\min} + \left(a - \hat{a}_{\text{low}}\right) \cdot \dfrac{a_{\max} - a_{\min}}{\hat{a}_{\text{high}} - \hat{a}_{\text{low}}} & \text{for } \hat{a}_{\text{low}} < a < \hat{a}_{\text{high}} \\ a_{\max} & \text{for } a \geq \hat{a}_{\text{high}} \end{cases}$$

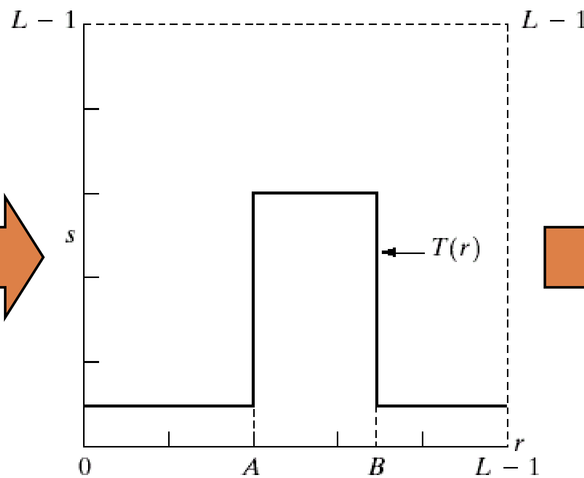
Graylevel slicing

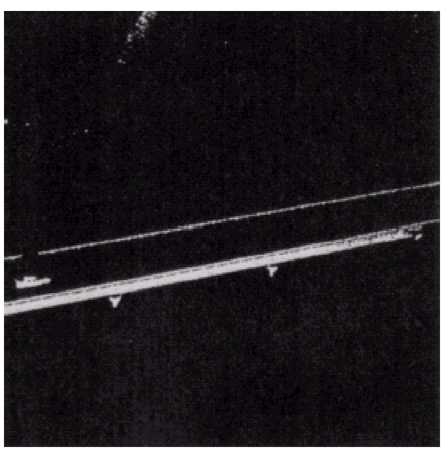
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

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



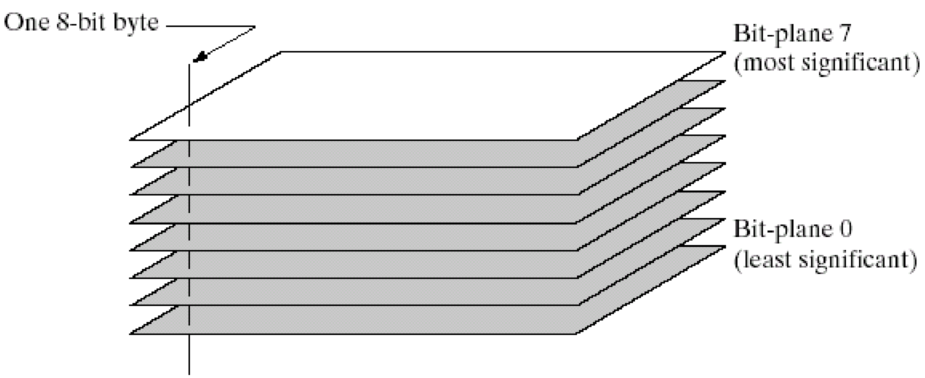


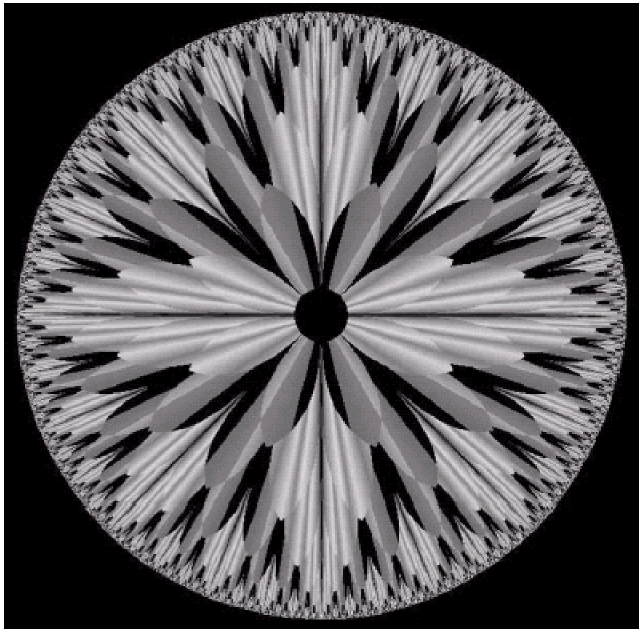




Bit plane slicing

- Instead of highlighting graylevel ranges, highlighting the contribution made to total image appearance by specific bits might be desired
  - Higher-order bits usually contain most of the significant visual information
  - Lower-order bits contain subtle details





Bit plane slicing…

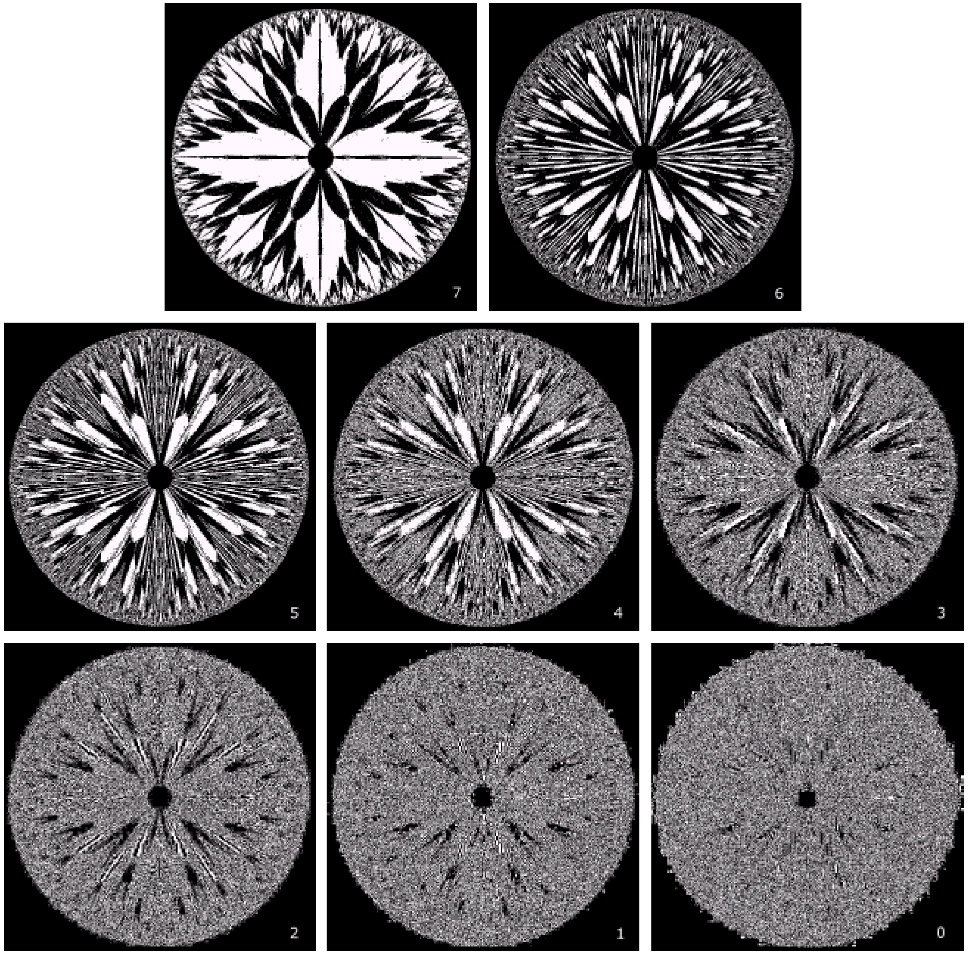


FIGURE 3.14 The eight bit planes of the image in Fig. 3.13. The number at the bottom, right of each image identifies the bit plane.