Jpeg

JPEG is an image compression standard which was accepted as an international standard in 1992.

Developed by the Joint Photographic Expert Group of the ISO/IEC for coding and compression of color/gray scale images.

Yields acceptable compression in the 10:1 range.

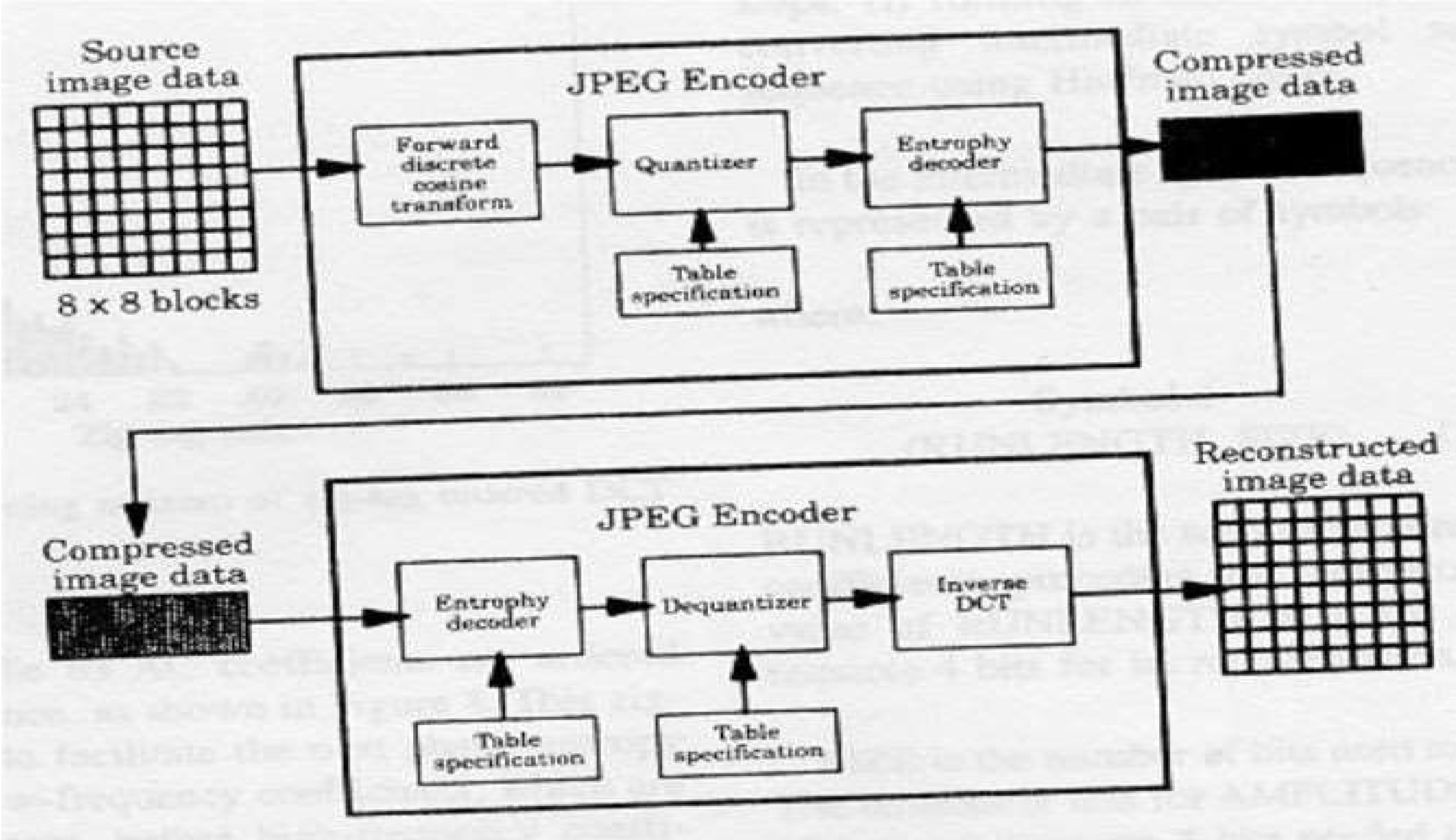
A scheme for video compression based on JPEG called Motion JPEG (MJPEG) exists

JPEG

Lossy Compression Technique based on use of Discrete Cosine Transform (DCT)

STEPS IN JPEG COMPRESSION:

1. Divide Each plane into 8x8 size blocks.
2. Compute DCT of each block
3. Treat separately DC components of each block.
4. Apply Quantization to discard values
5. Encode DC components and transmit data.


Source image data
8 x 8 blocks
JPEG Encoder
Forward discrete cosine transform
Quantizer
Entrophy decoder
Table specification
Table specification
Compressed image data
Compressed image data
JPEG Encoder
Entrophy decoder
Dequantizer
Inverse DCT
Table specification
Table specification
Reconstructed image data

Steps in JPEG

1. Divide the image into 8x8 subimages;

For each subimage do:

2.Shift the gray-levels in the range [-128, 127] - DCT requires range be centered around 0

3. Apply DCT (i.e., 64 coefficients)

    1 DC coefficient: F(0,0)

    63 AC coefficients: F(u,v)

Image Compression Standards

JPEG Encoding - Example

Sub-Image →

52	55	61	66	70	61	64	73
63	59	66	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

-128 →

−76	−73	−67	−62	−58	−67	−64	−55
−65	−69	−62	−38	−19	−43	−59	−56
−66	−69	−60	−15	16	−24	−62	−55
−65	−70	−57	−6	26	−22	−58	−59
−61	−67	−60	−24	−2	−40	−60	−58
−49	−63	−68	−58	−51	−65	−70	−53
−43	−57	−64	−69	−73	−67	−63	−45
−41	−49	−59	−60	−63	−52	−50	−34

DCT

−415	−29	−62	25	55	−20	−1	3
7	−21	−62	9	11	−7	−6	6
−46	8	77	−25	−30	10	7	−5
−50	13	35	−15	−9	6	0	3
11	−8	−13	−2	−1	1	−4	1
−10	1	3	−3	−1	0	2	−1
−4	−1	2	−1	2	−3	1	−2
−1	−1	−1	−2	−1	−1	0	−1

F(0,0) = (1/8) * addition of matrix = (1/8) *

F(0,0) = (1/8) * (-3317) = -414.62 =~ -415

Z represents quality

Z

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

T

−415	−29	−62	25	55	−20	−1	3
7	−21	−62	9	11	−7	−6	6
−46	8	77	−25	−30	10	7	−5
−50	13	35	−15	−9	6	0	3
11	−8	−13	−2	−1	1	−4	1
−10	1	3	−3	−1	0	2	−1
−4	−1	2	−1	2	−3	1	−2
−1	−1	−1	−2	−1	−1	0	−1

T/Z

−26	−3	−6	2	2	0	0	0
1	−2	−4	0	0	0	0	0
−3	1	5	−1	−1	0	0	0
−4	1	2	−1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$$
\begin{matrix}
-26 & -3 & -6 & 2 & 2 & 0 & 0 & 0 \\
1 & -2 & -4 & 0 & 0 & 0 & 0 & 0 \\
-3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\
-4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
$$

Zigzag ordering → [-26 -3 1 -3 -2 -6 2 -4 1 -4 1 1 5 0 2 0 0 -1 2 0 0 0 0 0 -1 -1 EOB]

If DC coefficient of the transformed and quantized sub-image to its immediate left was -17, the resulting DPCM:

[-26-(-17)]=-9 Category: 4 Length: 7

101 0110

(0111 -1)

TABLE 8.17
JPEG coefficient coding categories.

Range	DC Difference Category	AC Category
0	0	N/A
$-1, 1$	1	1
$-3, -2, 2, 3$	2	2
$-7, \ldots, -4, 4, \ldots, 7$	3	3
$-15, \ldots, -8, 8, \ldots, 15$	4	4
$-31, \ldots, -16, 16, \ldots, 31$	5	5
$-63, \ldots, -32, 32, \ldots, 63$	6	6
$-127, \ldots, -64, 64, \ldots, 127$	7	7
$-255, \ldots, -128, 128, \ldots, 255$	8	8
$-511, \ldots, -256, 256, \ldots, 511$	9	9
$-1023, \ldots, -512, 512, \ldots, 1023$	A	A
$-2047, \ldots, -1024, 1024, \ldots, 2047$	B	B
$-4095, \ldots, -2048, 2048, \ldots, 4095$	C	C
$-8191, \ldots, -4096, 4096, \ldots, 8191$	D	D
$-16383, \ldots, -8192, 8192, \ldots, 16383$	E	E
$-32767, \ldots, -16384, 16384, \ldots, 32767$	F	N/A

TABLE 8.18
JPEG default DC code (luminance).

Category	Base Code	Length	Category	Base Code	Length
0	010	3	6	1110	10
1	011	4	7	11110	12
2	100	5	8	111110	14
3	00	5	9	1111110	16
4	101	7	A	11111110	18
5	110	8	B	111111110	20

Run/ Category	Base Code	Length	Run/ Category	Base Code	Length
0/0	**1010 (= EOB)**	**4**			
0/1	00	3	8/1	11111010	9
0/2	01	4	8/2	111111111000000	17
0/3	100	6	8/3	1111111110110111	19
0/4	1011	8	8/4	1111111110111000	20
0/5	11010	10	8/5	1111111110111001	21
0/6	111000	12	8/6	1111111110111010	22
0/7	1111000	14	8/7	1111111110111011	23
0/8	1111110110	18	8/8	1111111110111100	24
0/9	1111111110000010	25	8/9	1111111110111101	25
0/A	1111111110000011	26	8/A	1111111110111110	26
1/1	1100	5	9/1	111111000	10
1/2	111001	8	9/2	1111111110111111	18
1/3	1111001	10	9/3	1111111111000000	19
1/4	111110110	13	9/4	1111111111000001	20
1/5	11111110110	16	9/5	1111111111000010	21
1/6	1111111110000100	22	9/6	1111111111000011	22
1/7	1111111110000101	23	9/7	1111111111000100	23
1/8	1111111110000110	24	9/8	1111111111000101	24
1/9	1111111110000111	25	9/9	1111111111000110	25
1/A	1111111110001000	26	9/A	1111111111000111	26
2/1	11011	6	A/1	111111001	10
2/2	11111000	10	A/2	1111111111001000	18
2/3	1111110111	13	A/3	1111111111001001	19
2/4	1111111110001001	20	A/4	1111111111001010	20
2/5	1111111110001010	21	A/5	1111111111001011	21
2/6	1111111110001011	22	A/6	1111111111001100	22
2/7	1111111110001100	23	A/7	1111111111001101	23

TABLE 8.19
JPEG default AC code (luminance) *(continues on next page)*.

Table 8.19 (Con't)

2/8	1111111110001101	24	A/8	1111111111001110	24
2/9	1111111110001110	25	A/9	1111111111001111	25
2/A	1111111110001111	26	A/A	1111111111010000	26
3/1	111010	7	B/1	111111010	10
3/2	111110111	11	B/2	1111111111010001	18
3/3	11111110111	14	B/3	1111111111010010	19
3/4	1111111110010000	20	B/4	1111111111010011	20
3/5	1111111110010001	21	B/5	1111111111010100	21
3/6	1111111110010010	22	B/6	1111111111010101	22
3/7	1111111110010011	23	B/7	1111111111010110	23
3/8	1111111110010100	24	B/8	1111111111010111	24
3/9	1111111110010101	25	B/9	1111111111011000	25
3/A	1111111110010110	26	B/A	1111111111011001	26
4/1	111011	7	C/1	1111111010	11
4/2	1111111000	12	C/2	1111111111011010	18
4/3	1111111110010111	19	C/3	1111111111011011	19
4/4	1111111110011000	20	C/4	1111111111011100	20
4/5	1111111110011001	21	C/5	1111111111011101	21
4/6	1111111110011010	22	C/6	1111111111011110	22
4/7	1111111110011011	23	C/7	1111111111011111	23
4/8	1111111110011100	24	C/8	1111111111100000	24
4/9	1111111110011101	25	C/9	1111111111100001	25
4/A	1111111110011110	26	C/A	1111111111100010	26

[-26 -3 1 -3 -2 -6 2 -4 1 -4 1 1 5 0 2 0 0 -1 2 0 0 0 0 0 -1 -1 EOB]

1010110 0100 001 0100 0101 100001 0110 100011 001 100011 001 001 100101
11100110 110110 0110 11110100 000 1010

−26	−3	−6	2	2	0	0	0
1	−2	−4	0	0	0	0	0
−3	1	5	−1	−1	0	0	0
−4	1	2	−1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

T * Z

−416	−33	−60	32	48	0	0	0
12	−24	−56	0	0	0	0	0
−42	13	80	−24	−40	0	0	0
−56	17	44	−29	0	0	0	0
18	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

DCT^{-1}

−70	−64	−61	−64	−69	−66	−58	−50
−72	−73	−61	−39	−30	−40	−54	−59
-68	−78	−58	−9	13	−12	−48	−64
-59	−77	−57	0	22	−13	−51	−60
−54	−75	−64	−23	−13	−44	−63	−56
−52	−71	−72	−54	−54	−71	−71	−54
−45	−58	−70	−68	−67	−67	−61	−50
−35	−47	−61	−66	−60	−48	−44	−44

+128

58	64	67	64	59	62	70	78*n*
56	55	67	89	98	88	74	69
60	50	70	119	141	116	80	64
69	51	71	128	149	115	77	68
74	53	64	105	115	84	65	72
76	57	56	74	75	57	57	74
83	69	59	60	61	61	67	78
93	81	67	62	69	80	84	84

Error

−6	−9	−6	2	11	−1	−6	−5
7	4	−1	1	11	−3	−5	3
2	9	−2	−6	−3	−12	−14	9
−6	7	0	−4	−5	−9	−7	1
−7	8	4	−1	11	4	3	−2
3	8	4	−4	2	11	1	1
2	2	5	−1	−6	0	−2	5
−6	−2	2	6	−4	−4	−6	10

The BMP file format uses a form of run-length encoding in which image data is represented in two different modes: encoded and absolute—and either mode can occur anywhere in the image. In *encoded* mode, a two byte RLE representation is used. The first byte specifies the number of consecutive pixels that have the color index contained in the second byte. The 8-bit color index selects the run's intensity (color or gray value) from a table of 256 possible intensities.

In *absolute* mode, the first byte is 0 and the second byte signals one of four possible conditions, as shown in Table 8.8. When the second byte is 0 or 1, the end of a line or the end of the image has been reached. If it is 2, the next two bytes contain unsigned horizontal and vertical offsets to a new spatial position (and pixel) in the image. If the second byte is between 3 and 255, it specifies the number of uncompressed pixels that follow—with each subsequent byte containing the color index of one pixel. The total number of bytes must be aligned on a 16-bit word boundary.

Second Byte Value	Condition
0	End of line
1	End of image
2	Move to a new position
3–255	Specify pixels individually

Decode the BMP encoded sequence $\{3, 4, 5, 6, 0, 3, 103, 125, 67, 0, 2, 47\}$.

Using the BMP specification given in Example 8.8 of Section 8.2.5, the first two bytes indicate that the uncompressed data begins with a run of 4s with length 3. In a similar manner, the second two bytes call for a run of 6s with length 5. The first four bytes of the BMP encoded sequence are encoded mode. Because the 5th byte is 0 and the 6th byte is 3, absolute mode is entered and the next three values are taken as uncompressed data. Because the total number of bytes in absolute mode must be aligned on a 16-bit word boundary, the 0 in the 10th byte of the encoded sequence is padding and should be ignored. The final two bytes specify an encoded mode run of 47s with length 2. Thus, the complete uncompressed sequence is {4, 4, 4, 6, 6, 6, 6, 6, 103, 125, 67, 47, 47}.

Predictive coding

The approach,commonly referred to as predictive coding,is based on eliminating the redundancies of closely spaced pixels—in space and/or time— by extracting and coding only the new information in each pixel.The new information of a pixel is defined as the difference between the actual and predicted value of the pixel

1. Lossless predictive coding

The system consists of an encoder and a decoder, each containing an identical *predictor*. As successive samples of discrete time input signal, $f(n)$, are introduced to the encoder, the predictor generates the anticipated value of each sample based on a specified number of past samples. The output of the predictor is then rounded to the nearest integer, denoted $\hat{f}(n)$, and used to form the difference or *prediction error*

$$e(n) = f(n) - \hat{f}(n)$$

The decoder reconstructs $e(n)$ from the received variable-length code words and performs the inverse operation

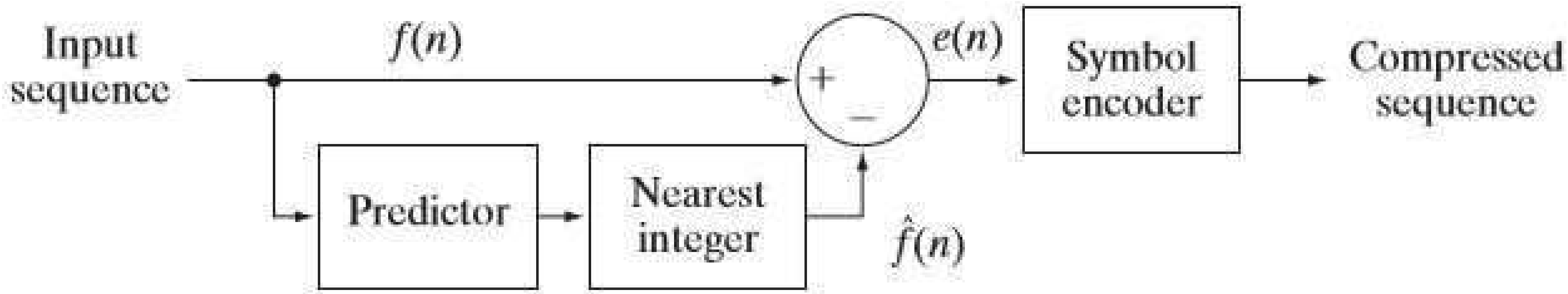
$$f(n) = e(n) + \hat{f}(n)$$

to decompress or recreate the original input sequence.

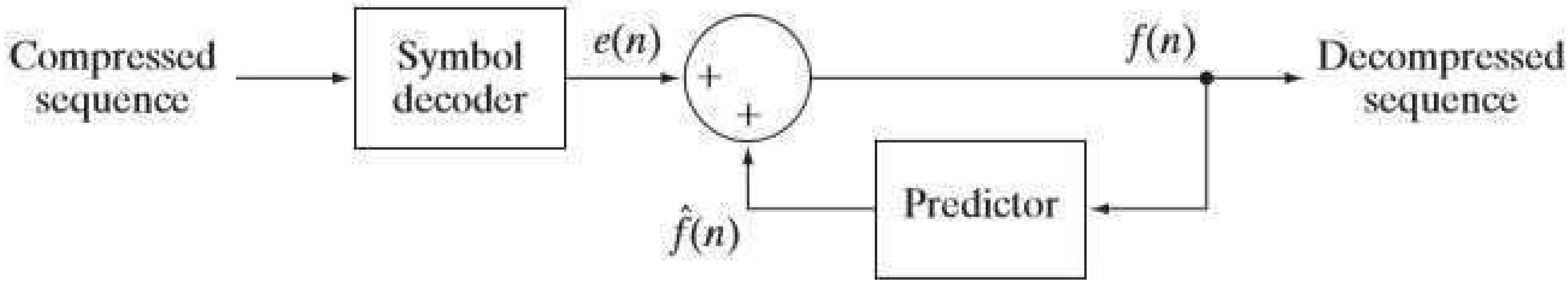
Various local, global, and adaptive methods (see the later subsection entitled Lossy predictive coding) can be used to generate $\hat{f}(n)$. In many cases, the prediction is formed as a linear combination of m previous samples. That is,

$$\hat{f}(n) = \text{round}\left[\sum_{i=1}^{m} \alpha_i f(n-i)\right]$$

where m is the *order* of the linear predictor, round is a function used to denote the rounding or nearest integer operation, and the α_i for $i = 1, 2, \ldots, m$ are prediction coefficients



Input sequence
$f(n)$
$e(n)$
Symbol encoder
Compressed sequence
Predictor
Nearest integer
$\hat{f}(n)$



Compressed sequence
Symbol decoder
$e(n)$
$f(n)$
Decompressed sequence
Predictor
$\hat{f}(n)$

the m samples used to predict the value of each pixel come from the current scan line (called 1-D linear predictive coding), from the current and previous scan lines (called 2-D linear predictive coding), or from the current image and previous images in a sequence of images (called 3-D linear predictive coding). Thus, for 1-D linear predictive image coding, Eq. (8.2-32) can be written as

$$\hat{f}(x, y) = \text{round}\left[\sum_{i=1}^{m} \alpha_i f(x, y - i)\right]$$