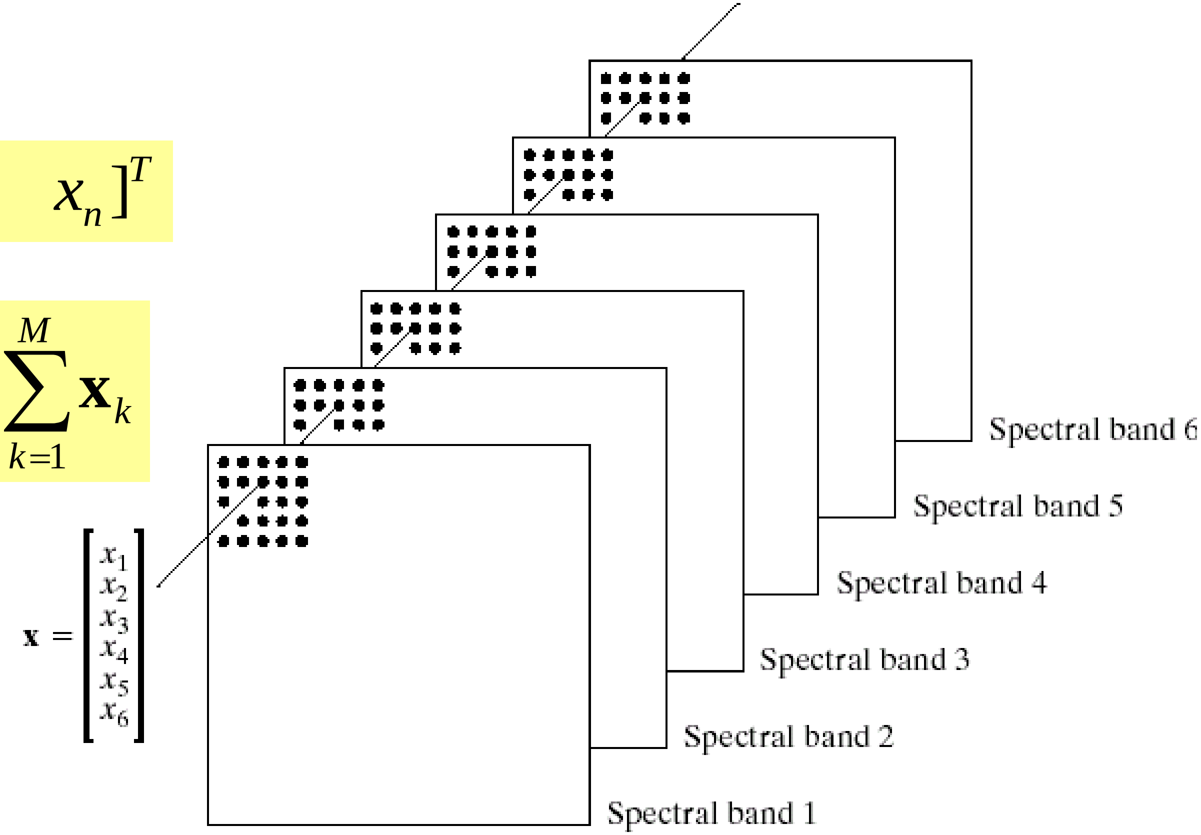
Digital Image Processing:

Hotelling transformation

The Hotelling transforms has several useful properties that make it an important tool for image processing.

Let $\mathbf{x} = [x_1 \quad x_2 \quad \dots \quad x_n]^T$

Mean: $\mathbf{m}_x = E\{\mathbf{x}\} = \frac{1}{M}\sum_{k=1}^{M}\mathbf{x}_k$



Covariance matrix $\mathbf{C}_x = E\{(\mathbf{x}-\mathbf{m}_x)(\mathbf{x}-\mathbf{m}_x)^T\} = \frac{1}{M}\sum_{k=1}^{M}\mathbf{x}_k\mathbf{x}_k^T - \mathbf{m}_x\mathbf{m}_x^T$

Hotelling transformation

Let $\mathbf{y} = \mathbf{A}(\mathbf{x} - \mathbf{m}_x)$

Where A is created from eigenvectors of C_x as follows

Row 1 contain the 1st eigenvector with the largest eigenvalue.

Row 2 contain the 2nd eigenvector with the 2nd largest eigenvalue.

....

Then we get

$$\mathbf{m}_y = E\{\mathbf{y}\} = 0$$

$$\mathbf{C}_y = \mathbf{A}\mathbf{C}_x\mathbf{A}^T$$

and

$$\mathbf{C}_y = \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & \ldots & \lambda_n \end{bmatrix}$$

Then elements of $\mathbf{y} = \mathbf{A}(\mathbf{x} - \mathbf{m}_x)$ are uncorrelated. The component of y with the largest λ is called the principal component.

Eigenvector and Eigenvalue

Eigenvector and eigenvalue of Matrix **C** are defined as

Let **C** be a matrix of size NxN and **e** be a vector of size Nx1. If

$$\mathbf{Ce} = \lambda \mathbf{e}$$

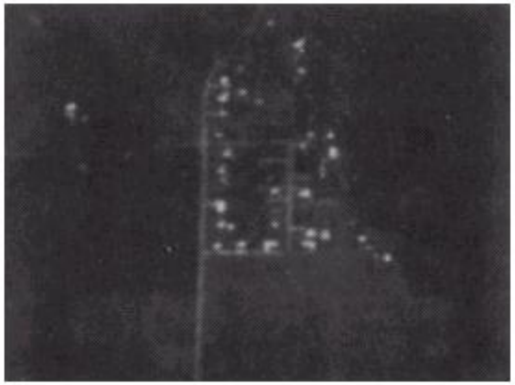
Where λ is a constant

We call **e** as an eigenvector and λ as eigenvalue of **C**

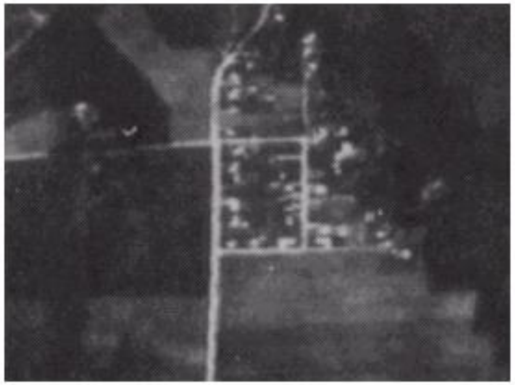
Example: Principal Components

6 spectral images from an airborne Scanner.

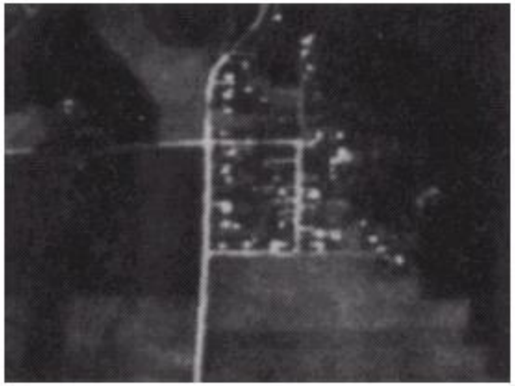
Channel	Wavelength band (microns)
1	0.40–0.44
2	0.62–0.66
3	0.66–0.72
4	0.80–1.00
5	1.00–1.40
6	2.00–2.60



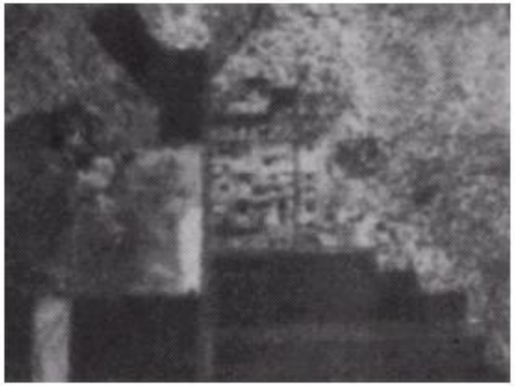
Channel 1



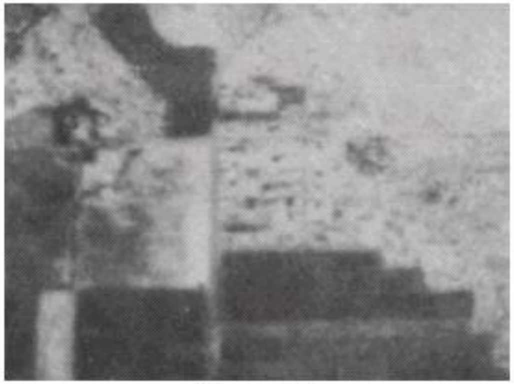
Channel 2



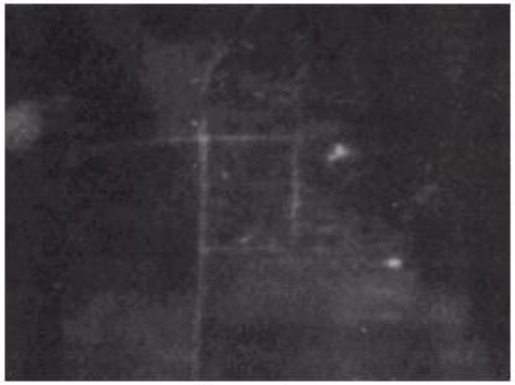
Channel 3



Channel 4

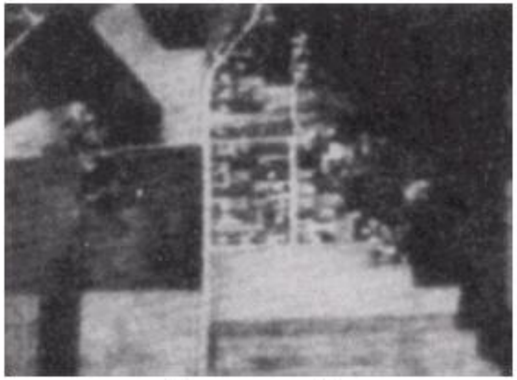


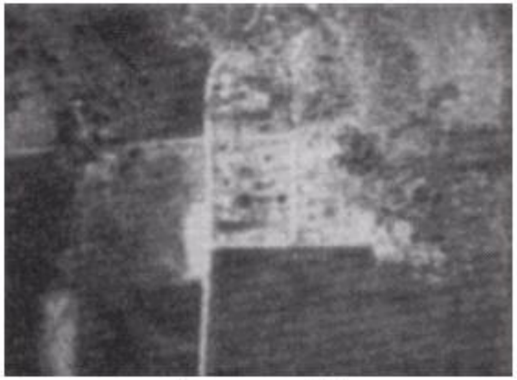
Channel 5

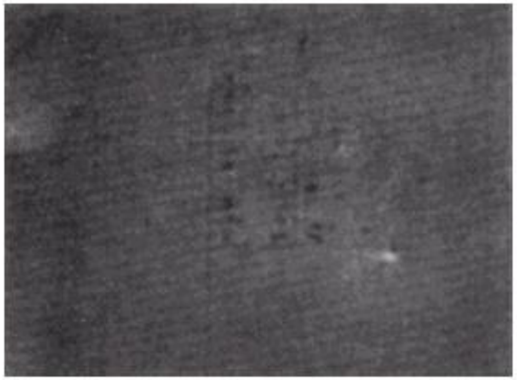


Channel 6

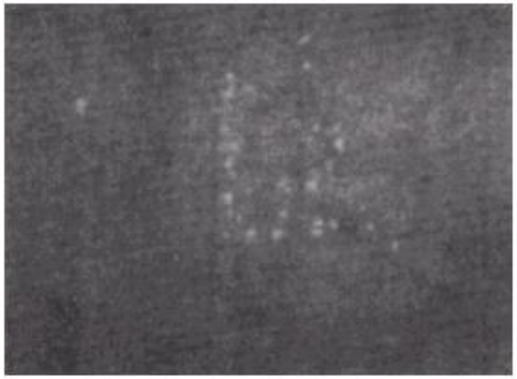
Example: Principal Components (cont.)

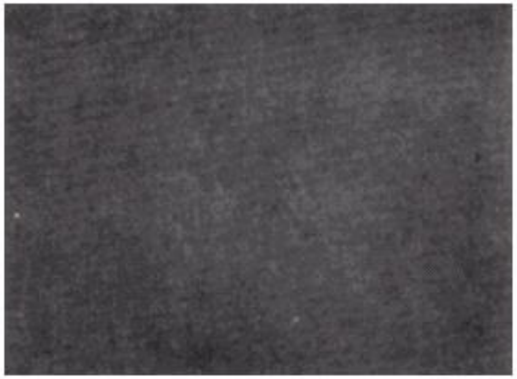

Component 1


Component 2


Component 3

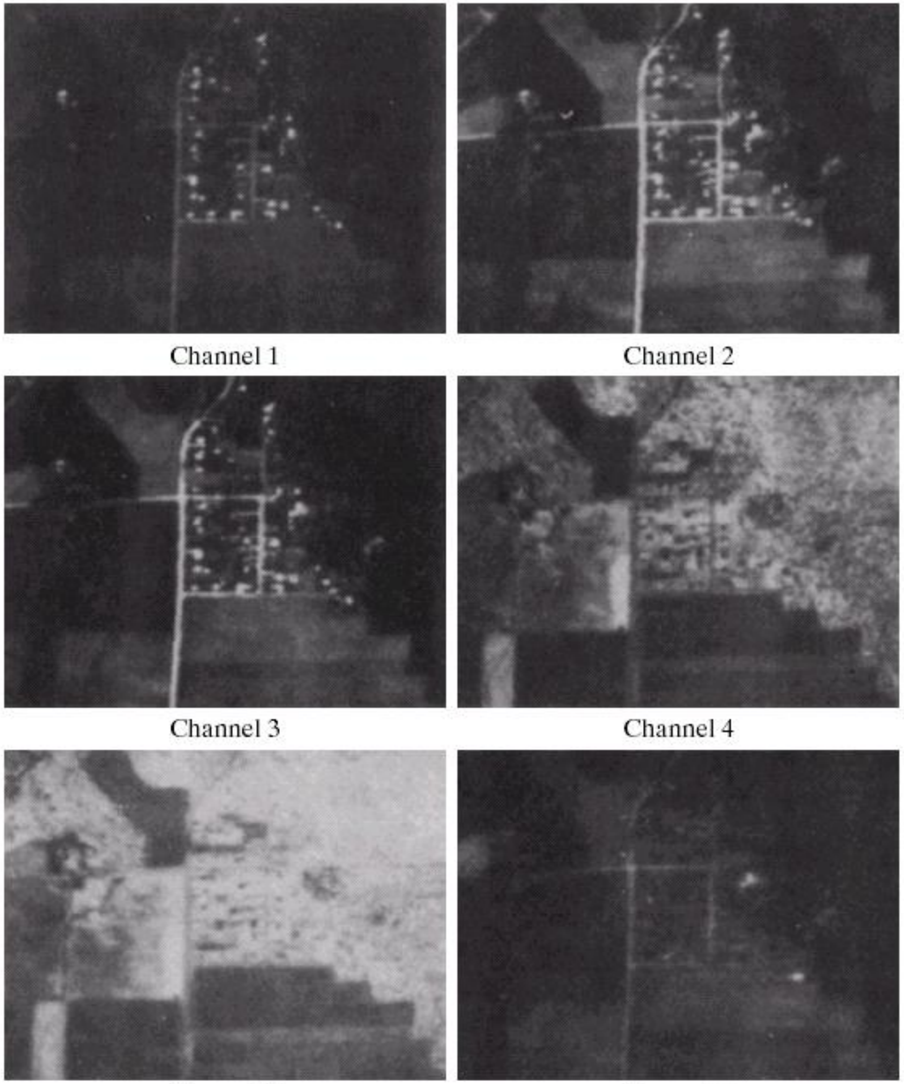
Component 4


Component 5


Component 6

Component	λ
1	3210
2	931.4
3	118.5
4	83.88
5	64.00
6	13.40

Example: Principal Components (cont.)



Channel 1

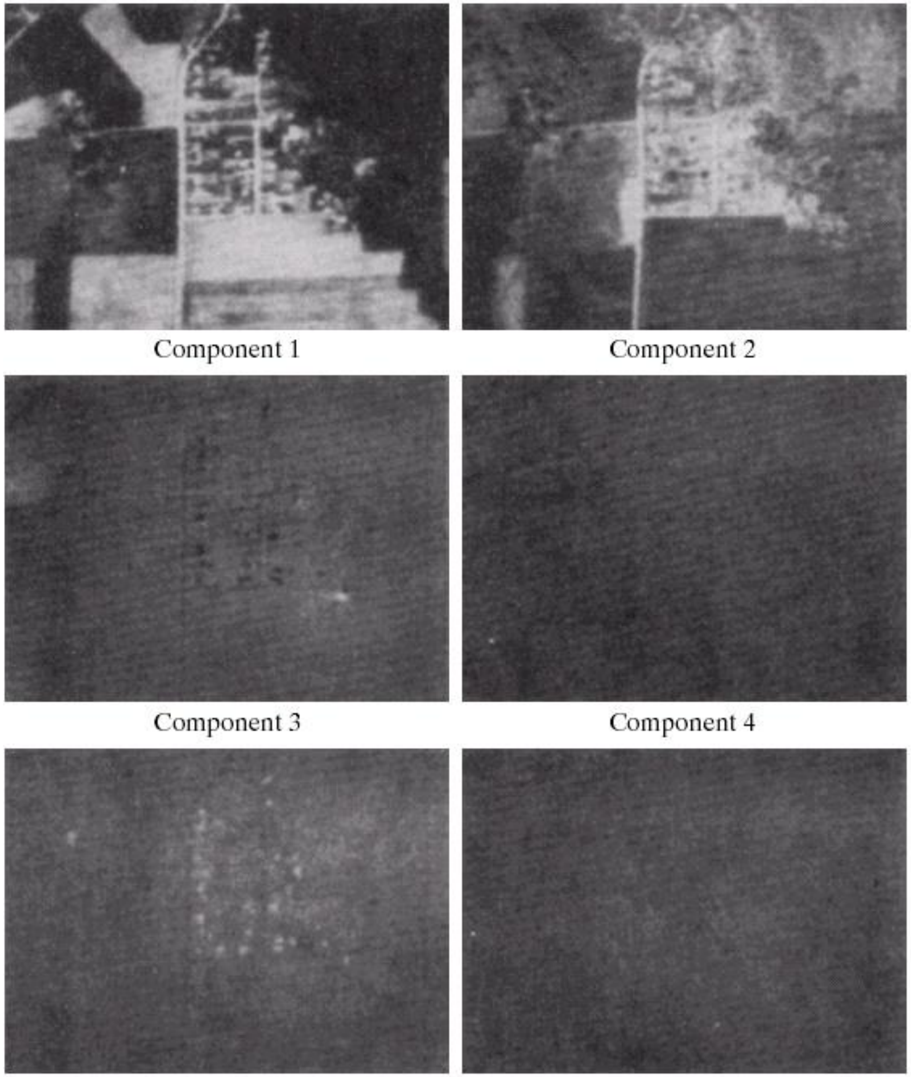
Channel 2

Channel 3

Channel 4

Channel 5

Channel 6



Component 1

Component 2

Component 3

Component 4

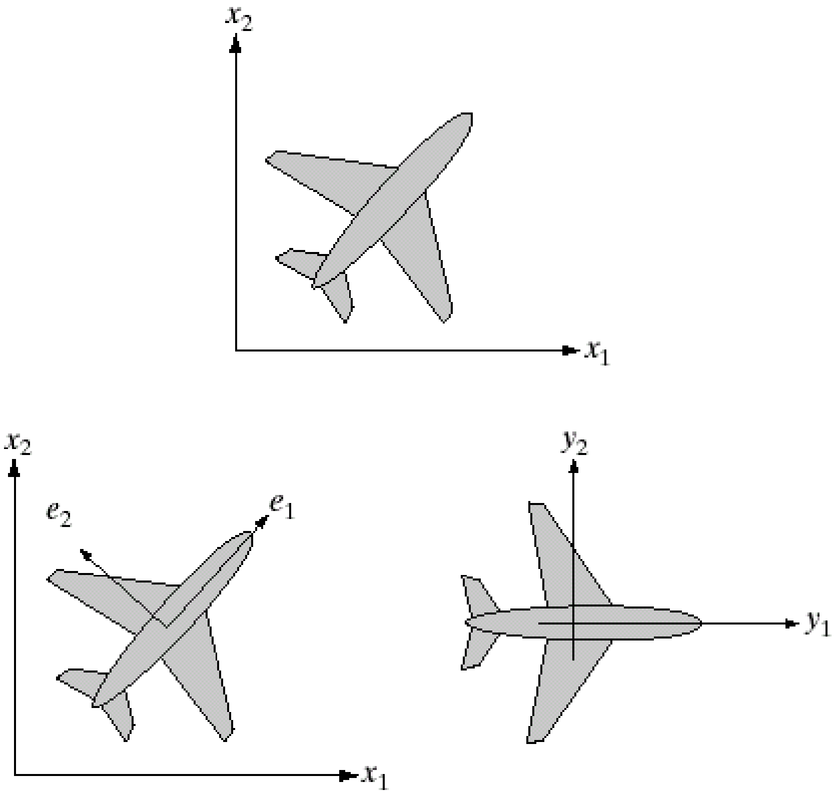
Component 5

Component 6

Original image

After Hotelling transform

Principal Components for Description



a
b c

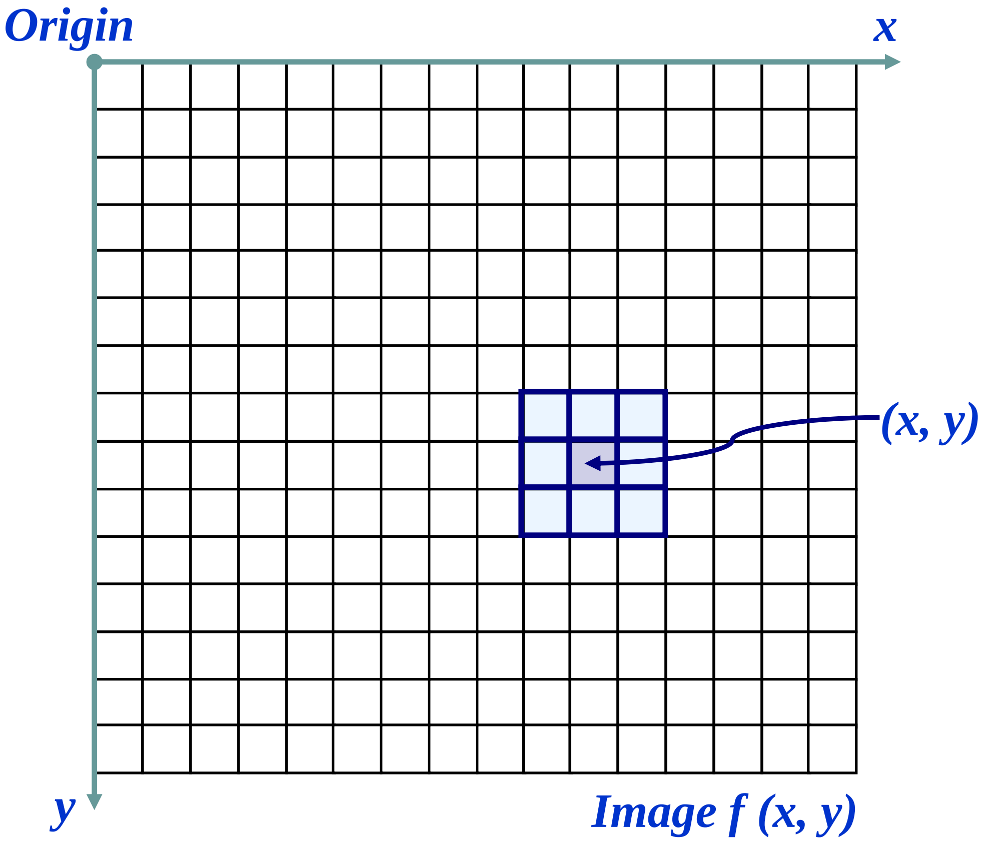
FIGURE 11.29 (a) An object. (b) Eigenvectors. (c) Object rotated by using Eq. (11.4-6). The net effect is to align the object along its eigen axes.

Basic Spatial Domain Image Enhancement

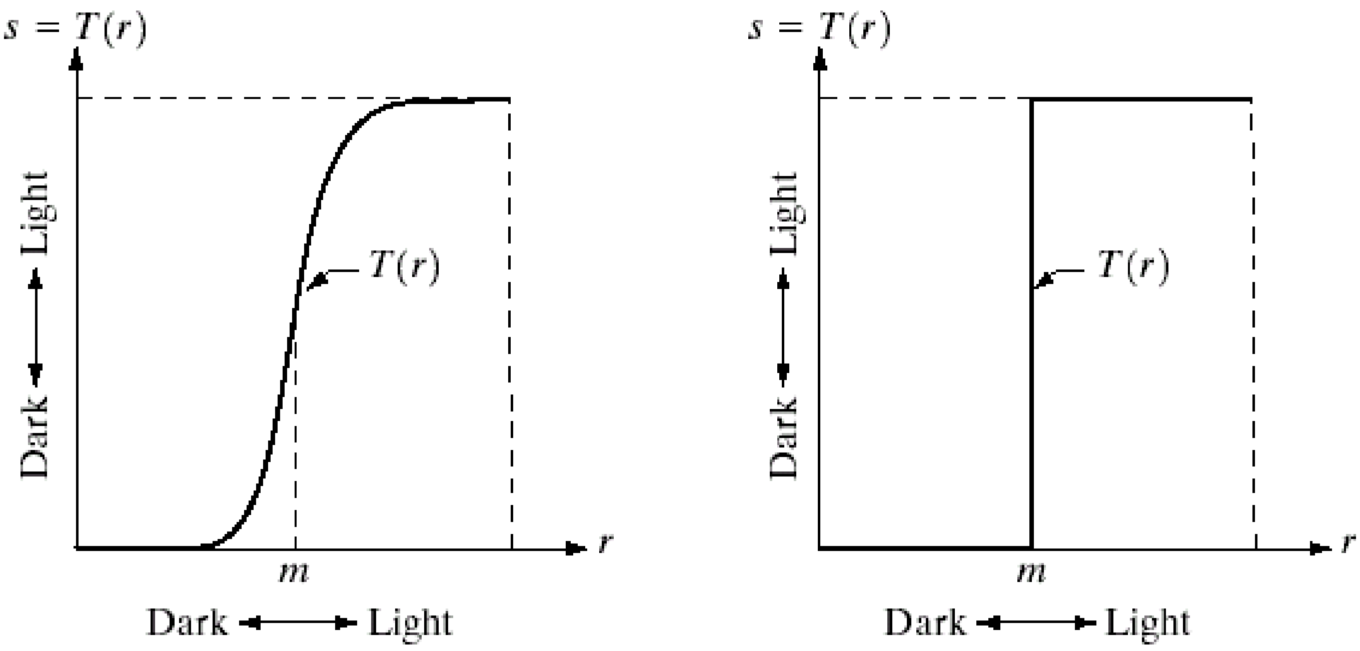
- Most spatial domain enhancement operations can be reduced to the form

$g\ (x, y) = T[\ f\ (x, y)]$

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)



Spatial Domain Image Enhancement



a b

FIGURE 3.2 Gray-level transformation functions for contrast enhancement.

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

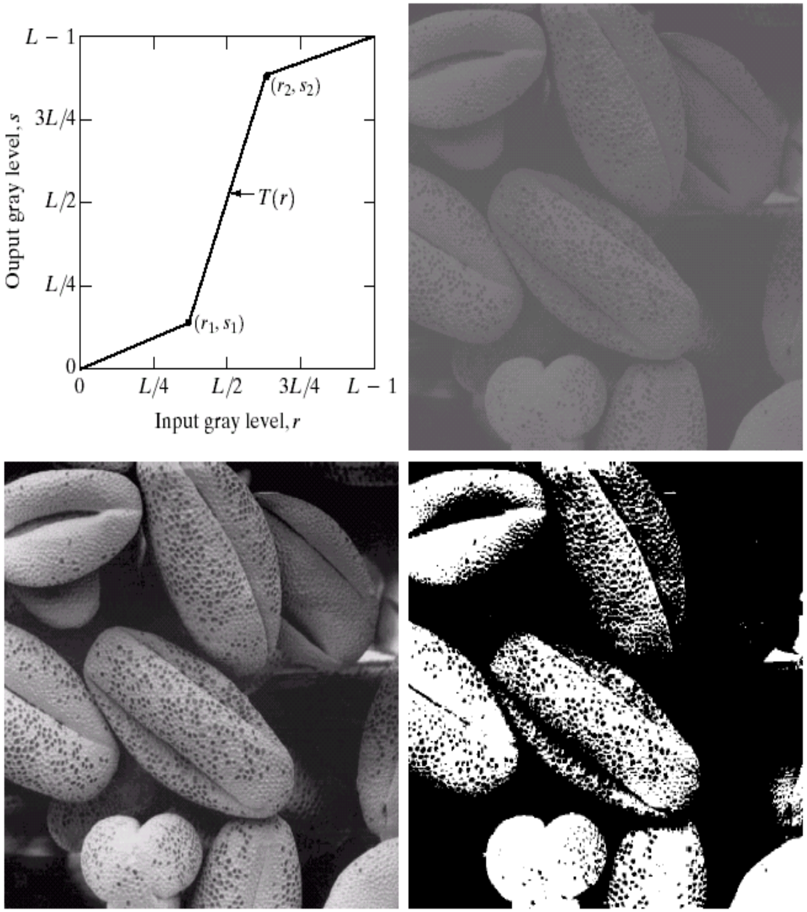
$$s = T (r)$$

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

Contrast stretching

- 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

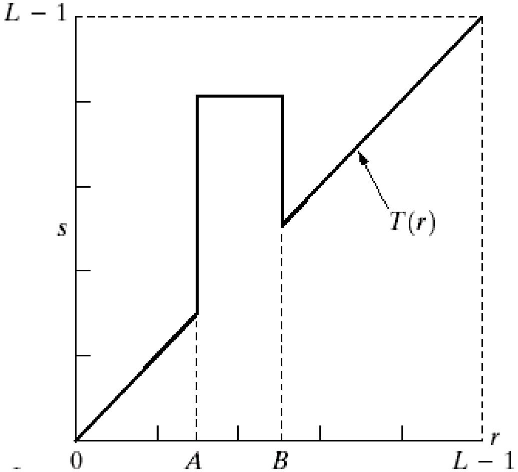


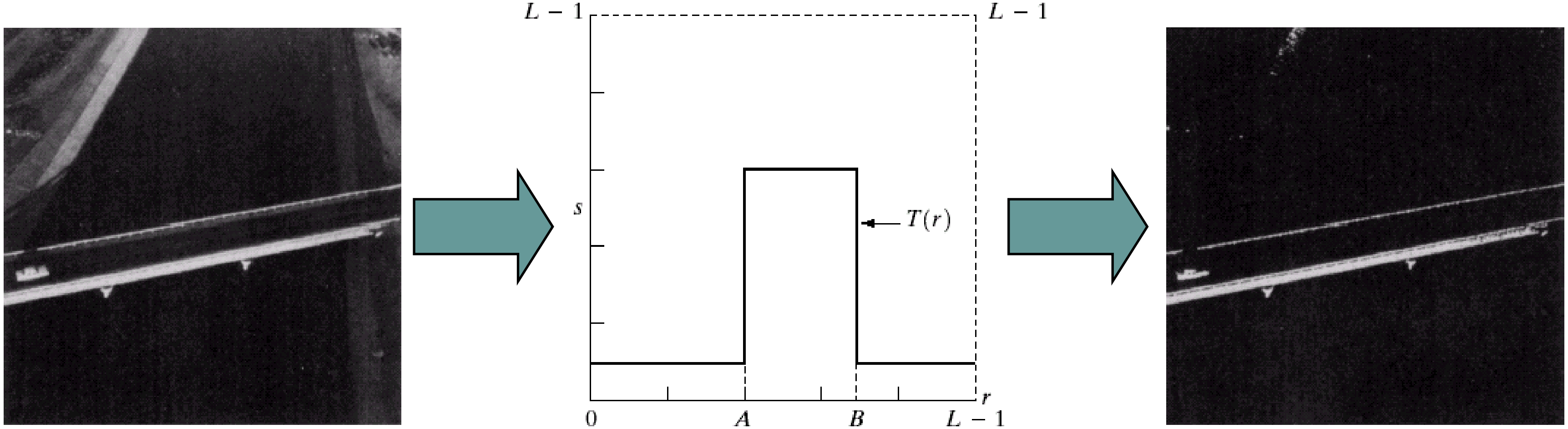
a b
c d

FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

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





Logarithmic Transformations

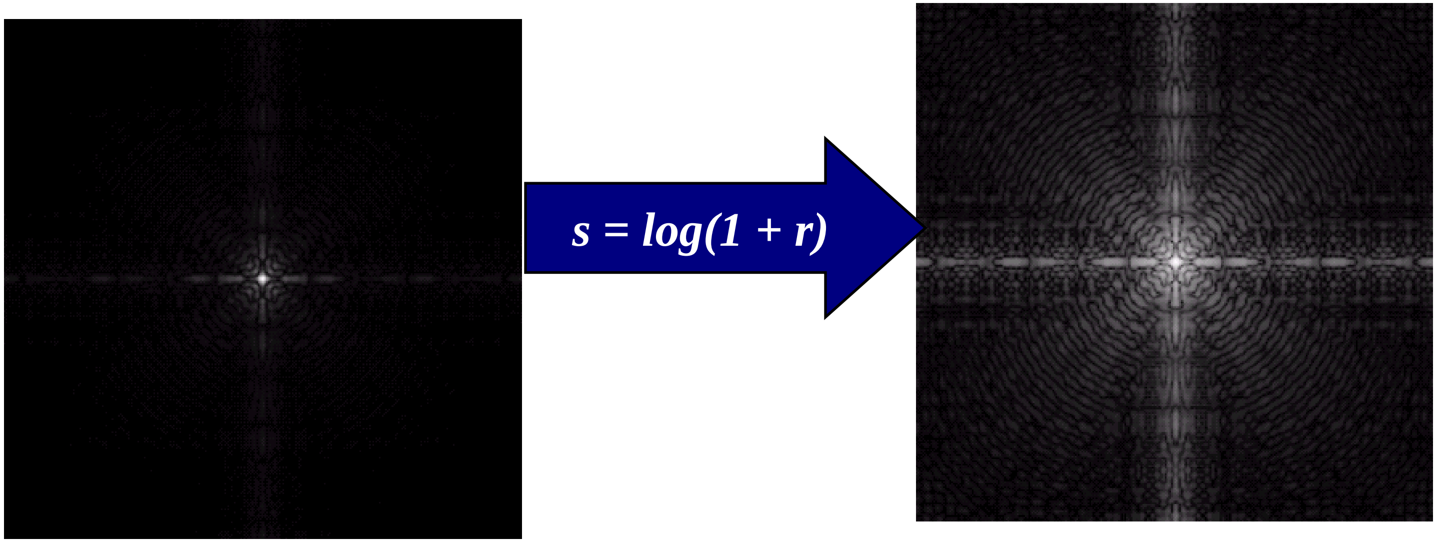
- The general form of the log transformation is

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

- 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



$s = log(1 + r)$

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

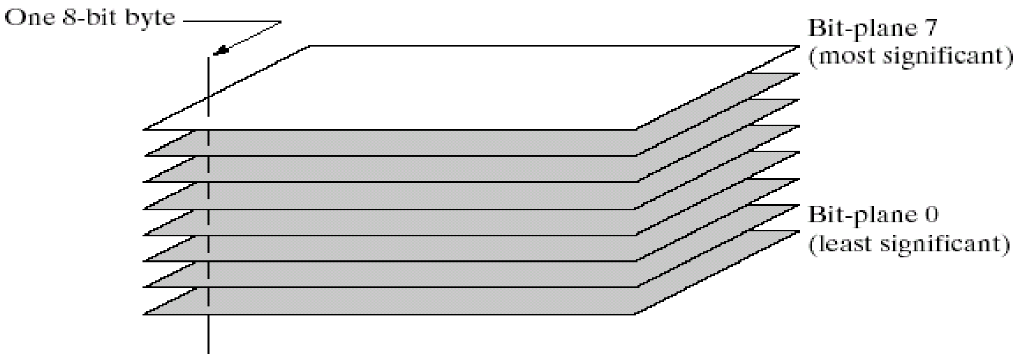

One 8-bit byte
Bit-plane 7 (most significant)
Bit-plane 0 (least significant)

FIGURE 3.12 Bit-plane representation of an 8-bit image.

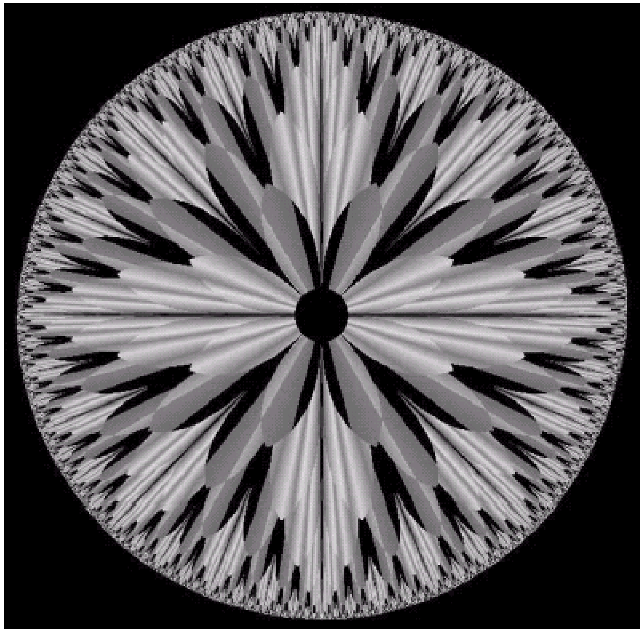
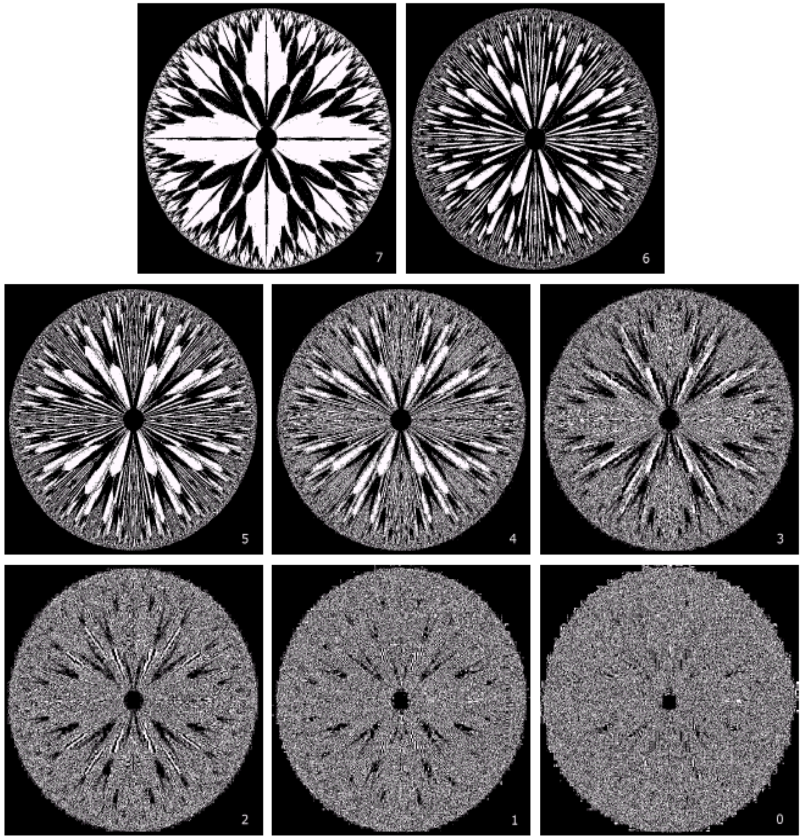


FIGURE 3.13 An 8-bit fractal image. (A fractal is an image generated from mathematical expressions). (Courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA.)

Bit Plane Slicing (cont…)

[10000000]

[01000000]



[00100000]

[00001000]

[00000100]

[00000001]

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.