7 Still Image Coding Standard: JPEG

In this chapter, the JPEG standard is introduced. This standard allows for lossy and lossless encoding of still images and four distinct modes of operation are supported: sequential DCT-based mode, progressive DCT-based mode, lossless mode and hierarchical mode.

7.1 INTRODUCTION

Still image coding is an important application of data compression. When an analog image or picture is digitized, each pixel is represented by a fixed number of bits, which correspond to a certain number of gray levels. In this uncompressed format, the digitized image requires a large number of bits to be stored or transmitted. As a result, compression become necessary due to the limited communication bandwidth or storage size. Since the mid-1980s, the ITU and ISO have been working together to develop a joint international standard for the compression of still images. Officially, JPEG [jpeg] is the ISO/IEC international standard 10918-1; digital compression and coding of continuous-tone still images, or the ITU-T Recommendation T.81. JPEG became an international standard in 1992. The JPEG standard allows for both lossy and lossless encoding of still images. The algorithm for lossy coding is a DCT-based coding scheme. This is the baseline of JPEG and is sufficient for many applications. However, to meet the needs of applications that cannot tolerate loss, e.g., compression of medical images, a lossless coding scheme is also provided and is based on a predictive coding scheme. From the algorithmic point of view, JPEG includes four distinct modes of operation, namely, sequential DCT-based mode, progressive DCT-based mode, lossless mode, and hierarchical mode. In the following sections, an overview of these modes is provided. Further technical details can be found in the books by Pennelbaker and Mitchell (1992) and Symes (1998).

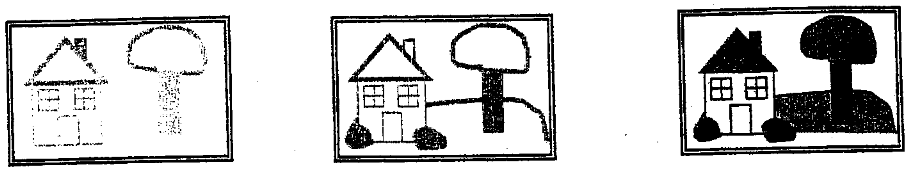
In the sequential DCT-based mode, an image is first partitioned into blocks of 8×8 pixels. The blocks are processed from left to right and top to bottom. The 8×8 two-dimensional Forward DCT is applied to each block and the 8×8 DCT coefficients are quantized. Finally, the quantized DCT coefficients are entropy encoded and output as part of the compressed image data.

In the progressive DCT-based mode, the process of block partitioning and Forward DCT transform is the same as in the sequential DCT-based mode. However, in the progressive mode, the quantized DCT coefficients are first stored in a buffer before the encoding is performed. The DCT coefficients in the buffer are then encoded by a multiple scanning process. In each scan, the quantized DCT coefficients are partially encoded by either spectral selection or successive approximation. In the method of spectral selection, the quantized DCT coefficients are divided into multiple spectral bands according to a zigzag order. In each scan, a specified band is encoded. In the method of successive approximation, a specified number of most significant bits of the quantized coefficients are first encoded and the least significant bits are then encoded in subsequent scans.

The difference between sequential coding and progressive coding is shown in Figure 7.1. In the sequential coding an image is encoded part by part according to the scanning order, while in the progressive coding the image is encoded by a multiscanning process and in each scan the full image is encoded to a certain quality level.

As mentioned earlier, lossless coding is achieved by a predictive coding scheme. In this scheme, three neighboring pixels are used to predict the current pixel to be coded. The prediction difference

(a) Sequential coding



(b) Progressive coding

FIGURE 7.1 (a) Sequential coding, (b) progressive coding.

is entropy coded using either Huffman or arithmetic coding. Since the prediction is not quantized, the coding is lossless.

Finally, in the hierarchical mode, an image is first spatially down-sampled to a multilayered pyramid, resulting in a sequence of frames as shown in Figure 7.2. This sequence of frames is encoded by a predictive coding scheme. Except for the first frame, the predictive coding process is applied to the differential frames, i.e., the differences between the frame to be coded and the predictive reference frame. It is important to note that the reference frame is equivalent to the previous frame that would be reconstructed in the decoder. The coding method for the difference frame may use the DCT-based coding method, the lossless coding method, or the DCT-based processes with a final lossless process. Down-sampling and up-sampling filters are used in the hierarchical mode. The hierarchical coding mode provides a progressive presentation similar to the progressive DCT-based mode, but is also useful in the applications that have multiresolution requirements. The hierarchical coding mode also provides the capability of progressive coding to a final lossless stage.

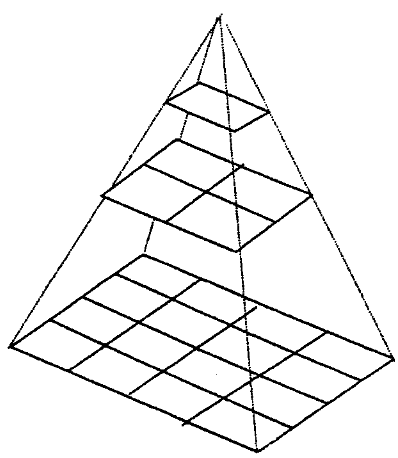


FIGURE 7.2 Hierarchical multiresolution encoding.

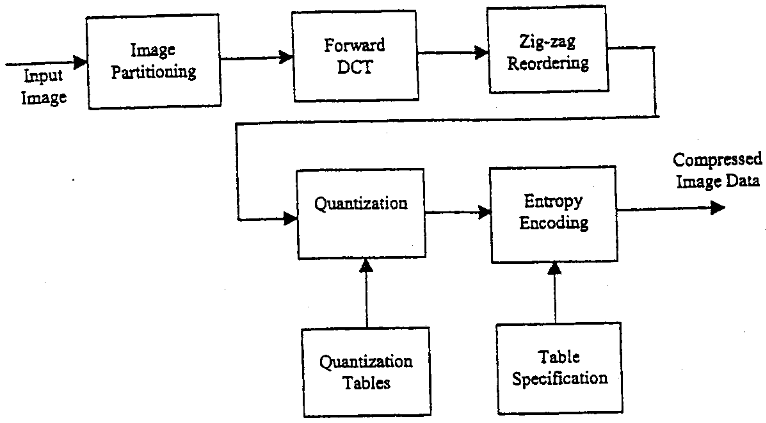


FIGURE 7.3 Block diagram of a sequential DCT-based encoding process.

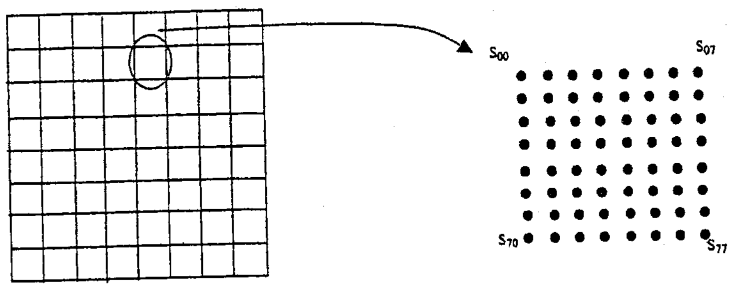


FIGURE 7.4 Partitioning to 8 × 8 blocks.

7.2 SEQUENTIAL DCT-BASED ENCODING ALGORITHM

The sequential DCT-based coding algorithm is the baseline algorithm of the JPEG coding standard. A block diagram of the encoding process is shown in Figure 7.3. As shown in Figure 7.4, the digitized image data are first partitioned into blocks of 8 × 8 pixels. The two-dimensional forward DCT is applied to each 8 × 8 block. The two-dimensional forward and inverse DCT of 8 × 8 block are defined as follows:

$$\text{FDCT:} \quad S_{uv} = \frac{1}{4} C_u C_v \sum_{i=0}^{7} \sum_{j=0}^{7} s_{ij} \cos\frac{(2i+1)u\pi}{16} \cos\frac{(2j+1)v\pi}{16}$$

$$\text{IDCT:} \quad s_{ij} = \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} C_u C_v S_{uv} \cos\frac{(2i+1)u\pi}{16} \cos\frac{(2j+1)v\pi}{16} \tag{7.1}$$

$$C_u C_v = \begin{cases} \frac{1}{\sqrt{2}} & \textit{for } u, v = 0 \\ 1 & \textit{otherwise} \end{cases}$$

where s_{ij} is the value of the pixel at position (i,j) in the block, and S_{uv} is the transformed (u,v) DCT coefficient.

TABLE 7.1
Two Examples of Quantization Tables Used by JPEG

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	100	103	99

Luminance quantization table

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Chrominance quantization table

After the forward DCT, quantization of the transformed DCT coefficients is performed. Each of the 64 DCT coefficients is quantized by a uniform quantizer:

$$S_{quv} = round\left(\frac{S_{uv}}{Q_{uv}}\right) \tag{7.2}$$

where the S_{quv} is the quantized value of the DCT coefficient, S_{uv}, and Q_{uv} is the quantization step obtained from the quantization table. There are four quantization tables that may be used by the encoder, but there is no default quantization table specified by the standard. Two particular quantization tables are shown in Table 7.1.

At the decoder, the dequantization is performed as follows:

$$R_{quv} = S_{quv} \times Q_{uv} \tag{7.3}$$

where R_{quv} is the value of the dequantized DCT coefficient. After quantization, the DC coefficient, S_{q00}, is treated separately from the other 63 AC coefficients. The DC coefficients are encoded by a predictive coding scheme. The encoded value is the difference (*DIFF*) between the quantized DC coefficient of the current block (S_{q00}) and that of the previous block of the same component (*PRED*):

$$DIFF = S_{q00} - PRED \tag{7.4}$$

The value of *DIFF* is entropy coded with Huffman tables. More specifically, the two's complement of the possible *DIFF* magnitudes are grouped into 12 categories, "SSSS". The Huffman codes for these 12 difference categories and additional bits are shown in the Table 7.2.

For each nonzero category, additional bits are added to the codeword to uniquely identify which difference within the category actually occurred. The number of additional bits is defined by "SSSS" and the additional bits are appended to the least significant bit of the Huffman code (most significant bit first) according to the following rule. If the difference value is positive, the "SSSS" low-order bits of *DIFF* are appended; if the difference value is negative, then the "SSSS" low-order bits of *DIFF-1* are appended. As an example, the Huffman tables used for coding the luminance and chrominance DC coefficients are shown in Tables 7.3 and 7.4, respectively. These two tables have been developed from the average statistics of a large set of images with 8-bit precision.

TABLE 7.2
Huffman Coding of DC Coefficients

SSSS	DIFF Values	Additional Bits
0	0	–
1	–1,1	0,1
2	–3,–2,2,3	00,01,10,11
3	–7,...,–4,4,...,7	000,...,011,100,.,111
4	–15,...,–8,8,...,15	0000,..0111,1000,...,1111
5	–31,...,–16,16,...,31	00000,...,01111,10000,...,11111
6	–63,...–32,32,...63	...,...
7	–127,...,–64,64,...,127	...,...
8	–255,...,–128,128,...,255	...,...
9	–511,...,–256,256,...,511	...,...
10	–1023,...,–512,512,...,1023	...,...
11	–2047,...,–1024,1024,...,2047	...,...

TABLE 7.3
Huffman Table for Luminance DC Coefficient Differences

Category	Code Length	Codeword
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

In contrast to the coding of DC coefficients, the quantized AC coefficients are arranged to a zigzag order before being entropy coded. This scan order is shown in Figure 7.5.

According to the zigzag scanning order, the quantized coefficients can be represented as:

$$ZZ(0) = S_{q00},\ ZZ(1) = S_{q01},\ ZZ(2) = S_{q10},\ \ldots,\ ZZ(63) = S_{q77}. \tag{7.5}$$

Since many of the quantized AC coefficients become zero, they can be very efficiently encoded by exploiting the run of zeros. The run-length of zeros are identified by the nonzero coefficients. An 8-bit code 'RRRRSSSS' is used to represent the nonzero coefficient. The four least significant bits, 'SSSS', define a category for the value of the next nonzero coefficient in the zigzag sequence, which ends the zero run. The four most significant bits, 'RRRR', define the run-length of zeros in the zigzag sequence or the position of the nonzero coefficient in the zigzag sequence. The composite value, RRRRSSSS, is shown in Figure 7.6. The value 'RRRRSSSS' = '11110000' is defined as ZRL, "RRRR" = "1111" represents a run-length of 16 zeros and "SSSS" = "0000" represents a zero amplitude. Therefore, ZRL is used to represent a run-length of 16 zero coefficients followed

TABLE 7.4
Huffman table for chrominance DC coefficient differences

Category	Code Length	Codeword
0	2	00
1	2	01
2	2	10
3	3	110
4	4	1110
5	5	11110
6	6	111110
7	7	1111110
8	8	11111110
9	9	111111110
10	10	1111111110
11	11	11111111110

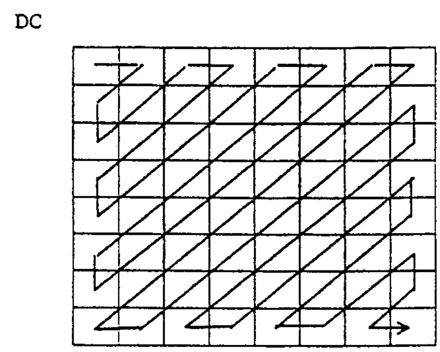


FIGURE 7.5 Zigzag scanning order of DCT coefficients.

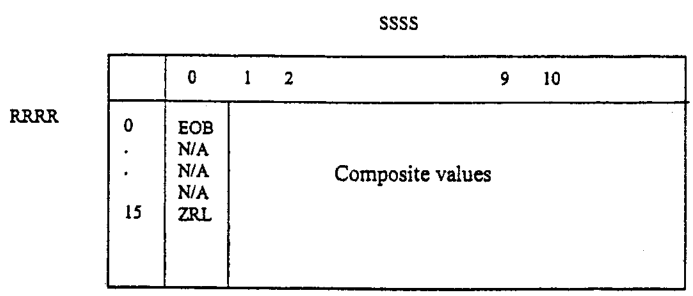


FIGURE 7.6 Two-dimensional value array for Huffman coding.

by a zero-amplitude coefficient, it is not an *abbreviation*. In the case of a run-length of zero coefficients that exceeds 15, multiple symbols will be used. A special value 'RRRRSSSS' = '00000000' is used to code the end-of-block (EOB). An EOB occurs when the remaining coefficients in the block are zeros. The entries marked "N/A" are undefined.

TABLE 7.5
Huffman Coding for AC Coefficients

Category (SSSS)	AC Coefficient Range
1	–1,1
2	–3,–2,2,3
3	–7,...,–4,4,...,7
4	–15,...,–8,8,...,15
5	–31,...,–16,16,...,31
6	–63,...,–32,32,...,63
7	–127,...,–64,.64,...,127
8	–255,...,–128,128,...,255
9	–511,...,–256,256,...,511
10	–1023,..,–512,512,...,1023
11	–2047,...,–1024,1024,...,2047

The composite value, RRRRSSSS, is then Huffman coded. SSSS is actually the number to indicate "category" in the Huffman code table. The coefficient values for each category are shown in Table 7.5.

Each Huffman code is followed by additional bits that specify the sign and exact amplitude of the coefficients. As with the DC code tables, the AC code tables have also been developed from the average statistics of a large set of images with 8-bit precision. Each composite value is represented by a Huffman code in the AC code table. The format for the additional bits is the same as in the coding of DC coefficients. The value of SSSS gives the number of additional bits required to specify the sign and precise amplitude of the coefficient. The additional bits are either the low-order SSSS bits of ZZ(k) when ZZ(k) is positive, or the low-order SSSS bits of ZZ(k)-1 when ZZ(k) is negative. Here, ZZ(k) is the *kth* coefficient in the zigzag scanning order of coefficients being coded. The Huffman tables for AC coefficients can be found in Annex K of the JPEG standard (jpeg) and are not listed here due to space limitations.

As described above, Huffman coding is used as the means of entropy coding. However, an adaptive arithmetic coding procedure can also be used. As with the Huffman coding technique, the binary arithmetic coding technique is also lossless. It is possible to transcode between two systems without either of the FDCT or IDCT processes. Since this transcoding is a lossless process, it does not affect the picture quality of the reconstructed image. The arithmetic encoder encodes a series of binary symbols, zeros or ones, where each symbol represents the possible result of a binary decision. The binary decisions include the choice between positive and negative signs, a magnitude being zero or nonzero, or a particular bit in a sequence of binary digits being zero or one. There are four steps in the arithmetic coding: initializing the statistical area, initializing the encoder, terminating the code string, and adding restart markers.

7.3 PROGRESSIVE DCT-BASED ENCODING ALGORITHM

In progressive DCT-based coding, the input image is first partitioned to blocks of 8×8 pixels. The two-dimensional 8×8 DCT is then applied to each block. The transformed DCT-coefficient data are then encoded with multiple scans. At each scan, a portion of the transformed DCT coefficient data is encoded. This partially encoded data can be reconstructed to obtain a full image size with lower picture quality. The coded data of each additional scan will enhance the reconstructed image quality until the full quality has been achieved at the completion of all scans. Two methods have been used in the JPEG standard to perform the DCT-based progressive coding. These include spectral selection and successive approximation.