Example 8.6

The following shows the time required to transmit an image of 1280 × 720 pixels using the transmission rate of 100 kbps.

a. Using a black and white image with a bit depth of 1, we need.

Transmission time = (1280 × 720 × 1) / 100,000 ≈ 9 seconds

b. Using a gray image with a bit depth of 8, we need.

Transmission time = (1280 × 720 × 8) / 100,000 ≈ 74 seconds

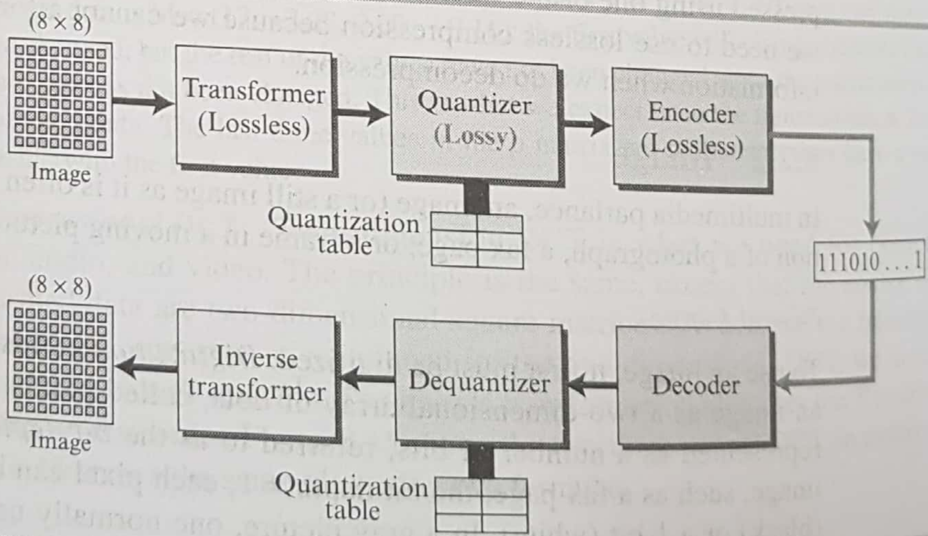
c. Using a color image with a bit depth of 24, we need.

Transmission time = (1280 × 700 × 24) / 100,000 ≈ 215 seconds

Image Compression: JPEG

Although there are both lossless and lossy compression algorithms for images, in this section we discuss the lossy compression method called *JPEG*. The **Joint Photographic Experts Group (JPEG)** standard provides lossy compression that is used in most implementations. The JPEG standard can be used for both color and gray images. However, for simplicity, we discuss only the grayscale pictures; the method can be applied to each of the three channels in a color image. In JPEG, a grayscale picture is divided into blocks of 8 × 8 pixels. The compression and decompression each goes through three steps, as shown in Figure 8.17.

Figure 8.17 *Compression in each channel of JPEG*



The purpose of dividing the picture into blocks is to decrease the number of calculations, because, as we showed in two-dimensional DCT, the number of mathematical operations for each picture is the square of the number of units.

Transformation

JPEG normally uses DCT in the first step in compression and inverse DCT in the last step in decompression. Transformation and inverse transformation are applied on 8 × 8 blocks. We discussed DCT in the previous section.

Quantization

The output of DCT transformation is a matrix of real numbers. The precise encoding of these real numbers requires a lot of bits. JPEG uses a quantization step that not only rounds real values in the matrix, but also changes some values to zeros. The zeros can be eliminated in the encoding step to achieve a high compression rate. As we discussed earlier, the result of DCT transformation defines the weights of different frequencies in the source matrix. Since high frequencies mean sudden changes in the value of pixels, they can be eliminated because human vision cannot recognize them. The quantization step creates a new matrix in which each element, C(m, n), is defined as shown below.

$$\mathbf{C}\,(m, n) = \mathbf{round}\,[\mathbf{M}(m, n)\,/\,\mathbf{Q}(m, n)]$$

in which M(m, n) is an entry in the transformed matrix and Q(m, n) is an entry in the quantization matrix. The round function first adds 0.5 to a real value and then truncates the value to an integer. This means that 3.7 is rounded to integer 4, but 3.2 is rounded to integer 3.

JPEG has defined 100 quantization matrices, Q1 to Q100, in which Q1 gives the poorest image quality but the highest level of compression and Q100 gives the best image quality but the lowest level of compression. It is up to the implementation to choose one of these matrices. Figure 8.18 shows some of these matrices.

Figure 8.18 *Three different quantization matrices*

$$\begin{bmatrix} 80 & 60 & 50 & 80 & 120 & 200 & 255 & 255 \\ 55 & 60 & 70 & 95 & 130 & 255 & 255 & 255 \\ 70 & 65 & 80 & 120 & 200 & 255 & 255 & 255 \\ 70 & 85 & 110 & 145 & 255 & 255 & 255 & 255 \\ 90 & 110 & 185 & 255 & 255 & 255 & 255 & 255 \\ 120 & 175 & 255 & 255 & 255 & 255 & 255 & 255 \\ 245 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \end{bmatrix}$$

Q10

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 110 & 103 & 99 \end{bmatrix}$$

Q50

$$\begin{bmatrix} 3 & 2 & 2 & 3 & 5 & 8 & 10 & 12 \\ 2 & 2 & 3 & 4 & 5 & 12 & 12 & 11 \\ 3 & 3 & 3 & 5 & 8 & 11 & 14 & 11 \\ 3 & 3 & 4 & 6 & 10 & 17 & 16 & 12 \\ 4 & 4 & 7 & 11 & 14 & 22 & 21 & 15 \\ 5 & 7 & 11 & 13 & 16 & 12 & 23 & 18 \\ 10 & 13 & 16 & 17 & 21 & 24 & 24 & 21 \\ 14 & 18 & 19 & 20 & 22 & 20 & 20 & 20 \end{bmatrix}$$

Q90

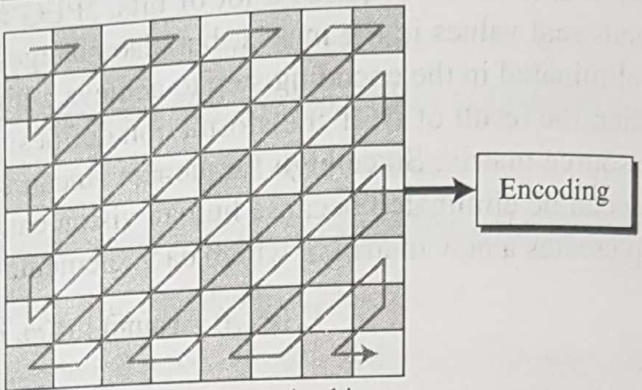
Note that the only phase in the process that is not completely reversible is the quantizing phase. We lose some information here that is not recoverable. As a matter of fact, the only reason that JPEG is called *lossy compression* is because of this quantization phase.

Encoding

After quantization, the values are reordered in a zigzag sequence before being input into the encoder. The zigzag reordering of the quantized values is done to let the values related to the lower frequency feed into the encoder before the values related to the higher frequency. Since most of the higher-frequency values are zeros, this means non-zero values are given to the encoder before the zero values. Figure 8.19 shows the process.

The encoding in this case is a lossless compression using either run-length coding or arithmetic coding.

Figure 8.19 *Reading the table*

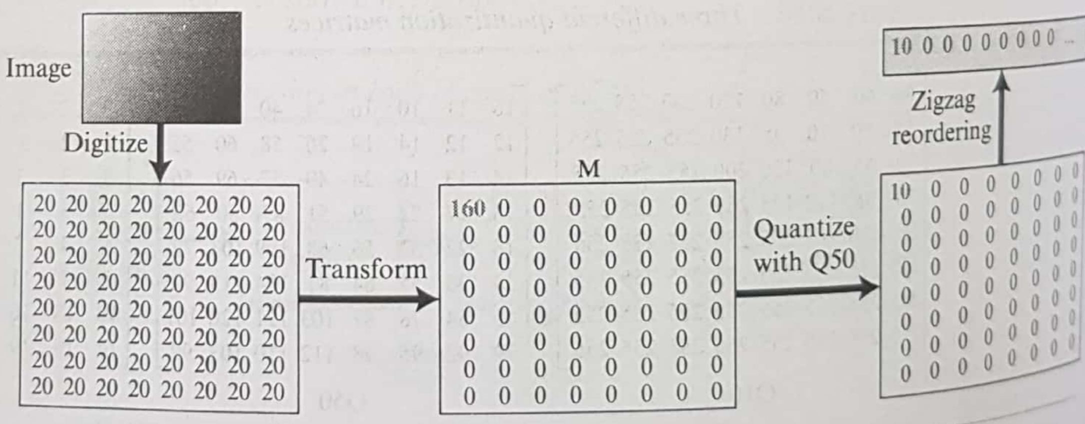


Zigzag ordering of quantized image

Example 8.7

To show the idea of JPEG compression, we use a block of gray image in which the bit depth for each pixel is 20. We have used a Java program to transform, quantize, and reorder the values in zigzag sequence; we have shown the encoding (Figure 8.20).

Figure 8.20 *Example 8.7: uniform gray scale*



Example 8.8

As the second example, we have a block that changes gradually; there is no sharp change between the values of neighboring pixels. We still get a lot of zero values, as shown in Figure 8.21.