J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*
Copyright © 1999 John Wiley & Sons, Inc.

IMAGE CODES

Although usually acquired from analog optical devices, images are often sampled into the digital form, because they are more easily stored, transmitted, and processed digitally. The major difficulty with digital image storage and transmission, however, is the size of bits required to record the image data. For a 512×512 gray-scale image with eight-bit resolution, 256 kbytes of storage space are required. With color images or digital video data, the amount of data is enormously greater. For example, the bit rate is 8.70 Mbytes/s for a color video sequence with 352×288 pixels per picture, eight bits per color channel, and 30 pictures/s. For a 30 s video clip at such a bit rate, the total data takes up 261 Mbytes of storage space, or 21.12 h of transmission time with a 28800-baud modem. Therefore it is desirable to use data compression techniques to reduce the amount of data for digital images and video.

There are some important differences between digital image/video compressions and other digital data compression. First, for most other data compression applications, it is desirable to have the data themselves unaltered. In digital image compression, on the other hand, some information loss is allowed as long as the visual appearance of the image or video is not severely impaired. In some cases, though, lossless image compression is required. For example, it may be preferred that medical images be losslessly compressed, because small deviations from the original image may affect the doctor's diagnosis. The second difference is that natural images contain much redundant information that is very useful for compression. The background of a natural image, for instance, contains a lot of pixels with similar luminance or color components. These background pixels are represented more efficiently by using various image compression techniques.

Generally speaking, digital image/video compression techniques are classified into two categories: lossless compression and lossy compression. Lossless image/video compression uses many lossless compression techniques mentioned in Data compression, lossy. Lossy image/video compression is more important in image/video compression because the compression ratio is more flexibly adjusted without having to preserve every detail in the image/video. This section primarily focuses on this category, and so do many international standards to be introduced later in this section.

General Lossy Image/Video Compression Framework

The most important issue in image/video compression is reducing the redundancy in the image/video. Most of state-of-the-art lossy image/video compression techniques use transform coding for this purpose. A general image/video compression framework using transform coding includes four major parts: color space conversion, transform, quantization, and entropy coding, as shown in Fig. 1.

Color Coordinates and Chrominance Subsampling. Images are often displayed by the cathode ray tube (*CRT*) using red (R), green (G), and blue (B) phosphor emissions. In compression, however, the RGB color coordinate is not the most efficient for representing the color components of images or video. It is known that the luminance (the intensity of the light, the gray-scale projection of the image) is more important than the chrominance (colors hue and saturation) components in human visual perception. Therefore it is

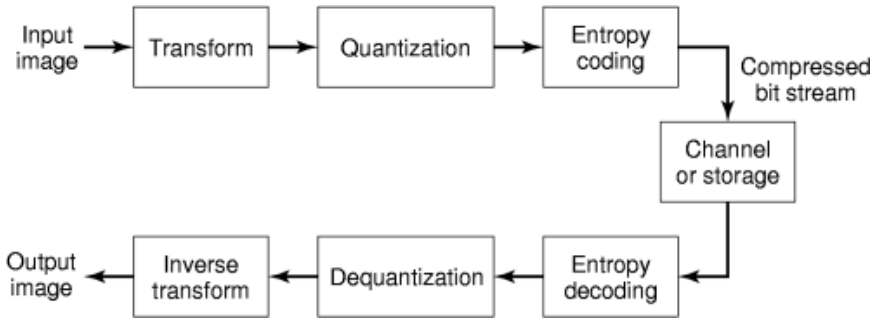

Fig. 1. Block diagram of a general image transform coder. The decoder performs the inverse steps to reconstruct the image.

preferable to transform the color components from the RGB color coordinate to some luminance-chrominance representation so that we put more emphasis on the luminance and discard more unimportant information from the chrominance components without affecting much of the visual perception of compressed images/video. Three often used luminance-chrominance color coordinate systems are YUV, YIQ, and YCbCr color spaces.

The YUV color coordinate was developed by National Television Systems Committee (*NTSC*) and now used in Phase Alternation Line (*PAL*) and Sequentiel Couleur Avec Mémoire (*SECAM*) color television systems. NTSC later developed the YIQ coordinate by rotating the U and V components in YUV space to further reduce the color component bandwidths. The luminance Y and the color components U, V and I, Q in their respective coordinates can be transformed via

$$\begin{bmatrix} \mathrm{Y} \\ \mathrm{U} \\ \mathrm{V} \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.148 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} \mathrm{R} \\ \mathrm{G} \\ \mathrm{B} \end{bmatrix}$$

and

$$\begin{bmatrix} \mathrm{Y} \\ \mathrm{I} \\ \mathrm{Q} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -\sin(33^\circ) & \cos(33^\circ) \\ 0 & \cos(33^\circ) & \sin(33^\circ) \end{bmatrix} \begin{bmatrix} \mathrm{Y} \\ \mathrm{U} \\ \mathrm{V} \end{bmatrix}$$

The YCbCr color coordinate was developed as the standard color coordinate system for digital video, as described in ITU-R Recommendation 601 (1). It is an offset and scaled version of the YUV coordinate to limit the dynamic range of luminance and chrominance components within the dynamic ranges of the original RGB components:

$$\begin{bmatrix} \mathrm{Y} \\ \mathrm{Cb} \\ \mathrm{Cr} \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} \mathrm{R} \\ \mathrm{G} \\ \mathrm{B} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.500 \\ 0.500 \end{bmatrix}$$

Because the human visual system is less sensitive to chrominance, the two chrominance components Cb and Cr are usually subsampled to reduce the data volume before compression. Several subsampling formats are commonly used. Two of them, the 4:2:2 format and 4:2:0 format, are used in image/video coding standards, such

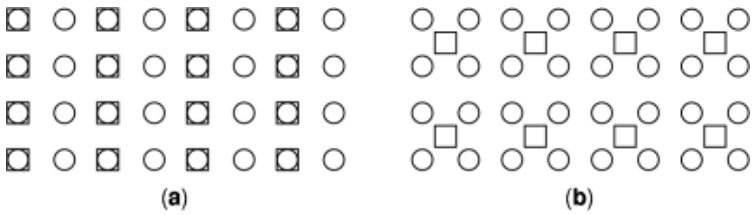

Fig. 2. Different subsampling formats differ in both the ratio of luminance and chrominance samples and their relative positions. Two YCbCr subsampling formats are shown in this figure: (a) 4:2:2, (b) 4:2:0 format. Luminance (Y) and chrominance (Cb, Cr) samples are represented by circles and squares, respectively.

as *JPEG* (Joint Photographic Experts Group) and *MPEG* (Motion Picture Experts Group). These two formats are shown in Fig. 2.

Transform. Transform coding is the most popular coding scheme in scholarly research and industry standards. The purpose of the transformation is to map the digital image from the spatial domain to some transform domain so that its total energy is packed in a small portion of transform coefficients, whereas most other transform coefficients are very close to zero. We can coarsely quantize these unimportant coefficients or simply throw them away in later steps to achieve the goal of compression.

There are several other desirable properties for the transforms used in transform coding. First, a unique inverse transform is required because we have to recover the image from its transform domain. Second, the transform should conserve the total energy in the image. Unitary transforms satisfy these requirements. But not all unitary transforms are suitable for image compression. The energy compaction property and the computational complexity of the transforms are always as important in practical implementation. The optimal transform for energy compaction is known as the Karhunen–Loève transform (*KLT*), but the computational complexity is too high to be practical. Most of the compression standards use the discrete cosine transform (*DCT*). It has a good energy compaction property, and fast algorithms for its forward and inverse transforms are available. Wavelet transform is another promising transform for transform coding and is described in a later section.

Quantization. Most of the transforms convert integral pixel values into floating-point transform coefficients. Encoding these coefficients as floating-point numbers is not economic for lossy compression. Quantization is the process of converting continuous numbers to discrete-value samples. Most transform coding techniques use scalar quantization. The principles of quantization are described in Data compression codes—Lossy and are not discussed in detail here. The output of the scalar quantizer is the index of the reconstruction level. Because quantization is a many-to-one mapping, this is the primary step that causes loss of information in the whole transform coding process.

Entropy Coding. The final stage of the transform coder is to losslessly compress the quantization indexes using an entropy coder for further reduction of compressed bit-stream volume. Two often used entropy coders are the Huffman coder and the arithmetic coder. The details of entropy coders are described in Data compression, lossy.

Image Error Measures. To evaluate the performance of image compression techniques, proper image error measures, which evaluate the difference between the original and compressed images, are necessary. A commonly used image error measure is the mean square error (*MSE*), defined as

$$\mathrm{MSE} = E\{[X(i,j) - X'(i,j)]^2\}$$

where $\mathrm{E}\{\cdot\}$ is the expectation operator, X and X' represent the original and compressed images, respectively, and i, j are the image coordinates of the pixel. The peak signal-to-noise ratio (*PSNR*) is more frequently used

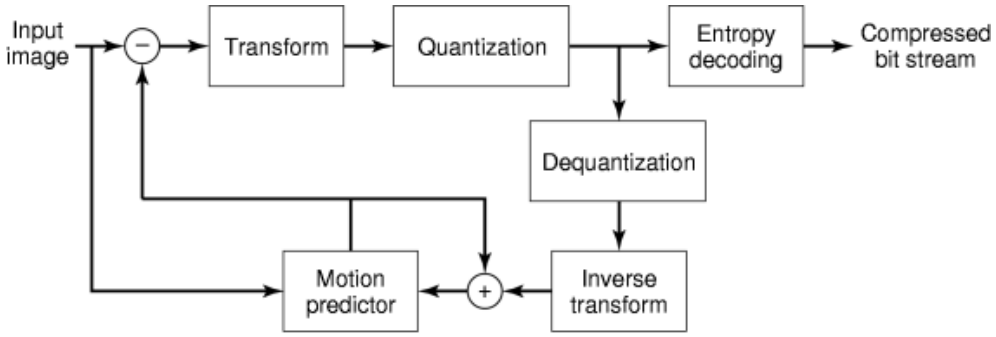

Fig. 3. Block diagram of a general video transform coder using motion-compensated predictive coding. A motion predictor is used to find the motion vector and the estimation error is also transform coded.

in image compression:

$$\text{PSNR} = 10 \log_{10} \left(\frac{P^2}{\text{MSE}} \right) \text{dB}$$

where P is the peak input amplitude. For example, for an eight-bit gray-scale image, $P = 255$. A lower MSE or higher PSNR value means that the compressed image has higher fidelity to the original image. Both MSE and PSNR are conventionally used for gray-scale image error evaluation. There are no consensual error measures for color image compression yet.

Motion-Compensated Predictive Coding. The temporal domain is involved in digital video compression. A digital video is a series of images, called pictures or frames, to be played sequentially. A straightforward way of compressing digital video by image compression techniques is to treat each frame as independent images and compress them separately. However, digital video has redundancies in the temporal domain that are exploited for further compression. Unless there are scene changes, videos usually have many of the same objects in adjacent frames, though the spatial locations on each frame may differ because of object motion. It is a waste to code the same objects on different frames repeatedly. We can encode the object on the first frame and only the direction and distance of object motion in subsequent frames. At the decoder end, after the first frame is decoded, the object on subsequent frames is reconstructed by pasting the object at different spatial locations according to the object motion information. The objection motion direction and distance information are called the motion vector (*MV*), the process to estimate the motion vector between adjacent frames is called motion estimation (*ME*), and the scheme to perform ME and paste the object with respect to the motion vector is called motion compensation (*MC*).

The same object appearing on adjacent frames, however, may appear differently because of light reflection, object rotation, cameras panning or zooming, and so on. Furthermore, new objects may appear which cannot be well estimated with other objects on the previous frame. Therefore motion compensation is only a prediction from previous frames. The difference between the prediction and the actual pixel values has to be computed and encoded. The prediction error, however, would be quite small as long as the ME algorithm finds the minimal-error match from previous frames. The error histogram usually has its peak around zero with small probabilities at large values. The prediction error can be quantized and encoded very efficiently. The block diagram of a general video coder using motion-compensated predictive coding is shown in Fig. 3.

Compression Standards

JPEG Standard. The Joint Photographic Experts Group (*JPEG*) is a working group formed in 1982 under the International Organization for Standardization (*ISO*). This group joined the International Organization for Standardization Consultative Committee (*CCITT*) Special Rapporteur Group (*SRG*) on New Forms of Image Communication to establish an international standard for digital color image compression. After evaluating numerous proposals, they completed the draft technical specification in 1990, and the draft became an international standard in 1992. Some further extensions were developed in 1994. The resulting standard (2), also called JPEG, is now used worldwide for still, continuous-tone, monochrome, and color image compression.

The original JPEG quality requirement is to have indistinguishable images when compressed at 1.50 bits to 2.00 bits per pixel (bpp), excellent image quality at 0.75 bpp to 1.50 bpp, good to very good quality at 0.50 bpp to 0.75 bpp, and moderate to good quality at 0.25 bpp to 0.50 bpp. There are four modes of JPEG operation. They are the sequential baseline DCT-based mode, the progressive DCT-based mode, the sequential lossless mode, and the hierarchical mode. These four modes provide different compression techniques for applications with different requirements. The baseline mode, however, is the mode most often used. The three other modes are rarely used so that many JPEG decode software programs do not even support them.

Baseline Mode. The block diagram of the sequential baseline DCT-based mode JPEG coder and decoder is similar to that shown in Fig. 1. The color image is first converted into the YCbCr coordinates, and then the three components are compressed separately. The core transform used is the discrete cosine transform (DCT), which transforms spatial-domain pixel values into frequency-domain coefficients. To represent the DCT coefficients with 11-bit precision for eight-bit input image (and 15-bit precision for 12-bit input), the three color components in the YCbCr space are level shifted by subtracting 128 (or 2048 for 12-bit input) before performing the DCT. For computational efficiency, the whole input image is partitioned into square blocks of 8×8 pixels each. Then the two-dimensional 8×8 DCT is performed on each block separately:

$$S_{uv} = \frac{C_u}{2}\frac{C_v}{2}\sum_{x=0}^{7}\sum_{y=0}^{7} s_{yx}\cos[(2x+1)u\pi/16]\cos[(2y+1)v\pi/16]$$

$$C_u = \begin{cases} 1/\sqrt{2} & \text{for } u = 0 \\ 1 & \text{for } u > 0, \end{cases} \quad C_v = \begin{cases} 1/\sqrt{2} & \text{for } v = 0 \\ 1 & \text{for } v > 0 \end{cases}$$

where s and S are the 2-D spatial-domain pixel values and the 2-D DCT coefficients, respectively. The subscripts yx and vu are the spatial-domain and frequency-domain coordinates, respectively. S_{00} is called the DC coefficient, and the rest of the 63 coefficients are called AC coefficients. The 8×8 inverse discrete cosine transform (IDCT) used at the decoder end is given by

$$s_{yx} = \frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7} C_u C_v S_{vu}\cos[(2x+1)u\pi/16]\cos[(2y+1)v\pi/16]$$

and 128 (or 2048 for 12-bit input) is added back to restore the original pixel value levels. Numerous fast DCT and IDCT algorithms are available but are not discussed in detail here.

After the DCT operation, the 64 DCT coefficients are quantized by using a lookup table, called the quantization matrix. The default quantization matrices for luminance and chrominance are different because the Human Visual System (*HVS*) has different luminance and color perception characteristics. These two default quantization matrices Q_{vu} are given in Table 1, where the lowest frequency components Q_{00}'s are in the upper left corners. The encoder is also free to define its own quantization matrices, but they have to be

Table 1. JPEG Default Quantization Matrices

(a) Luminance Quantization Matrix

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

(b) Chrominance Quantization Matrix

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

included in the compressed data for the decoder to reconstruct the DCT coefficients. The quantization indexes Sq_{vu} are obtained by dividing the floating-point DCT coefficients by the quantization matrix and rounding the quotient to the nearest integer:

$$Sq_{vu} = \mathbf{R}\text{ound}(S_{vu}/Q_{vu})$$

The reconstructed DCT coefficients R_{vu} are obtained at the decoder side by multiplying the quantization indexes by the quantization matrix:

$$R_{vu} = Sq_{vu}Q_{vu}$$

Because the DC coefficients represent the mean values of the partitioned 8×8 blocks, these coefficients among adjacent blocks are usually quite close to each other in natural images. Thus they are encoded with differential coding in the raster-scan order to take advantage of this property. With the DCT energy compaction property, most of the energy of each 8×8 block is concentrated in low frequencies. Therefore the 63 AC coefficients are encoded in a zigzag order so that the significant coefficients are likely to be encoded first, and in most cases, there are consecutive zero AC coefficients near the end of the block so that they are encoded very efficiently. The differential coding of DC coefficients and the zigzag coding order of AC coefficients is shown in Fig. 4.

Two DC and two AC Huffman tables are used for entropy coding the DCT coefficients. The DC Huffman table for eight-bit resolution is shown in Table 2. The differential (*DIFF*) values range from -2047 to 2047, are classified into 12 categories, denoted as SSSS, and are coded by variable-length codewords. The AC coefficients range from -1023 to 1023 and are classified into 10 nonzero SSSS categories. Because runs of zeros are likely at the high frequencies along the zigzag scan, the lengths of zero runs are encoded with 15 four-bit categories, denoted as RRRR. The combination of RRRRSSSS is encoded using the AC Huffman table with 162 possible

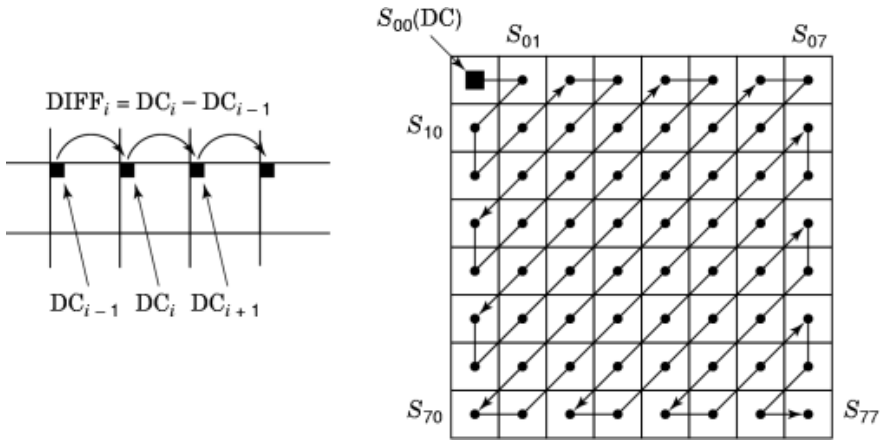

Fig. 4. Differential coding of DC coefficients (left) and zigzag scan order of AC coefficients (right). The differential coding of DC coefficients takes advantage of the cross-block correlation of DC values, whereas the zigzag scan order takes advantage of the energy compaction property so that it is very likely to have consecutive zeros toward the end of the block.

Table 2. Huffman Encoding for DC Coefficients

		Luminance		Chrominance	
DIFF Values	SSSS	Code Length	Code-word	Code Length	Code-word
0	0	2	00	2	00
21, 1	1	3	010	2	01
23, 22, 2, 3	2	3	011	2	10
27 . . . 24, 4 . . . 7	3	3	100	3	110
215 . . . 28, 8 . . . 15	4	3	101	4	1110
231 . . . 216, 16 . . . 31	5	3	110	5	11110
263 . . . 232, 32 . . . 63	6	4	1110	6	111110
2127 . . . 264, 64 . . . 127	7	5	11110	7	1111110
2255 . . . 2128, 128 . . . 255	8	6	111110	8	11111110
2511 . . . 2256, 256 . . . 511	9	7	1111110	9	111111110
21023 . . . 2512, 512 . . . 1023	10	8	11111110	10	1111111110
22047 . . . 21024, 1024 . . . 2047	11	9	111111110	11	11111111110

codes, which are not listed here. A particular RRRRSSSS code is used for zero runs with lengths exceeding 15, and another particular code is used to denote the end of block (*EOB*) when all remaining quantized coefficients in the block are zero.

Progressive Mode. In the baseline mode, the blocks in an image are encoded and decoded in the raster-scan order, that is, from left to right and from top to bottom. The decoder has to receive all the details for one block, decode it, and then proceed to the next block. If, for some reason, the bit stream is cut midway during

the transmission, there is no way the decoder-end user would know the content of the rest of the image. The progressive DCT-based mode uses multiple scans through the image. The DC coefficients of all blocks from the whole image are transmitted first, then the first several AC coefficients of the whole images are transmitted, and so on. In this way, even if the bit stream is cut midway during transmission, it is possible that the whole image with coarser resolution is already available for the decoder-end user to perceive the image content. Progressive image transmission is particularly preferable for image browsing over transmission channels with limited bandwidth. The user can decode the rough image to see if this image carries the required information. If it does, the user can continue the transmission to add more and more details to the image. Otherwise, the user can stop the transmission.

In practice, all DCT coefficients are computed as in the baseline mode and stored in a buffer. The encoder is free to choose the scan number and the coefficients to be transmitted in each scan. For example, the encoder may choose to send the DC coefficients S_{00} in the first scan, S_{01} and S_{10} in the second scan, S_{20}, S_{11}, S_{02} in the third run, S_{03}, S_{12}, S_{21}, S_{30} in the fourth run, and the rest of AC coefficients in the fifth. This choice, called spectral selection, is up to the encoder and can be specified explicitly in the scan header of the compressed bit stream.

In addition to intercoefficient spectral selection, intracoefficient successive approximation is also used for progressive transmission. In short, the successive approximation scheme quantizes the coefficient with lower precision so that a shorter bit stream is transmitted. In every subsequent stage, one more truncated bit is added back to improve the precision of the coefficients by one bit until full precision is reached.

Lossless Mode. Although primarily focused on lossy image compression, JPEG also provides a lossless compression mode. Rather than using the float-point DCT process that introduces error with integral quantization, the lossless mode uses 2-D predictive coding. This predictive coding method uses the upper, left, and upper left neighbor pixels to predict the present pixel value. One of the seven prediction types is chosen and specified in the scan header. The pixels are encoded according to the predictor selected. The lossless mode allows input precision from 2 to 16 bits/sample. The difference between the prediction value and the input is computed modulo 2^{16} and encoded using the Huffman table in Table 2, except that extra entries are added at the end of the table to code the SSSS value from 0 to 16. Arithmetic coding of the modulo difference is also allowed but not discussed in detail here.

Hierarchical Mode. The last mode, the hierarchical mode, is used to generate subimages of smaller size and coarser resolution. First the original image is successively downsampled by a factor of 2 horizontally, vertically, or both. The subimages are smaller versions of the original image with lower resolution. The smallest subimage is transmitted. Then it is upsampled and interpolated bilinearly to form the prediction of the next higher resolution image. The prediction error is encoded and transmitted. The process is repeated until the original image size and resolution are achieved. At the decoder end, a similar process is used to reconstruct the original image by upsampling and adding the prediction error to form multiresolution images. The encoding method can be one of the other three modes: sequential, progressive, or lossless. The hierarchical mode is preferable for platforms with a lower resolution display device or with limited computational power insufficient for reconstructing full-sized images.

JPEG 2000 Standard. New image compression techniques have emerged in recent years since the development of JPEG standard. In addition, JPEG either does not support or does not perform well for some recent applications, such as side-channel information and very low bit-rate coding. All of these encourage the creation of second-generation image compression standards. JPEG 2000, aiming to become an International Standard (*IS*) in year 2000, is the ongoing project for this purpose (3).

The goal of JPEG 2000 is to create a unified system for compressing different types of images (bilevel, gray-scale, or color) with various characteristics and conditions. The purpose of this standard is to complement but not replace the current JPEG standard. Therefore it will focus mainly on the applications for which JPEG fails to provide acceptable quality or performance. The new features of JPEG 2000 will likely include

- High-performance, low bit-rate compression: JPEG performs poorly at a low bitrate, where apparent blocking artifacts appear. JPEG 2000 intends to improve the rate-distortion performance at low bit rates, for example, below 0.25 bpp for gray-scale images, while keeping the excellent performance at higher bit rates. This is the most important function of JPEG 2000.
- Various-type image compression: JPEG focuses mainly on lossy gray-scale and color image compression. The standard for bilevel (such as text) image compression is currently the Joint Bilevel Image Experts Group (*JBIG*) standard. In addition, although providing a lossless mode, JPEG does not perform particularly well in that aspect. JPEG 2000 aims to provide efficient lossless and lossy compression of images with a wide dynamic range, such as bilevel, gray-scale, and color images, all within a unified architecture.
- Robustness to errors: JPEG performs poorly if there is bit error in the bit stream during transmission and storage, for example, when compressed data is transmitted through a noisy channel. JPEG 2000 will incorporate error-resilience or error-correction capabilities into the standard so that the compression bit stream is robust to unexpected errors.
- Content-based description and MPEG-4 interface: one of the most challenging topics in image compression is extracting the semantic content of images and its objects. It benefits applications, such as image retrieval and object-based compression. JPEG 2000 intends to shed light on this problem and hopefully to find some solution for it. In addition, the new video compression method MPEG-4 will use a descriptive language to describe the objects and provide methods to code them. JPEG 2000 is expected to provide an interface for object description and compression for objects.

Other features, including (but not limited to) fixed bit-rate compression, image security, such as image watermarking and encryption, side channel (such as alpha channel and transparency plane) information, random access and processing on arbitrary regions, are also expected to be incorporated into this standard. Though the transform, even the whole framework of JPEG 2000 is likely to be quite different from that used in the existing JPEG, it is desirable that JPEG 2000 should be backward-compatible for JPEG.

MPEG-1 and MPEG-2. The Moving Picture Experts Group (*MPEG*), another working group under ISO, was formed in 1988. It developed the original video coding standard, which was also commonly called MPEG later, for video and associated audio compression. Most of the MPEG parts became an international standard in 1992 (4). Different from JPEG, the second-generation project of MPEG started right after MPEG was completed. To distinguish among generations of MPEGs, the first generation of MPEG is often called MPEG-1. The three essential parts of MPEGs are : video, audio, and systems. We focus only on the video part of this coding standard.

The objective of MPEG-1 is to provide approximately VHS quality of compressed video with a medium bandwidth for 1 to 1.8 Mbps (Mbits/s). It is used for strictly noninterlaced (progressively scanned) video and is optimized for CD-ROM, video CD, and CD-interactive (CD-i) applications. The dimension limits for parameter-constrained video are 768 (h) $\times$ 576 (v) $\times$ 30 (frames/s, fps).

An MPEG-1 video stream includes several hierarchical layers. The whole video sequence is partitioned into at least one *group of pictures* (*GOP*), intended to allow random access into the sequence. Each GOP consists of a certain number of *pictures.* The picture, also called a frame, is the primary coding unit of a video sequence. Each picture consists of three rectangular matrices representing luminance (Y) and two chrominance (Cb, Cr) values. In MPEG-1, the YCbCr matrices are sampled by the 4:2:0 format, that is, the Cb and Cr matrices are subsampled by two horizontally and vertically, therefore their sizes are one-quarter of the Y matrix. Each picture consists of one or more *slices.* The major purpose of slices is to isolate the error in the transmitted bit stream. In the case of a noisy channel, the decoder can skip the erroneous slice and start decoding with the next slice. In an error-free environment, the encoder usually assigns the whole picture to one slice. Each slice is composed of one or more *macroblocks.* The macroblock is the basic coding unit in MPEG. The size of each macroblock is 16 $\times$ 16. In the 4:2:0 format, it consists of six 8 $\times$ 8 *blocks,* four of which are luminance (Y)

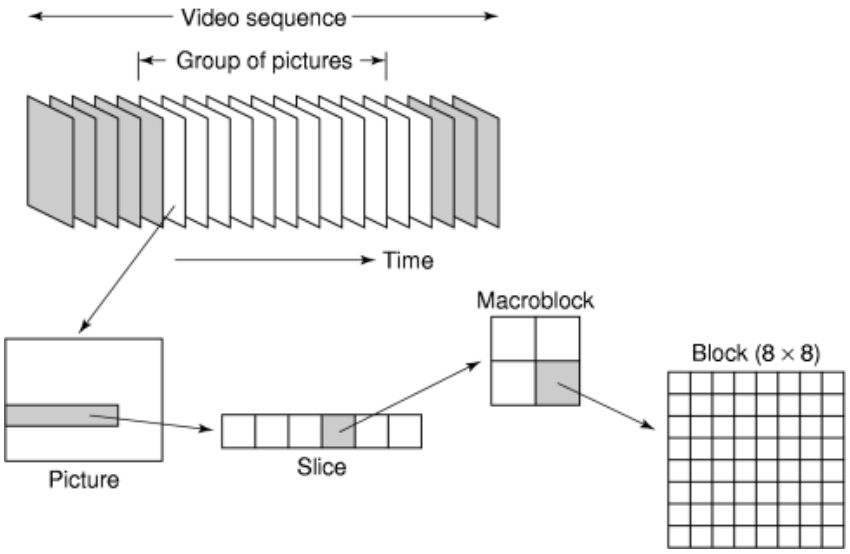

Fig. 5. Video stream data hierarchical layers in MPEG. The four blocks in the macroblocks represent the luminance (Y) blocks in the macroblock. The two 8×8 chrominance (Cb and Cr) blocks are downsampled from the 16×16 macroblock and are not shown in this figure.

blocks, and the other two are downsampled chrominance (Cb and Cr) blocks. This video stream data hierarchy is shown in Fig. 5.

MPEG-1 uses a DCT-based transform coding scheme to reduce the spatial redundancy, and the motion-compensation technique to reduce the temporal redundancy. MPEG defines four types of pictures: intra (I), predicted (P), bidirectional (B), and DC (D). The D pictures are used only for the fast-forward mode, in which only the DC value of each block is encoded for fast decoding. This type of picture cannot be used in conjunction with the other three types of pictures. It is seldom used thus is not discussed in detail here. A GOP must contain at least one I picture, and may be followed by any number of I, P, B pictures. The I picture is intracoded, which means that it is coded using a still image-coding technique without any temporal dependency on other pictures. The P picture is predicted from a previous I or P picture with its motion information. The B picture is inserted between two I or P pictures (or one of each) and is bidirectionally interpolated from both pictures. With this dependency of I, P, and B pictures, B pictures must be encoded after I or P pictures even though they are displayed before them. Figure 6 shows the picture dependency and the difference between video stream order and display order. Note that this figure only serves as an example of the way the three types of pictures are organized. The actual number of I, P, B pictures in a GOP can be specified arbitrarily by the encoder.

The I pictures are encoded using a JPEG-like image coding scheme. Each 8×8 block is level-shifted and transformed by using the 8×8 DCT and then quantized by using a default or user-defined intraquantization matrix. The default intraquantization matrix is shown in Table 3(a). The intraquantization matrix can be multiplied by a quantizer scale factor from 1 to 31 from macroblock to macroblock to achieve different bit rates, but the quantization step of DC coefficient is always set to eight. The DC coefficients are differential-coded, whereas the AC coefficients are zigzag scanned and encoded by using run-length coding and Huffman coding, which is similar (but not identical) to JPEG.

The motion-compensation technique is used in coding P pictures. For each 16×16 macroblock in the P picture, the most similar macroblock is found in the preceding I or P picture as the prediction or estimation

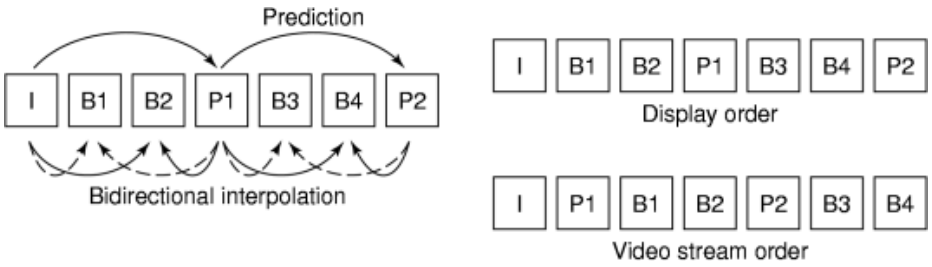

Fig. 6. Interpicture dependency (left) and picture order (right) in MPEG. While the encoder transmits the P frames before the B frames in order to provide interpolation information, it is necessary for the decoder to put the first decoded P frames in a buffer until the subsequent B frames are decoded and to rearrange them for display.

Table 3. MPEG-1 Default Quantization Matrices

(a) Intraquantization Matrix

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

(b) Nonintraquantization Matrix

16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16

of the target macroblock. A motion vector is used to record the spatial displacement between the original and estimated macroblocks in their respective pictures.

The MPEG-1 standard does not define the similarity between two macroblocks and the method of searching for the *most similar* macroblock in the reference picture. The encoder is free to develop its own similarity criterion and motion vector searching algorithm. Two distortion definitions are commonly used to provide the similarity measure:

- Mean squared error (MSE):

$$\mathrm{MSE}(k,l) = \frac{1}{16 \times 16} \sum_{i=0}^{15} \sum_{j=0}^{15} [S_1(u_1+i, v_1+j) - S_2(u_1+i+k, v_1+j+l)]^2$$

- Mean absolute error (MAE):

$$\mathrm{MAE}(k,l) = \frac{1}{16 \times 16} \sum_{i=0}^{15} \sum_{j=0}^{15} \left| S_1(u_1 + i, v_1 + j) - S_2(u_1 + i + k, v_1 + j + l) \right|$$

where S_1 and S_2 are the target and reference pictures, u_1, v_1 are the upper left coordinates in S_1 and S_2, and (k, l) is the MV.

When comparing two MSEs or MAEs, the division of 16×16 is a common factor and thus can be dropped. The smaller the MSE or MAE, the more similar the two macroblocks. The MAE has lower computational complexity and therefore is used more often in MPEG coders. The purpose of the motion vector searching algorithm is to find the MV with the smallest MSE or MAE, and choose it as the best MV representing the motion of the macroblock.

MPEG-1 allows the motion vector to take a large range of displacements in the picture from the reference macroblock. The computational cost to search the whole range, however, is too high to be practical. An efficient encoder usually limits its search to a reasonable range, say, in a 32×32 neighborhood region. This 32×32 region is conventionally called a [−16,15] searching window because the horizontal and vertical displacements for the MV are confined to the [−16, 15] range. The simplest searching algorithm is the full search, that is, to sweeping this searching window pixel-by-pixel and finding the macroblock with the least error. The computational cost is greatly reduced with the logarithmic searching algorithm. In the first step of the logarithmic searching algorithm, eight MVs with large displacement from the starting pixel are selected. The eight errors with respect to these eight MVs are computed and compared to find the smallest one. In the second step, the starting point is taken as the coordinate associated with the smallest error, and the searching displacement is halved. A similar process is repeated until the smallest displacement (one pixel) is met. The process of the full search and the logarithmic search are shown in Fig. 7. MPEG-1 allows half-pixel MV precision if it gives better matching results. The pixel values are bilinearly interpolated to achieve this half-pixel precision. The searching algorithms have to be modified accordingly and the searching range is four times as large with half-pixel precision. Generally speaking, MV search is the most computationally expensive part of the whole MPEG coding process. Many new and efficient searching algorithms have been developed and adopted by commercial encoders.

Depending on the magnitude of motion vector and the prediction error, various coding types can be selected for P picture coding based on the following decisions. How to make these selection decisions is left to the encoder and not explicitly defined in MPEG-1.

- Intra/nonintra: if the intracoding of the macroblock takes less bits than coding the motion vector and the prediction error, we may simply use intracoding as used in I pictures, else nonintracoding of the MV's and prediction errors is used.
- MC/no MC: if the motion vector is very close to zero, we may avoid using MC to save the bits for encoding MVs.
- New/old quantizer scale: if the currently used quantizer scale is not adequate for coding, for example, unable to satisfy the current bit-rate constraint, it may be changed to a new value.
- Coded/not coded: in the nonintracoding case, if the coefficients of a *block* are all zero, the whole block is not coded. In this way a significant number of bits is saved. Because a *macroblock* includes six blocks, if at least one block in a macroblock has to be coded, a coded block pattern has to be included in the bit stream to inform the decoder which blocks in the macroblock are actually encoded.

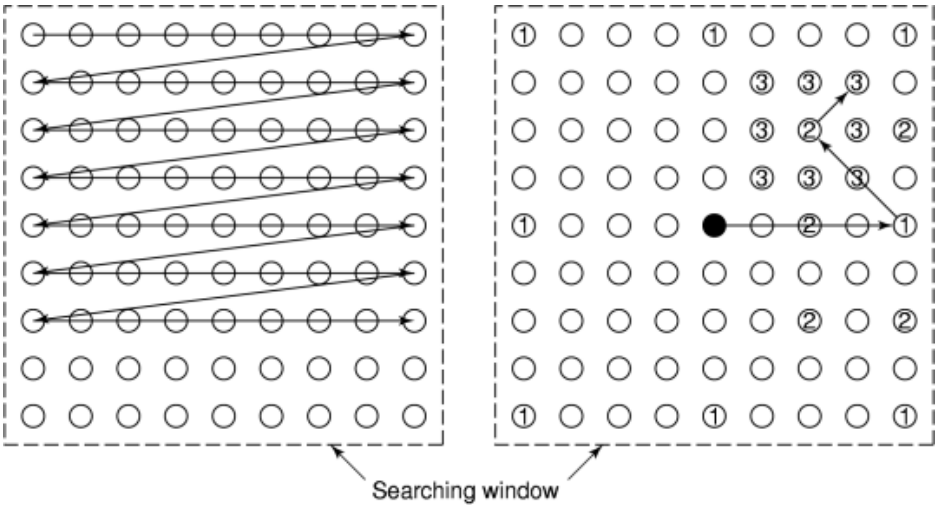

Fig. 7. Two MV searching algorithms: full search (left) and logarithmic search (right), where 1, 2, and 3 indicate the step numbers. In this illustrative example, it takes only 21 MV computations to obtain the best motion vector. On the other hand, all the 81 possible MVs have to be computed in the full search algorithm.

These four decisions generate eight different macroblock types and each is coded differently. A noteworthy macroblock type is the skipped block type, which is not intracoded, having all zero coefficients after quantization, and no MV and quantizer scale change is needed. In other words, the macroblock is identical to the macroblock in the previous I or P picture at exactly the same location. No variable-length code (*VLC*) is needed for this type of macroblocks

In most cases, both the MVs and the prediction errors have to be coded. The MVs of adjacent macroblocks are likely to have similar values because adjacent macroblocks are likely to move coherently in the same direction. Therefore the MVs are encoded with differential coding and further entropy-coded by a variable-length Huffman code. The 16×16 prediction error is encoded by using the transform coding technique similar to that used in encoding I pictures but with a nonintraquantization matrix. The default nonintraquantization matrix is shown in Table 3(b). Another difference from intrablock coding is that the DC coefficients are encoded together with all AC coefficients rather than using a separate differential coder.

The B pictures are obtained by bidirectionally interpolating I or P pictures. At first, one of three MC modes (forward, backward, interpolated) is selected. If the forward mode is selected, a macroblock from the previous I or P picture and the forward MV is used as the prediction. If the backward mode is selected, a macroblock from the future I or P picture and the backward MV is used. If the interpolated mode is selected, one macroblock from the previous and one from the future pictures are bilinearly interpolated to yield the prediction, and both the forward and backward MVs are transmitted.

Similar to P picture coding, three decisions (except for the MC/no MC decision, because all macroblocks have to be motion compensated) have to be made. There are a total of 11 macroblock types in coding B pictures. Different from P picture coding, a skipped block in a B picture uses the same motion vector and same macroblock type as its previous block.

The DCT, quantization, and variable-length codes for B pictures are the same as those of P pictures.

MPEG-2, the second generation of MPEG, focuses primarily on high-quality compressed video for broadcasting. The typical bit rate is 4 Mbps to 15 Mbps, compared to the 1.5 Mbps for MPEG-1. The major applications include digital video disk (*DVD*), digital video broadcasting (*DVB*), and TV systems, such as NTSC and PAL. There was a plan to develop MPEG-3 for high definition television (*HDTV*) applications, but it was later merged

Table 4. MPEG-2 Levels and Profiles

(a)

Level	Max. Dimensions, h × v × fps	Pixels/s	Max. Bit Rate	Significance
Low	352 × 240 × 30	3.04 M	4 Mbps	SIF, consumer tape equiv.
Main	720 × 480 × 30	10.40 M	15 Mbps	CCIR 601, studio TV
High 1440	1440 × 1152 × 30	47.00 M	60 Mbps	4 × CCIR 601, consumer HDTV
High	1920 × 1080 × 30	62.70 M	80 Mbps	Production SMPTE 240 Standard

(b)

Profile	Comments
Simple	Same as Main, only without B-pictures. Intended for software applications, perhaps CATV.
Main	Most decoder chips, CATV satellite. 95% of users.
Main1	Main with spatial and SNR scalability.
High	Main1 with 4:2:2 chroma format.

(c)

	Profile			
Level	Simple	Main	Main1	High
Low	Illegal	∪	Main with SNR scalability	Illegal
Main	∪	∪ (90% of users)	Main with SNR scalability	4:2:2 chroma
High 1440	Illegal	∪	With spatial scalability	4:2:2 chroma
High	Illegal	∪	Illegal	4:2:2 chroma

with MPEG-2. Three parts (systems, video, and audio) of MPEG-2 became IS in 1994, and the other parts were adopted from 1995 to 1997.

Instead of simply providing video compression as in MPEG-1, MPEG-2 focuses on providing more functionality to various applications. MPEG-2 supports both interlaced and noninterlaced video and has a wider range of picture sizes, called *levels.* MPEG-2 also includes several *profiles* to provide different functionality. The combination of levels and profiles for different applications is shown in Table 4. MPEG-2 also provides four scalable modes in Main+ and High profiles for decoders with different capabilities. The *spatial scalable mode* provides two spatial resolution video layers. The *SNR scalable mode* provides two video layers of the same resolution but different quality. The *temporal scalable mode* has one lower layer coded at the basic temporal rate, and the second enhancement layer uses the lower layer as its prediction. The *data partitioning mode* uses progressive transmission which is similar to JPEG's progressive mode.

The MPEG-2 coding scheme (5) is similar to that of MPEG-1 with some modifications and additions for the extra functionality not handled in MPEG-1. The MPEG-2 video syntax (for the main profile) is a superset of that of MPEG-1.

MPEG-4 and MPEG-7. MPEG-4 is an ongoing project of the MPEG family. MPEG-4 (6) focuses mainly on very low bit-rate (64 kbps or less) applications. The major applications are mobile or other telecommunication video applications (5 kbps to 64 kbps) and TV/film applications (up to 2 Mbps). It is expected to be finalized by November 1998.

The three major features of MPEG-4 are *content-based interactivity, compression,* and *universal access.* MPEG-4 plans to provide the functionality of these features to bridge among the TV/film audiovisual data, wireless telecommunication, and computer interactivity.

Topics involved in content-based interactivity include the following:

- Content-based multimedia data access tools: the objects in the coding sequence are segmented and called audio-visual objects (*AVO*). The video is separated into video object planes (*VOP*). Each VOP may have different spatial and temporal resolutions, may have sub-VOPs, and can be associated with different degrees of accessibility, and may be either separated or overlapping.
- Content-based manipulation and bit-stream editing: manipulation and/or editing are allowed to be performed on a VOP, such as spatial position change, spatial scaling, moving speed change, VOP insertion and deletion, etc.
- Synthetic and natural hybrid coding (SNHC): MPEG-4 is intended to compress both natural and synthetic (cartoon, graphics, etc.) video. Issues include the processing of synthetic data in geometry and texture, real-time synchronization and control, integration of mixed media types, and temporal modeling.
- Improved temporal random access.

In compression, several important issues are addressed. The coding efficiency is improved to reduce the bit rate under 64 kbps for mobile applications and 2 Mbps for high-quality TV/film applications. In addition to the objective error measures, such as PSNR, the subjective quality should also be higher compared with existing standards. The ability to encode multiple concurrent data streams, such as the multiple views of a stereo video scene, is also provided. The most important breakthrough in compression, however, should be the object-based coding support. The encoder should be able to encode VOPs with arbitrary shapes, transmit the shape and transparency information of each VOP, support I, P, and B frames of VOPs, and encode the input VOP sequences at fixed and variable frame rates. The coding scheme of MPEG-4 is still block-based DCT coding. To support content-based coding, a square bounding box for an object is first found from the segmentation result. Motion compensation is performed on the macroblocks inside the bounding box. If the macroblock is inside the object, conventional block matching, as in MPEG-1, is used. If the macroblock is complete outside the object, no matching is performed. If the macroblock is partly outside and partly inside the object (that is, the macroblock is on the boundary of the object), some reference pixels have to be padded onto the nonobject area in the macroblock for block matching. Another method is to approximate the boundary macroblocks with polygons and perform polygon matching rather than square block matching.

Two issues are stressed in the universal access feature. One is the robustness in error-prone environments. Because one of MPEG-4's major applications is telecommunication, the channel error over wired and wireless networks has to be considered. The bit stream should be robust for severe error conditions, such as long bursty errors. The other issue is content-based scalability. Scalability in content (spatial, temporal, etc.), quality, and complexity should be allowed.

Because MPEG-4 is intended to provide compression for various types of applications with different bit rates, quality, source material, algorithms, and so on, a *toolbox* approach is adopted for the purpose of integration. In the toolbox approach, a proper *profile* is chosen to satisfy the requirements of the application. The coder selects a compression *algorithm* according to the profile and picks suitable *tools* to realize this

Table 5. Comparison Between H.320 and H.324

Recommendation	H.320 (1990)	H.324 (1995)
Commu. framing and demultiplexing	H.221	H.223
Control	H.230	H.245
Call setup: point-to-point	H.242	H.245
Call setup: multipoint	H.243	H.243
Video coding	H.261	H.263
Audio coding	G.711/G.722/G.728	H.723
Data	T.120	T.120
Network	Above 64 kbps	Below 64 kbps
Network interface	I.400	V.34
Typical network	ISDN-BRI	POTS(GSTN)

algorithm. The MPEG-4 system description language (*MSDL*) allows the transmission in the bit stream of the object structure, rules for the decoder, and the tools not available at the decoder. MPEG-4 also has a close relationship with the virtual reality modeling language (*VRML*) to transmit the description of a 2-D or 3-D scene.

MPEG-7 (7), the newest standard, is currently under development. It will focus mainly on multimedia database management, such as image query, indexing, and retrieval. It is expected to be closely tied with MPEG-4 content-based coding techniques to serve database management purposes.

H.263 Standard. In the 1980s, the International Telephone and Telegraph Consultative Committee (*CCITT*) started its research on low bit-rate videophone and videoconferencing, intended to be transmitted by communication channels with very low bandwidth, such as the telephone lines. The resulting ITU-T H-series recommendations include two H.32X recommendations and their subsystems. The first H.32X system Recommendation H.320, "Narrow-band visual telephone systems and terminal equipment," was finalized in 1990. It targets the bit rate $p \times 64$ kbps ($p = 1$–30). The more recent Recommendation H.324, "Multimedia terminal for low bit-rate visual telephone services over the PSTN," was finalized in 1995. It focuses on bit rates below 64 kbps. The comparison of these two Recommendations is shown in Table 5. We focus here only on the video coding standards H.263 (8).

The acceptable picture formats for H.263 are listed in Table 6. The 4:2:0 YCbCr format is used to represent color frames. Similar to the MPEG family, hierarchical layer representation is also used in H.263. Four layers are used in H.263: the picture layer, the group of blocks (*GOB*) layer, the macroblock layer, and the block layer. Each picture is divided into GOBs. The height of a GOB is defined as 16 pixels for SQCIF, QCIF, and CIF formats, 32 pixels for the 4CIF format, and 64 pixels for the 16CIF format. Each GOB is divided into macroblocks of size 16×16. Each macroblock includes four 8×8 luminance (Y) blocks and two 8×8 chrominance (Cb and Cr) blocks. The building blocks of the encoder include motion-compensation, transform-coding, quantization, and variable-length codes. There are two prediction modes. The *intermode* uses the information in a previous frame for MC. The *intramode* uses only information present in the picture itself. Similar to the picture types in the MPEG family, there are also I, P, and B pictures in the H.263 standard. However, because there is no GOP layer in H.263, the sequence is allowed to extend to an arbitrary number of frames without recurring patterns. However, because the prediction error propagates with each P picture coding, H.263 requires the insertion of at least one I picture in every 132 pictures. H.263 supports half-pixel precision MVs with the searching window of $[-16,15.5]$.

Table 6. Picture Formats Accepted by H.263

Picture Format	Luminance Pixels	Luminance Lines	H.261 Support	H.263 Support	Uncompressed Bit Rate, Mbps, at 10 Frames/s		Uncompressed Bit Rate, Mbps, at 30 Frames/s	
					Gray	Color	Gray	Color
SQCIF	128	96	No	Yes	1.0	1.5	3.0	4.4
QCIF	176	144	Yes	Yes	2.0	3.0	6.1	9.1
CIF	352	288	Optional	Optional	8.1	12.2	24.3	36.5
4CIF	704	576	No	Optional	32.4	48.7	97.3	146.0
16CIF	1408	1152	No	Optional	129.8	194.6	389.3	583.9

Except for the basic mode described above, there are four negotiable modes in H.263, which make the major differences between H.263 and H.261 (and the MPEG family): the *syntax-based arithmetic coding mode,* the *unrestricted motion-vector mode,* the *advanced-prediction mode,* and the *PB-frame mode.*

In the syntax-based arithmetic coding mode, an arithmetic code is used instead of VLC. Arithmetic codes usually provide better entropy-coding performance. The average gain for interframes is about 3% to 4%. For intrablocks and frames, the gain averages about 10%.

The unrestricted MV mode allows the MVs to point outside the picture. This mode is very useful when an object moves along or beyond the edge of the picture, especially for the smaller picture formats. The pixels on the edge row (or column) are replicated to cover the area outside the picture for block matching. This mode includes an extension of the motion-vector range from [−16,15.5] to [−31.5,31.5] so that larger motion vectors can be used.

The advanced prediction mode is used in conjunction with the unrestricted MV mode to achieve better motion prediction. This mode turns on the *four MV option* and the *overlapped-block motion-compensation* (*OBMC*) *option.* With the four MV option, four 8×8 vectors instead of one 16×16 vector are used for some macroblocks in the picture. The MV for the two chrominance blocks is obtained by averaging the four MVs then further dividing the average by two. The OBMC option is used for the luminance component of P pictures. For each 8×8 luminance prediction block, the MV of the current luminance block and two remote MVs are chosen. One remote MV is selected from the two MVs of the blocks to the left and right of the current luminance block and the other from the two MVs of the blocks above and below the current luminance block. Each pixel value in the prediction block is a weighted sum of the three predicted values obtained from the three MVs. The remote MV selection depends on the pixel position in the current block. If the pixel is on the left half of the block, the left-block MV is chosen, otherwise the right-block MV is chosen. The same is true for the top/bottom selection. If one of the surrounding blocks was not coded or was coded in intramode, the corresponding remote MV is set to zero. If the current block is at the border of the picture and therefore a surrounding block is not present, the corresponding remote MV is replaced by the current MV. The advantage of OBMC is that the blocking artifact is greatly reduced since every pixel is predicted by three overlapped blocks.

In the PB-frame mode, two pictures are coded as one unit, called a PB-frame. One P picture is predicted from the last decoded P picture. It is followed by one B-picture predicted from both the last decoded P picture and the P picture currently being decoded. Because most of the information transmitted is for the P picture in the PB-frame, the frame rate can be doubled with this mode without increasing the bit rate much for relatively simple sequences. For sequences with a lot of motion, however, PB-frames do not work as well as the B pictures in MPEG.

With H.263, it is possible to achieve the same quality as H.261 with 30–50% of the bit usage due to the half-pixel prediction and negotiable options in H.263. There are also less overhead and improved VLC tables in H.263. H.263 also outperforms MPEG-1/MPEG-2 for low resolution and low bit rates due to the use of the

four negotiable options. Another reason is that H.263 is less flexible than MPEG thus much less overhead is needed.

More enhancements were made for H.263 and summarized in the new H.263+, a near-term version of H.263. H.263+ supports more picture formats and provides more coding options. The advanced intracoding mode allows predicting an intrablock using neighboring intrablocks. The deblocking filter mode further reduces the blocking artifact by postfiltering the 8×8 block edges with a low-pass filter. The slice-structured mode provides MPEG-like slice structures which act as resynchronization points for bit error and packet loss recovery. The PB-frame mode is also improved to enhance predictive performance. It also adds temporal, SNR, and spatial scalability to H.263. Other enhancements are also made.

H.263L, another refinement of H.263, will be completed in a longer time frame than H.263+. A large change from H.263 and H.263+ is expected, and H.263L might be aligned with the MPEG-4 development and carry similar functionalities.

Other Compression Techniques

Wavelet Compression. Wavelet transform recently attracted a lot of attention in the transform coding field. It provides better performance than DCT-based coding techniques, both in high and low bit rates. Both JPEG 2000 and MPEG-4 are likely to adopt wavelet transforms in their codec.

The wavelet theory states that a signal can be represented by a series of translations and dilations of a basis function that meets certain mathematical requirements. Instead of using the global transformation as in DCT, the wavelet transform uses finite-impulse-response (*FIR*) filters to capture the space-frequency localization characteristics of the signal. This is usually accomplished by using the filter bank approach. The signal is passed through a quadrature mirror filter (*QMF*) bank consisting of a low- and high-pass filter pair denoted, respectively, by $h(k)$ and $g(k)$ with $g(k) = (-1)^k\, h(1-k)$. The forward transform is written as

$$c_k = \sqrt{2} \sum_n h(n-2k) f(n)$$

and

$$d_k = \sqrt{2} \sum_n g(n-2k) f(n)$$

whereas the inverse transform takes the form

$$f'[n] = \sqrt{2} \left(\sum_n h(n-2k) c_k + \sum_n g(n-2k) d_k \right)$$

The $h(n-2k)$ and $g(n-2k)$ are implemented by filtering followed by downsampling operations, when performing the forward transform, or filtering preceded by upsampling operations when performing the inverse transform. The low- or high-passed signals are called the subband signals. The data amount in each of these is half of that of the original signal because of the downsampling process. The 2-D wavelet transform is performed by cascading a horizontal filtering operation and a vertical filtering operation. Thus each subband after the 2-D transform has one-quarter the number of coefficients. The wavelet transform and the subband representation are shown in Fig. 8(a).

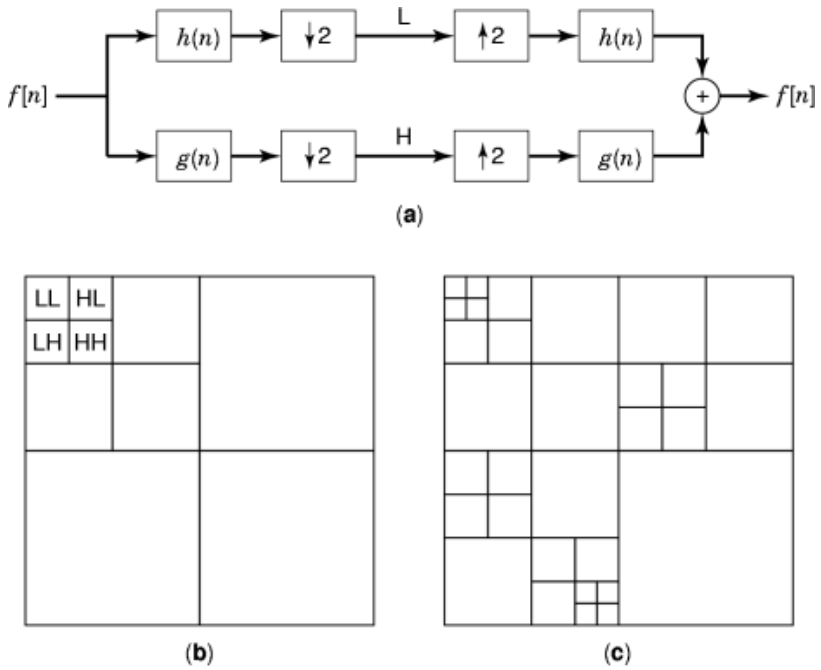

Fig. 8. Wavelet transform: (a) Filter bank implementation. The signal is filtered by high-pass and low-pass filters and downsampled by two. Each of the resulting low-pass and high-pass signals has half of the number of samples. (b) Pyramidal wavelet transforms. Each LL subband is further decomposed to form a regular pyramidal structure. (c) Wavelet packet transform. The subbands are further decomposed according to some criterions, for example, the energy distribution. They do not necessarily form a regular structure, therefore additional structural information has to be coded.

An advantage of wavelet transform in image processing is its flexibility to further decompose the image in the subbands of interest. With the desirable energy compaction property in mind, we can further decompose the subbands with higher energies to refine the bit-allocation strategy in these bands. This approach is called the wavelet packet transform (*WPT*). For most images, however, the low-pass subband usually has the highest energy. Therefore the successive decomposition of the LL band gives fairly good performance. This approach is called the pyramid wavelet transform. These two approaches are shown in Fig. 8(b) and (c).

Wavelet image compression started quite early with performance comparable to DCT compression. The advent of a series of modern wavelet coders, however, boosted the performance while providing a nice embedded bit-stream property. In an embedded bit stream, the transform coefficients are quantized successively so that the most significant coefficients are quantized and transmitted first. More details of the image can be successively added to refine the image if the bit rate allows. In this way, the bit rate is precisely controlled down to the bit level while keeping good performance.

The performance breakthough of modern wavelet coders results from exploiting the correlation between parent and child subbands. Shapiro's embedded zerotree wavelet (*EZW*) coder (9) partitioned the subbands into parent-child groups with same horizontal-vertical wavelet decomposition. If one coefficient in the parent subband is less than some threshold, then the coefficients in the corresponding child subbands are most likely also smaller than this threshold. Therefore only coding the *zerotree* root is enough. After the quantization and grouping procedure, the wavelet coefficients are represented by four symbols, positive, negative, isolated zero, and the zerotree root. They are coded with an arithmetic coder. Both the subjective quality and PSNR are greatly improved.

Several embedded wavelet coders followed the EZW and made more improvements in both the performance and coding complexity. The coefficient representation and prediction scheme were refined by the layer zero coding (*LZC*) technique (10). In LZC, the coefficients were simply represented by zero and one according to their significance, rather than the four-symbol representation of the EZW. The prediction of wavelet coefficients is implemented in the context choice of the adaptive arithmetic coder. The parent-child relationship and the zerotree structure were further exploited by set partitioning in the hierarchical tree (*SPHIT*) algorithm (11), which identified more special classes of tree structures of bits in the significant trees. The multithreshold wavelet coding (*MTWC*) technique (12) uses multiple quantization thresholds for each subband for better bit allocation and rearranges the transmission order of wavelet coefficients to achieve better performance. The latter two have low computational complexity and can be implemented for real-time image compression.

Vector Quantization. Different from scalar quantization, vector quantization (*VQ*) uses a quantization index (codeword) to represent a *vector* to be quantized. Therefore VQ reduces the redundancy if the vectors are closely correlated. VQ is applied to image coding in two ways. One is to use VQ as the replacement of the scalar quantizer in transform coding schemes, and the other is to treat clusters of image pixels as the vectors and perform the VQ.

The first method is useful if the transform coefficients are correlated in some way. For DCT coding, the correlation among DCT coefficients in the same block or across adjacent blocks is not very strong so that VQ cannot improve the performance too much. For wavelet coding, the coefficients are more closely correlated among nearby coefficients in the same subband or among parent-child subbands. Thus using VQ to quantize the wavelet coefficients can improve the performance to a larger extent.

In the second method, the correlation among adjacent pixels in the spatial domain is exploited. To perform VQ, a fixed block size is chosen. The pixel values from the block are chosen as the vector. All of the vectors from several "typical" images are collected as the training vectors, and a training algorithm is chosen. These vectors are trained to form a codebook with a number of representative vectors, called codewords. When compressing an image, every block from the image is extracted as a vector and the nearest codeword from the codebook is found. The index of the codeword is transmitted and the corresponding codeword is used as the reconstruction vector.

The disadvantage of VQ is that the training time required may be too long to form a codebook. Considering the training cost, the proper block size may be about 4×4. In addition, the performance depends on which images are used in the training set and which image is to be compressed. If a universal codebook is used, the performance is not optimal. The performance of traditional VQ is usually a lot worse than the transform coders. The advantage of the VQ coder is that once the training is completed, the encoding speed is quite fast if the size of the codebook is not too large. The decoding speed is extremely fast because no real computation is needed. There are several variants of VQ that improve the performance and reduce the computational complexity (see Data compression, lossy).

Fractal Compression. Fractal compression schemes (13) exploit the spatial redundancy by utilizing the self-similarity in the same image. Given a target region in an image, there could be another region similar to this region with different rotation, scale, and contrast. If we could find this approximation, we could encode the transformation (rotation, scale, contrast, and the displacement) from the target region. This could be very efficient because only a small amount of information needs to be encoded. The coding and decoding of the image is based on the partitioned iterated function system (*PIFS*). The classical way of separating the image into regions is to partition it into fixed-size blocks and find a similar block with some transformations. The transformation information is coded and transmitted. At the decoder end, an initial image (it could even be a random image!) is chosen, and the transformation is applied iteratively to each corresponding block. According to its mathematical theory, the image will converge to the original image after iterations.

The advantage of fractal compression is that it is good for low bit-rate compression because most of the information is included in the image itself and only a small amount of transformation information is needed for encoding a large block. It also alleviates the blocking artifacts in other block-based coding schemes. Another

advantage is that it can be used to enhance the resolution of images even beyond the original resolutions of the images because the iterative process can be extended to subpixel levels. The disadvantage is that the encoding may be too time-consuming in finding a similar block. Rather than pixel-wise matching to find a matched block, fractal compression has to perform all possible transformations to a block in the searching window to find the best match. The decoding time, on the other hand, is relatively fast without too much computation involved. An advantage of the iterative process is that we can cut off the decoding process at an arbitrary number of iterations. But, of course, the result may not be good if too few iterations are performed.

BIBLIOGRAPHY

1. ITU-R Recommendation BT.601, *Encoding parameters of digital television for studios*, 1982.
2. ISO/IEC JTC1 10918-1, *Information technology—digital compression and coding of continuous-tone still images: requirements and guidelines*, 1994.
3. ISO/IEC JTC1/SC29/WG1 N390R, *New work item: JPEG 2000 image coding system*, 1997.
4. ISO/IEC-11172, *Information technology—Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbps*, 1992.
5. ISO/IEC-13818—ITU-T Rec. H.262: *Information technology: Generic coding of moving pictures and associated audio*, 1996.
6. ISO/IEC JTC1/SC29/WG11 N1730, *Overview of the MPEG-4 standard*, 1997.
7. ISO/IEC JTC1/SC29/WG11 N1920, *MPEG-7: Context and objectives*, 1997.
8. ITU-T Recommendation H.263, *Video coding for low bit-rate communication*, 1995.
9. J. Shapiro Embedded image coding using zerotrees of wavelet coefficients, *IEEE Trans. Signal Process.*, **41**: 3445–3462, 1993.
10. D. Taubman A. Zakhor Multirate 3-D subband coding of video, *IEEE Trans. Image Process.*, **3**: 572–588, 1994.
11. A. Said W. A. Pearlman A new, fast, and efficient image codec based on set partitioning in hierarchical trees, *IEEE Trans. Circuits Syst. Video Technol.*, **6**: 243–250, 1996.
12. H.-J. Wang C.-C. J. Kuo A multi-threshold wavelet coder (MTWC) for high fidelity image, *IEEE Signal Process. Soc., 1997 Int. Conf. Image Process.*, 1997.
13. Y. Fisher, ed. *Fractal Image Compression: Theory and Applications*, New York: Springer-Verlag, 1995.

YUNG-KAI LAI
C.-C. JAY KUO
University of Southern California