Multimedia Systems (1994) 1:187–204

Multimedia Systems

[Stei94c] *Ralf Steinmetz;* **Data Compression in Multimedia Computing: Standards and Systems**; acm/Springer Zeitschrift "Multimedia Systems", 1994, Band 1, Nr. 5, S. 187-204, März 1994.

Data compression in multimedia computing – standards and systems

Ralf Steinmetz

IBM European Networking Center, Vangerowstrasse 18, D-69115 Heidelberg, Germany

5 JPEG

Since June 1982 Work Group 8 (WG8) of the ISO has been working on standardization of compression and decompression of still images (Hudson et al. 1988). In June 1987, ten different techniques for still color and gray-scaled images were presented. These proposals were compared and three of them were analyzed further. An adaptive transform coding technique based on the DCT achieved the best (subjective) results and therefore was adopted for JPEG (Leger et al. 1988; Wallace et al. 1988). JPEG (Joint Photographic Experts Group) is a joint project of ISO/IEC JTC1/SC2/WG10 and the commission Q.16 of CCITT SGVIII. In 1992 JPEG became an ISO International Standard (IS) (JPEG 1993).

JPEG applies to color and gray-scaled still images (Leger et al. 1991; Mitchell and Pennebaker 1991; Wallace 1991). A fast coding and decoding of still images is also used for video sequences known as Motion JPEG. Today, parts of JPEG are already available in software-only packages, or in packages including specific hardware support. It should be taken into consideration that in most cases only the very basic JPEG algorithms with a limited spatial resolution are supported by these products.

In addition to the requirements described in the respective section of this paper, this standard fulfills the following requirements in order to guarantee a further distribution and application of JPEG (Wallace 1991).

- The JPEG implementation should be independent of image size.
- It should be applicable to any image and pixel aspect ratio.
- The color representation itself should be independent of the special implementation.
- The image content may be of any complexity and with any statistical characteristics.
- It should be (or be near) the state of the art regarding the compression factor and the image quality achieved.
- The processing complexity must permit a software solution to run on as many as possible available standard processors. Additionally, the use of specialized hardware should substantially enhance the quality.
- Sequential decoding (line by line) and progressive decoding (refining the whole image) should be possible. A lossless, hierarchical coding of the same image with different resolutions similar to the photo CD images should be supported.

The user can select the quality of the reproduced image, the compression processing time, and the size of the compressed image by an individual choice of the appropriate parameters.

Applications do not have to include both an encoder and a decoder. In many applications only one of them is needed. The encoded data stream has a fixed interchange format that includes the encoded image data as well as the chosen parameters and the tables of the coding process. If the compression and decompression processes agree on a common set of, e.g., coding tables to be used, then they need not be included in the data stream. If there is this common context between coding and decoding, the interchange format can have an *abbreviated format*. This format does not guarantee inclusion of the necessary tables; however, some may be provided [see Appendix B of (JPEG 1993)]. The interchange format in the regular mode (i.e., the nonabbreviated format) includes all of the information necessary for decoding without any previous knowledge of the coding process.

Figure 3 outlines the steps of the JPEG compression for the overall scheme shown in Fig. 1. Four different combinations can be determined that lead to *4 modes*. Each mode includes further variations in itself:

- The *lossy sequential DCT based mode* (baseline process) must be supported by every JPEG implementation.
- The *expanded lossy DCT based mode* provides a set of further enhancements to the baseline mode.
- The *lossless mode* has a low compression ratio that allows perfect reconstruction of the original image.
- The *hierarchical mode* comprises images of different resolutions and selects its algorithms from these three modes.

According to Fig. 3 the *baseline process* comprises the following techniques: Block, MCU, FDCT, runlength, and Huffman, which are explained in more detail in this section as are the other modes. In the next section the image preparation for all modes is presented; later the remaining steps, image processing, quantization, and entropy encoding are described.

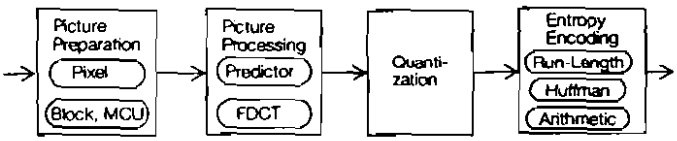


Fig. 3. Steps of the JPEG compression process taking into account the different JPEG modes

5.1 Image preparation

For the first step of image preparation as shown in Fig. 3, JPEG introduces a very general image model. In this model it is possible to describe most of the well-known two-dimensional image representations. For instance, the model is not based on three image components with 9-bit YUV coding and a fixed number of lines and columns. Mapping of encoded chrominance values is not coded either. This fulfills the demanded independence of image parameters, like image size, image and pixel aspect ratio.

A source image consists of at least one and at most of 255 components or planes, as shown on the left side of Fig. 4. Each component Ci may have a different number of pixels in the horizontal (X_i) and vertical (Y_i) axis. Note that the index denotes the number of the component or plane. These components may be assigned to the three colors RGB (red, green, blue), YIQ (Y denotes the luminance component; the chrominance is amplitude-modulated onto the color subcarrier at two phases: I for in-phase at 0 degrees and Q for quadrature at 90 degress) or YUV signals, for example.

Figure 5 shows three components of an image, each with the same resolution, each having a rectangular array C_i of $X_i \times Y_i$ pixels. The three X_i values and all three Y_i values are the same.

The resolution of the individual components may be different. Figure 6 shows an example of an image with half of the number of columns (i.e., half number of horizontal samples) in the second and third plane compared to the first component: $Y_1 = Y_2 = Y_3$, and $X_1 = 2X_2 = 2X_3$.

A gray-scale image will, in most cases, consist of a single component. An RGB color representation has three components with equal resolution (same number of lines $Y_1 = Y_2 = Y_3$, and same number of columns $X_1 = X_2 = X_3$). An example of YUV color images is used by DVI, with subsampling of the chrominance components. For JPEG image processing in DVI there are three components with $Y_1 = 4Y_2 = 4Y_3$ and $X_1 = 4X_2 = 4X_3$.

Each pixel is represented by p bits with values in the range of 0 to 2^{p-1}. All pixels of all components within the same image are coded with the same number of bits. The lossy modes of JPEG use a precision of either 8 or 12 bits per pixel. The lossless modes are defined from 2 to 12 bits per pixel. If a JPEG application wants to make use of any other number of bits, the application itself has to perform a suitable transformation of the image to the well defined numbers of bits in the JPEG standard.

The dimensions of a compressed image are defined by new values X (maximum of all X_i), Y (maximum of all Y_i), H_i and V_i. H_i and V_i are the relative horizontal and vertical sampling ratios specified for each component i. H_i and V_i must be integer values in the range between 1 and 4. This awkward looking definition is needed for an interleaving of the components that is described later.

Consider the following example as shown in (JPEG 1993). A picture is given with the maximum horizontal and vertical resolution of 512 pixels and three components (X = 512 and Y = 512). The following sampling factors are given:

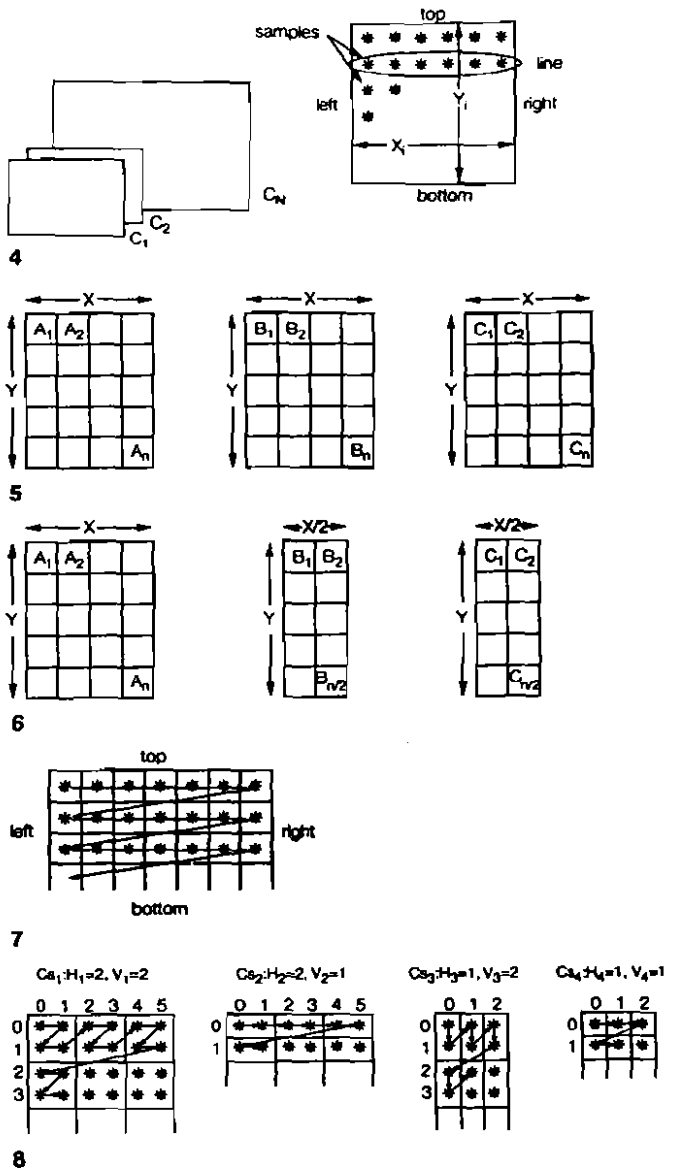


Fig. 4. Digital uncompressed still image with the definition of the respective image components according to the JPEG standard
Fig. 5. Example of JPEG image preparation with three components having the same resolution
Fig. 6. Example of JPEG image preparation with three components having different resolutions
Fig. 7. Non-interleaved order of data units, the processing of one component according to the JPEG standard
Fig. 8. Interleaved data units, an example with 4 components as derived from the JPEG standard

level 0 : $H_0 = 4$, $V_0 = 1$
level 1 : $H_1 = 2$, $V_1 = 2$
level 2 : $H_2 = 1$, $V_2 = 1$

This leads with X = 512, Y = 512, $H_{max} = 4$ and $V_{max} = 2$ to

level 0 : $X_0 = 512$, $Y_0 = 256$
level 1 : $X_1 = 256$, $Y_1 = 512$
level 2 : $X_2 = 128$, $Y_2 = 256$

because with the ceiling function X_i and Y_i are calculated according to the following formula:

$$X_i = \left\lceil X \times \frac{H_i}{H_{max}} \right\rceil$$

$$Y_i = \left\lceil Y \times \frac{V_i}{V_{max}} \right\rceil .$$

For the use of compression the image is divided in data units. The lossless mode uses one pixel as one data unit. The lossy mode uses blocks of 8 × 8 pixels. This definition of data units is a result of the DCT that always transforms connected blocks.

In most cases the data units are processed component by component, and passed, according to Fig. 3, in this generated order, to the image processing step for further processing. As shown in Fig. 7, for one component, the order of processing the data units is left-to-right and top-to-bottom, one component after the other; this is known as *noninterleaved data ordering*. Using this noninterleaved mode for a RGB encoded image with very high resolution, the display would initially present only the red component, then in turn the blue and green would be drawn, resulting in the original image colors being reconstructed. Therefore due to the finite processing speed of the JPEG decoder it is often more suitable to interleave the data units as shown in Fig. 8.

Interleaved data units of different components are combined into minimum coded units (MCUs). If all the components have the same resolution ($X_i \times Y_i$), then a MCU consists of exactly one data unit of each component. The decoder displays the image MCU by MCU; this allows a correct color presentation, even for partly decoded images. In the case of different resolutions for the single components the construction of the MCUs becomes more complex (Fig. 8). For each component the *regions* of the data units (if necessary, with different numbers of data units) are determined. Each component consists of the same number of regions, for example, Fig. 8 shows six regions for each component. An MCU consists of exactly one region in each component. Again the data units within one region are ordered left-to-right and top-to-bottom.

Figure 8 shows an example with four components; the values of H_i and V_i for each component are provided in the figure. The first component has the highest resolution in both dimensions, and the fourth component has the lowest resolution. The arrows indicate the sampling direction of the data units of each component. The MCUs are built in the following order:

$MCU_1 = d^1_{00}\, d^1_{01}\, d^1_{10}\, d^1_{11}\, d^2_{00} d^2_{01}\, d^3_{00}\, d^3_{10}\, d^4_{00}$,
$MCU_2 = d^1_{02}\, d^1_{03}\, d^1_{12}\, d^1_{13}\, d^2_{02}\, d^2_{03} d^3_{01}\, d^3_{11}\, d^4_{01}$,
$MCU_3 = d^1_{04}\, d^1_{05}\, d^1_{14}\, d^1_{15}\, d^2_{04}\, d^2_{05} d^3_{02}\, d^3_{12}\, d^4_{02}$,
$MCU_n = d^1_{20}\, d^1_{21}\, d^1_{30}\, d^1_{31}\, d^2_{10}\, d^2_{11} d^3_{20}\, d^3_{30}\, d^4_{10}$.

The data units of the first component are

$Cs_1 : d^1_{00} \ldots d^1_{31}$.

The data units of the second component are

$Cs2 : d^2_{00} \ldots d^2_{11}$.

The data units of the third component are

$Cs_3 : d^3_{00} \ldots d^3_{30}$.

The data units of the fourth component are

$Cs_4 : d^4_{00} \ldots d^4_{10}$.

According to JPEG, up to four components can be encoded using the interleaved mode. Each MCU consists of at most 10 data units, and within an image some components can be encoded in the interleaved mode and others in the noninterleaved mode.

5.2 Lossy sequential DCT based mode

After image preparation, the uncompressed image samples are grouped into data units of 8 × 8 pixels and passed to the encoder; the order of these data units is defined by the MCUs. In this baseline mode, single samples are encoded using p = 8 bits. Each pixel is an integer in the range of 0 to 255.

The first step of the *image processing* in the baseline mode (baseline process) as shown in Fig. 9 is a DCT (Ahmed et al. 1974; Narasinka and Peterson 1978). The pixel values are shifted into the range −128–127 with zero as the center. These data units with 8 × 8 shifted pixel values are defined by S_{yx}, where x and y are in the range of 0–7. Then each of these values is transformed by the forward DCT (FDCT):

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

where C_u , $C_v = \frac{1}{\sqrt{2}}$ for u, v = 0 else C_u , $C_v = 1$.

Altogether this transform must be carried out 64 times per data unit. The result is 64 coefficients S_{vu}. The DCT is similar to the discrete Fourier transform (DFT); it maps the values from the time to the frequency domain, therefore each coefficient can be regarded as a two-dimensional frequency.

The coefficient S_{00} corresponds to the lowest frequency in both dimensions. It is known as the DC coefficient, which determines the fundamental color of the data unit of 64 pixels. The DC coefficient is the DCT coefficient for which the

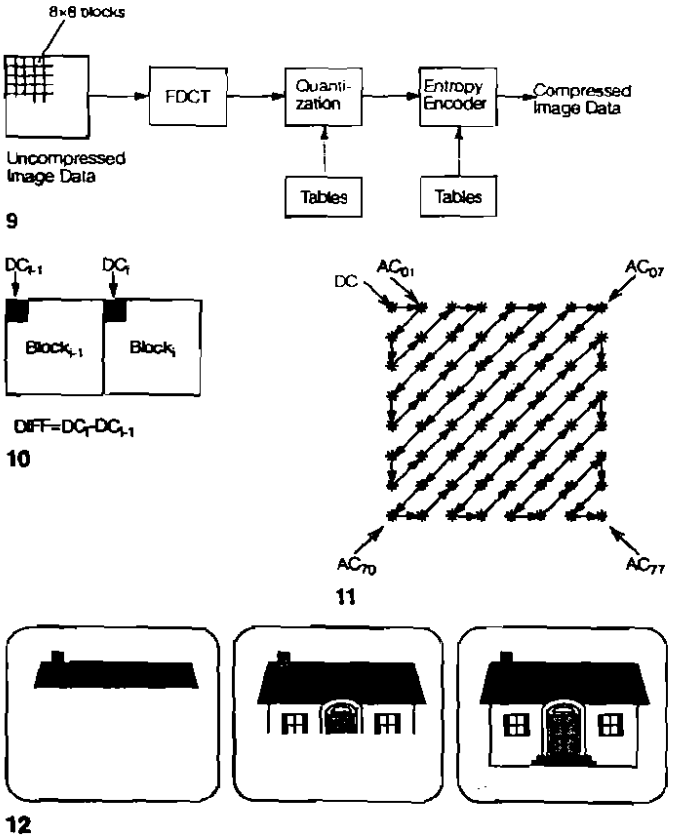


Fig. 9. Steps of the lossy sequential DCT-based coding mode
Fig. 10. Preparation of DCT DC-coefficients for entropy encoding, calculation of the difference between neighboring values
Fig. 11. Preparation of the DCT AC-coefficients for entropy encoding: Order with increasing frequencies
Fig. 12. Sequential picture presentation used, e.g., in the lossy DCT-based mode

frequency is zero in both dimensions. The other coefficients are called AC coefficients. The AC coefficients are all DCT coefficients for which the frequency in one or both dimensions is not zero. For instance, S_{70} represents the highest frequency that occurs in the horizontal direction, which is the closest separation of vertical lines that is possible in the 8×8 data unit. S_{07} represents the highest frequency in the vertical dimension, i.e., the closest separation of horizontal lines. S_{77} indicates the highest frequency appearing equally in both dimensions. The absolute value of S_{77} is greatest if the source 8×8 data unit consists of a full matrix, i.e., with as many 1×1 components as possible. One or both dimensions are not zero. Accordingly, for example, S_{44} will be greatest if the block consists of 16 squares of 4×4 pixels. Taking a closer look at the formula, we recognize that the cosine expressions only depend upon x and u, y and v respectively; but they do not depend upon s_{yx}. Therefore these cosine expressions represent constants that do not have to be calculated over and over again. There are many effective techniques and implementations of the DCT. Important contributions can be found in Duhamel and Guillemot (1990), Feig (1990), Hou (1988), Lee (1984), Linzer and Feig (1991), Suhiro and Hatori (1986), Vetterli (1985), and Vetterli and Nussbaumer (1985).

For reconstructing the image, the decoder uses the inverse DCT (IDCT). The coefficients S_{vu} must be used for the calculation:

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

$$\text{where } C_u, C_v = \frac{1}{\sqrt{2}} \text{ for } u, v = 0 \text{ ; else } C_u, C_v = 1$$

If the FDCT, as well as the IDCT, could be calculated with full precision, it would be possible to reproduce the 64 source pixels exactly. From a theoretical point of view the DCT would be lossless in this case. In practice the precision is restricted, and the DCT is lossy; however, the JPEG standard does not define any precision. This is the reason why two different implementations of a JPEG decoder could generate different images as output of the same compressed image. JPEG merely defines the maximum tolerance.

Most of the areas of a typical image consist of large regions of a single color which, after applying the DCT, are represented by many coefficients with very low values. The edges, however, are transformed into coefficients that represent high frequencies. Images of average complexity consist of many AC coefficients with a value of almost zero. Therefore entropy encoding is used in order to achieve a considerable data reduction.

Following the steps of Fig. 3 all the DCT coefficients are *quantized*. This is a lossy transform. For this step, the JPEG application provides a table with 64 entries. Each entry is used for the quantization of one of the 64 DCT coefficients. Thereby each of the 64 coefficients can be adjusted separately. The application has the ability to affect the relative significance of the different coefficients and specific frequencies can be given more importance than others. These coefficients should be determined in relation to the characteristics of the source images. The possible compression is influenced at the expense of the achievable image quality.

Each table entry is an 8-bit integer value Q_{vu}. The quantization is defined by:

$$Sq_{vu} = \text{round}\left(\frac{S_{vu}}{Q_{vu}}\right) .$$

The quantization becomes less accurate the larger the table entries are. The *dequantization* at the decoder before the application of the IDCT is defined as:

$$R_{vu} = Sq_{vu} \times Q_{vu} .$$

Quantization and dequantization must use the same tables. No default values for quantization tables are specified in JPEG; applications may specify values that customize the desired picture quality according to the particular image characteristics.

At the initial step of *entropy encoding*, the quantized DC coefficients are treated separately from the quantized AC coefficients. The order of processing of the whole set of coefficients is given by the zigzag sequence as shown in Fig. 11.

- The *DC coefficients* determine the basic color of the data units. Between adjacent data units the variation of color is fairly small. Therefore a DC coefficient is encoded as the difference between the current DC coefficient and the previous one. Only the differences are processed subsequently (Fig. 10).
- The order of DCT processing of *AC coefficients* using the zigzag sequence illustrates that coefficients with lower frequencies (typically with higher values) are encoded first, followed by the encoding of higher frequencies (with typically small, almost zero, values). The result is an extended sequence of similar data bytes permitting very efficient entropy encoding. Note that the arrow between the DC coefficient and the first AC coefficient just denotes that this DC value has the lowest frequency.

JPEG specifies Huffman and arithmetic encoding as entropy encoding methods. For the lossy sequential DCT-based mode discussed in this section only the Huffman encoding may be used. In both methods a run length encoding of zero values of the quantized AC coefficients is applied first. Additionally, nonzero AC coefficients as well as the DC coefficients are transformed into a spectral representation in order to compress the data even more: the number of bits required depends on the value of the representation. A nonzero AC coefficient is be represented by 1 to 10 bits. For the representation of the DC coefficients, a higher resolution of 1 bit to a maximum of 11 bits is used. The result is a representation according to the *ISO intermediate symbol sequence* format that specifies the following information:

- The number of subsequent coefficients with the value zero
- The number of bits used for the representation of the coefficient that follows
- The value of the coefficient represented in the specified number of bits

The major advantage of the Huffman encoding over arithmetic coding is the free implementation, because it is not protected by any patent.

Disadvantageous is the fact that the application must provide encoding tables, since JPEG does not predefine any of them. This baseline mode allows the use of different Huffman tables for AC and DC coefficients.

In the case of the described sequential encoding, the whole image is coded and decoded in a single run. Figure 12 shows an example of decoding with immediate presentation; the picture is presented from top-to-bottom.

5.3 Expanded lossy DCT based mode

Image preprocessing in this mode differs from the previously described mode in terms of the number of bits per sample. A sample precision of 12 bits per sample as well as 8 bits per sample can be used. The image processing is DCT based and follows analogous rules to the baseline DCT mode.

With the expanded lossy DCT based mode, JPEG defines *progressive encoding* in addition to sequential encoding. In the first run a very rough representation of the image appears that looks out of focus and that is refined during succeeding steps. A schematic example is shown in Fig. 13.

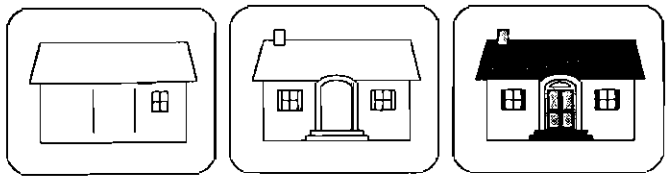


Fig. 13. Progressive picture presentation used, e.g., in the expanded lossy DCT based mode

Table 2. Types of image processing in the extended lossy DCT based mode

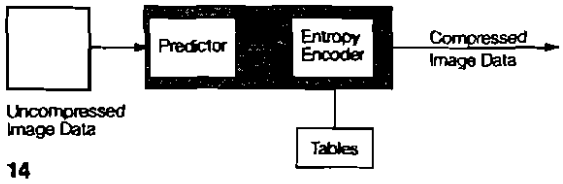
Image display	Bits per sample	Entropy coding
Sequential	8	Huffman coding
Sequential	8	Arithmetic coding
Sequential	12	Huffman coding
Sequential	12	Arithmetic coding
Progressive successive	8	Huffman coding
Progressive spectral	8	Huffman coding
Progressive successive	8	Arithmetic coding
Progressive spectral	8	Arithmetic coding
Progressive successive	12	Huffman coding
Progressive spectral	12	Huffman coding
Progressive successive	12	Arithmetic coding
Progressive spectral	12	Arithmetic coding

Progressive image representation is achieved by an expansion of quantization. This is also known as *layered coding*. For this expansion a buffer is added at the output of the quantizer that temporarily stores all coefficients of the quantized DCT. Progressiveness is achieved in two different ways:

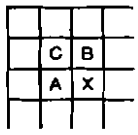
- By using *spectral selection*. In the first run only the quantized DCT coefficients of low frequencies of each data unit are passed to the entropy encoding. In the succeeding runs the coefficients of higher frequencies are processed.
- *Successive approximation* transfers all of the quantized coefficients in each run, but single bits are differentiated according to their significance. The most significant bits are encoded first and then the less significant bits.

Besides Huffman encoding, arithmetic entropy encoding can be used in this mode. The arithmetic encoding requires no tables for the application as it is automatically adapted to the statistical characteristics of an image. Several publications state that the compression achieved by arithmetic coding is sometimes between 5% and 10% better than by Huffman encoding. Other authors assume a similar compression rate. Arithmetic coding is slightly more complex and its protection by patents must be considered [see Appendix L of JPEG (1993)].

Four coding tables for the transformation of DC and AC coefficients can be defined by the JPEG application. In a simpler mode a choice of only two Huffman tables each for the DC and AC coefficients of one image is allowed. For this reason twelve alternative types of processing can be used in this mode



14



15

Fig. 14. Lossless mode which is based on a prediction
Fig. 15. Principle of the prediction in the lossless mode

Table 3. Predictors for lossless coding

Selection value	Prediction
0	No prediction
1	X = A
2	X = B
3	X = C
4	A + B − C
5	X = A + (B − C)/2
6	X = B + (A − C)/2
7	X = (A + B)/2

(Table 2). The most commonly used is the sequential display mode with 8 bits per sample and the Huffman encoding.

5.4 Lossless mode

The lossless mode shown in Fig. 14 uses data units of single pixels for *image preparation*. Any precision between 2 and 16 bits per pixel can be used.

In this mode the image processing and quantization use a predictive technique instead of a transformation encoding technique. As shown in Fig. 15, for each pixel X one of eight possible predictors is selected. The selection criterion is the best possible prediction of the value of X from the already known adjacent samples A, B, and C. The specified predictors are listed in Table 3. The number of the chosen predictor as well as the difference of the prediction to the actual value are passed to the subsequent entropy encoding. Entropy encoding can use either the Huffman or the arithmetic encoding technique.

In summary, this mode allows one to choose from eight different modes of processing each using between two and sixteen bits per pixel. Each of the variations can be combined with either Huffman or arithmetic encoding.

5.5 Hierarchical mode

The hierarchical mode uses either any of the lossy DCT based algorithms already described or, alternatively, the lossless compression technique. The main feature of this mode is the encoding of an image at different resolutions, i.e., the encoded data contains images at several resolutions. The prepared image is initially sampled at a lower resolution (reduced by the factor 2^n). Subsequently the resolution is enhanced by a factor of 2 vertically and horizontally. This compressed image is then subtracted from the previous result. The process is repeated until the full resolution of the image is encoded.

Hierarchical encoding requires considerably more storage capacity, but the encoded image is immediately available at different resolutions. Therefore applications working with lower resolutions do not have to decode the whole image and subsequently apply image processing algorithms to reduce the resolution. Scaling becomes cheap. In the author's experience with scaled images in the context of DVI, any scaling performed by the application consumes considerable time. It takes a CPU less time to display an image with full resolution than to process a scaled-down image and display it with a reduced number of pixels. Image coding based on the JPEG hierarchical mode causes the display of a reduced-size picture to consume less processing power than at any higher resolution.

6 H.261 (px64)

The driving force behind the H.261 (px64) video coding standard is ISDN. The two B channels of an ISDN connection (or part of them) can be used to transfer video in addition to audio data. This implies that both users that are connected via the B channel must use the same codec for video signals. Note that "codec" means coder and decoder, i.e., encoding and decoding, compression and decompression. In the case of an ISDN connection, exactly two B channels and one D channel are available at the user interface. The European ISDN hierarchy allows a connection with, e.g., 30 B channels that were originally intended for the private automatic branch exchange (PABX). Here, we use "B channels" to mean one or more ISDN channels. The prime ISDN applications considered were videophone and video conferencing systems. For these dialogue applications coding and decoding must be carried out in real time. In 1984 study group XV of the CCITT established a committee that worked on this standard for the compression of moving pictures (Liou 1991).

First, a compressed data stream with a data rate of m × 384 kbits/s at m = 1, 2, ..., 5 was forseen, later a demand for standardization with n × 64 kbits/s at n = 1, 2, ..., 5 arose. Due to advances in video coding technology and the necessary support of narrow band ISDN, a decision in favor of video compression with a data rate of p × 64 kbits/s at p = 1, 2, ..., 30 was taken. Five years later the CCITT Recommendation H.261 "Video Codec for Audiovisual Services at p × 64 kbits/s" (CCITT 1990) was finalized in December 1990. This recommendation is also known as *px64*, because of the compressed data rate of p × 64 kbits/s. North America adopted the recommendation in a slightly modified way.

The CCITT recommendation H.261 was developed for the real-time process of encoding and decoding. The maximum signal delay of both compression and decompression must

not exceed 150 ms. If the end-to-end delay is too long, an application using this technology will be affected considerably.

6.1 Image preparation

Unlike JPEG, H.261 defines a very precise image format. The image refresh frequency at the input must be 29.97 frames/s. During encoding it is possible to generate a compressed image sequence with a lower frame rate of 10 or 15 still images/s. Only noninterleaved images are allowed at the input of the coder. The image is encoded as luminance signal (Y) and chrominance difference signals Cb, Cr according to the CCIR 601 subsampling scheme (2:1:1). Later this was also adopted by MPEG.

Two resolution formats each with an aspect ratio of 4:3 are specified, the so-called common intermediate format (CIF) defines a luminance component of 288 lines, each with 352 pixels. The chrominance components have a resolution with a rate of 144 lines and 176 pixels per line to fulfil the 2:1:1 requirement. Quarter CIF (QCIF) has exactly half the CIF resolution, i.e., 176 × 144 pixels for the luminance and 88 × 72 pixels for the other components. All H.261 implementations must be able to encode and decode QCIF; CIF is optional.

The necessary compression ratio for images with the low resolution of QCIF (determined by the bandwidth of an ISDN B channel) is illustrated by means of the following example. The uncompressed QCIF is composed of a data stream with 29.97 frames/s with a data rate of about 9.115 Mbits/s; for CIF (with the same number of images/s) an uncompressed data rate of about 36.45 Mbits/s is produced. The image should be reduced to a frame rate of 10 pictures/s. This leads to a compression ratio of about 1:47.5, which can be supported with today's technology. Using a CIF format with the same compression ratio, a reduction to approximately the bandwidth of 6 ISDN B channels is possible.

In H.261 data units of the size of 8 × 8 pixels are used for the representation of the Y as well as of the C_b and C_r components. A macro block is the result of combining four blocks of the Y matrix each with one block of the C_b and the C_r component. A *group of blocks* is defined as 33 macro blocks. Therefore a QCIF image consists of three groups of blocks, and a CIF image comprises twelve groups of blocks.

6.2 Coding algorithms

The H.261 standard uses two different ways of coding: *intraframe* and *interframe*. In the case of intraframe coding no advantage is taken from the redundancy between frames. This coding technique corresponds to the I frame coding of MPEG (Sect. 7.1). For interframe coding, information from previous or subsequent frames is used; this corresponds to the P frame encoding of MPEG (Sect. 7.1). The standard does not provide any criteria for the choice of mode. The decision must be made during the coding process. It depends on the specific implementation.

Similarly to JPEG, for intraframe encoding, each block of 8 × 8 pixels is transformed into 64 coefficients by a DCT. The quantization of the DC coefficients differs from the quantization of the AC coefficients. The next step is to apply entropy encoding to the AC and DC parameters, resulting in a variable length encoded word. *Interframe coding* is based on a prediction for each macro block of an image. This is determined by a "comparison" of macro blocks from previous images with the current image. The motion vector is defined by the relative position of the previous macro block with respect to the current macro block. Note that according to H.261, the coder need not be able to determine a motion vector. Therefore a simple H.261 implementation considers only the differences between macro blocks located at the same position of consecutive images. In such cases the motion vector is always a zero vector.

Subsequently the motion vector and the DPCM coded macro block are processed. The DPCM coded macro block is transformed by a DCT if and only if its value exceeds a certain threshold. If the difference is less than this threshold, the corresponding macro block is not encoded any further; only the relevant motion vector is processed. The components of the motion vector are entropy encoded with a variable length coding system that is lossless. All of the transformed coefficients are quantized linearly and variable length encoded.

Additionally, an optical low-pass filter can operate between the DCT and entropy encoding process. This filter deletes any remaining high-frequency noise. Note that such a filter is optional, and few implementations actually incorporate it.

For H.261 the quantization is a linear function and the step size is dependent on the amount of data in the transform buffer. This mechanism enforces a constant data rate at the output of the coder. Therefore the quality of the encoded video data depends on the contents of individual images as well as on the motion within the respective video scene.

6.3 Data stream

According to H.261, a data stream has a hierarchical structure composed of several layers, the bottom layer containing the compressed picture. H.261 has the following characteristics; for further details see CCITT (1990):

- The data stream of an image includes information for error correction.
- For each image a 5-bit image number is used as a temporal reference.
- If a certain command is passed from the application to the decoder, the image displayed last is "frozen" as a still image. This allows the application at the decoding station to stop/freeze and start/play a video scene without any additional effort.
- Using further commands sent by the encoder (and not by the application), it is also possible to switch between still image node and moving image mode. Alternatively a timeout signal can also be used instead of this explicit command.

H.261 was designed for conferencing systems and video telephony. Most of today's implementations can be found in this scope.

7 MPEG

MPEG has been developed by ISO/IEC JTC1/SC 29/WG11. It covers motion video as well as audio coding according to the ISO/IEC standardization process. Considering the state of the art in the digital mass storage of CD technology, MPEG is striving for a compression of the data stream to a rate of about 1.2 Mbits/s, which is today's typical CDROM data transfer rate. MPEG can deliver a data rate of 1,856,000 bit/s at most, which "should" not be exceeded (MPEG 1993a). Audio data leads to a rate between 32 and 448 kbits/s; this data rate enables video and audio compression of acceptable quality. In 1993 MPEG was issued as an IS (MPEG 1993a). In 1993 the first commercially available MPEG products appeared on the market.

MPEG explicitly considers the activities of other standardization organizations:

- JPEG: a video sequence can be regarded as a sequence of still images. Furthermore, the JPEG development was always ahead of the MPEG standardization. Therefore MPEG activities make use of JPEG.
- H.261: As the H.261 standard was already available during the work on MPEG, the working group strived for compatibility (at least in some areas) with this standard. Implementations that are capable of H.261 as well as of MPEG may arise, however MPEG is the more advanced technique.

MPEG is suitable for both symmetric and asymmetric compression. Asymmetric compression requires more effort for coding than for decoding. Compression is carried out once, whereas the same data are decompressed many times. A typical application area is that of retrieval systems. Symmetric compression is known to require a similar effort for compression and decompression. Interactive dialogue applications make use of this encoding technique where a restricted end-to-end delay is required.

Besides the specification of video (Le Gall 1991; Viscito and Gonzales 1991) and audio coding, the MPEG standard provides a *system definition*. This definition describes the combination of several individual data streams.

7.1 Video encoding

In contrast to JPEG, but like H.261, the *image preparation* phase of MPEG, according to our reference scheme shown in Fig. 1, defines the format of an image exactly. Each image consists of three components (similar to the YUV format); the luminance component has twice as many samples in the horizontal and vertical axes as the other two components. This is known as *color subsampling*. The resolution of the luminance component should not exceed 768 × 576 pixels; for each component a pixel is coded with 8 bits.

The MPEG data stream includes more information than a data stream compressed according to the JPEG standard; for example, the aspect ratio of a pixel is included. MPEG provides 14 different image aspect ratios for a pixel. The most important are:

- A square pixel (1:1) is suitable for most computer graphics systems.
- With 702 × 575 pixels an aspect ratio of 4:3 is defined.
- With 711 × 487 pixels an aspect ratio of 4:3 is defined.
- For an image with 625 lines an aspect ratio of 16:9 is defined. This is the ratio required for HDTV.
- For an image with 525 lines an aspect ratio of 16:9 is defined. This is the ratio required for HDTV.

The image refresh frequency is also encoded in the data stream. Eight frequencies were defined: 23.976 Hz, 24 Hz, 25 Hz, 29.97 Hz, 30 Hz, 50 Hz, 59.94 Hz, and 60 Hz.

A temporal prediction of still images leads to a considerable compression ratio. Moving images often contain non-translational moving patterns, such as rotations or waves at the seaside. Areas in an image with these irregular patterns of strong motion can only be reduced by a ratio similar to that of intraframe encoding. The use of temporal predictors requires the storage of a great amount of information and image data. There is a need to balance this required storage capacity and the achievable compression rate. In most cases this predictive encoding only makes sense for parts of an image and not for the whole image. Therefore each image is divided into areas called macro blocks. Each macro block itself is partitioned into 16 × 16 pixels for the luminance component and 8 × 8 pixels for each of the two chrominance components. These turn out to be very suitable for a compression based on motion estimation. This is a compromise of costs for prediction and the resulting data reduction. A macro block consists of six blocks of 8 × 8 pixels each, four luminance blocks and two chrominance blocks.

Due to the required frame rate, each image must be built up within a maximum of 41.7 ms. From the user's perspective, a progressive image display has no advantages over a sequential display. The user has no need, and it is not possible, to define minimum coded units (MCUs) in MPEG (in contrast to JPEG).

MPEG distinguishes four types of coding of an image for *processing* according to Fig. 3. The reasons behind this are the contradictory demands for an efficient coding scheme and fast random access. To achieve a high compression ratio, temporal redundancies of subsequent pictures must be exploited (interframe), whereas the demand for fast random access requires intraframe coding. The following types of images are distinguished ("image" is used as a synonym for "still images" or "frame"):

- *I frames* (intra coded images) are self-contained, i.e., coded without any reference to other images; an I frame is treated as a still image. MPEG makes use of JPEG for I frames; however, contrary to JPEG, the compression must often be executed in real time. The compression rate of I frames is the lowest within MPEG. I frames are points for random access in MPEG streams.
- *P frames* (predictively coded frames) require information of the previous I and/or P frames for encoding and decoding, i.e., the data of the last I frame as well as from all P frames that were in between. The achievable compression ratio of P frames is considerably higher than the ratio for I frames

only. A P frame can be accessed after the decoding of the previous I frame and all other P frames between the previous I frame and the current P frame to be accessed.

- *B frames* (bidirectionally predictively coded frames) require information from the previous and following I and/or P frames for encoding and decoding. The highest compression ratio is attained by using these frames. A B frame is defined as the difference of a prediction of the past image and the following P or I frame. B frames can never be directly accessed in a random fashion.
- *D frames* (DC coded frames) are intraframe encoded. They can be used for fast forward or rewind mode. The DC parameters are DCT coded; the AC coefficients are neglected.

Figure 16 shows a sequence of I, P and B frames. As an example, the prediction for the first P frames and a bidirectional prediction for a B frame is shown. Note that due to the use of B frames, the order of the images in an MPEG coded data stream often differs from the actual decoding order. A P frame to be displayed after a related B frame must be decoded first because its data is required for the decompression of the B frame. This fact introduces an additional end-to-end delay.

The regularity of a sequence of I, P and B frames is determined by the MPEG application. For fast random access, the best resolution would be achieved by coding the whole data stream as I frames. On the other hand, the highest degree of compression is attained by using as many B frames as possible. For practical applications the following sequence has proved useful: IBBPBBPBB IBBPBBPBB .... In this case with 25 images/s random access would have a resolution of nine still images (i.e., about 360 ms), and it still provides a very good compression ratio. The following consolidated description of the *image processing, quantization, and entropy encoding* distinguishes the different types of images.

I frames use 8 × 8 blocks defined within a macro block, on which a DCT is performed. The DC coefficients are then DPCM coded. Differences of successive blocks of one component are computed and transformed with variable length coding. MPEG distinguishes two types of macro blocks. The first type includes only the encoded data, and the second covers a parameter used for scaling by adjustment of the quantization characteristics.

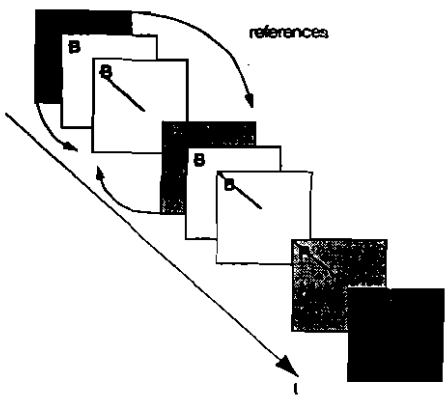


Fig. 16. Types of images in MPEG: I, B, and P frames

The coding of P frames is based on the fact that, in successive images, areas of these images often do not change at all, but instead, the whole area is shifted. In this case of *temporal redundancy*, the block of the last P or I frame that is most similar to the block under consideration is determined. Several methods for motion estimation are available to the encoder. The more processing-intensive methods tend to give better results. There is a trade-off to be made in the encoder: computational power, and hence cost, versus video quality (MPEG 1993a). Several matching criteria are available, e.g., the differences of all absolute values of the luminance component are computed. The minimal number of the sum of all differences indicates the best matching macro block. Thereby MPEG does not provide a certain algorithm for motion estimation, but instead specifies the coding of the result. Only the motion vector (the difference between the spatial location of the macro blocks) and the small difference of content of these macro blocks are left to be encoded. The search range (i.e., the difference between maximum size of the motion vector, and the location of the respective macroblock to be encoded) is not defined in the standard, but it is constrained by the definable motion vector range. The larger the search range, the better the motion estimation; however, the computation is slower.

Like I frames, P frames consist of macro blocks with encoded data only and six predictive macro blocks. The coder must determine if a macro block should be coded predictively or as a macro block of an I frame, and furthermore, if there is a motion vector that has to be encoded. A P frame can contain macro blocks that are encoded with the same technique as I frames. The coder for specific macro blocks of P frames must consider the differences of macro blocks as well as the motion vector. The difference of all six 8 × 8 pixel blocks of the best matching macro block and the macro block to be coded are transformed by a two-dimensional DCT. For further data rate reduction, blocks that only have DCT coefficients with all values being zero are not processed further. These are stored as 6-bit values that are added to the encoded data stream. Subsequently, the DC and the AC coefficients are encoded with the same technique. Note that this differs from JPEG and from the coding of macro blocks of I frames. In the next step a run length encoding and the determination of a code of variable length (similar to Huffman coding) is applied. The motion vectors of adjacent macro blocks often differ only slightly, so a DPCM encoding is used. The result is again transformed with the aid of a table leading to a variable length encoded word.

For the prediction of B frames, the previous as well as the following P or I frame are taken into account. The following example illustrates the advantages of a bidirectional prediction. In a video scene a ball moves from left to right in front of a static background. In the left area parts of the image appear, that in the former image were covered by the ball. A prediction of these areas can be derived from the following but not from the previous image. A macro block may be derived from the previous or the next macro blocks of P or I frames. Apart from a motion vector from the previous to the next image, a motion vector in the other direction can also be used. "Interpolative" motion compensation that uses both matching macro blocks is

allowed. In this case two motion vectors are encoded. The difference of the macro block to be encoded and the interpolated macro block is determined. Further quantization and entropy encoding are performed like P-frame-specific macro blocks. B frames must not be stored in the decoder as a reference for subsequent decoding of images.

D frames consist only of the lowest frequencies of an image. They only use one type of macro block, and only the DC coefficients are encoded. D frames are used for display in a fast-forward or fast-rewind mode. This could be also realized by a suitable order of I frames. For this purpose, these I frames must occur periodically in a data stream. Slow-rewind playback requires a huge storage capacity, therefore all images that were combined in a group must be decoded in the forward mode and stored. Afterwards, a rewind playback is possible. This is known as the *group of pictures* in MPEG.

Concerning quantization, it should be mentioned that the AC coefficients of B and P frames are usually large values, whereas those of I frames are smaller values, for which the MPEG quantization is adjusted. If the data rate increases over a certain threshold, the quantization enlarges the step size. In the opposite case the step size is reduced and the quantization is finer.

7.2 *Audio encoding*

The audio coding (Fig. 17) of MPEG uses the same sampling frequencies as compact disc digital audio (CD-DA) and digital audio tape (DAT). Apart from these (44.1 kHz and 48 kHz) 32 kHz is available, all at 16 bits.

Three different layers with different encoder and decoder complexity and performance are defined. An implementation of a higher layer must be able to decode the MPEG audio signals of lower layers (Musmann 1990).

Similar to the two-dimensional DCT for video, a transform into the frequency domain is applied for audio. The FFT is suitable for this coding, and the spectrum is split into 32 noninterleaved sub-bands; for each sub-band the amplitude of the audio signal is calculated. Also for each subband the noise level is determined simultaneously with the actual FFT by using a "psychoacoustic model". At a higher noise level the data are roughly quantized and at a lower noise level they are more finely quantized. The quantized spectral portions of layer one and two are PCM encoded, and those of layer three are Huffman encoded. The audio can be coded with a single channel, or two independent channels, or one stereo signal. In the definition of MPEG there are two different stereo modes; two channels that are processed either independently or as *joint stereo*. In the case of joint stereo, MPEG exploits the redundancy of both channels and achieves a higher compression ratio.

Each layer defines 14 fixed bit rates for the encoded audio data stream, which in MPEG are addressed by a bit rate index. The minimal value is always 32 kbits/s. These layers support different maximal bit rates: Layer 1 allows for a maximal bit rate of 448 kbits/s, layer 2 for 384 kbits/s and layer 3 for 320 kbits/s. A decoder is not required to support a variable bit rate at layer 1 and 2. In layer 3 a variable bit rate is specified by switching the bit rate index. For layer 2, not all combinations of bit rate and mode are allowed:

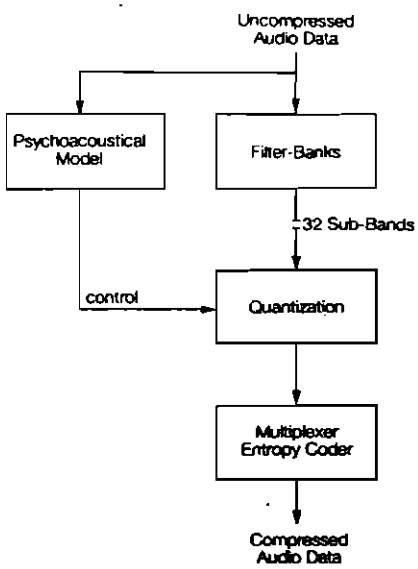


Fig. 17. Basic steps of audio encoding

- 32 kbits/s, 48 kbits/s, 56 kbits/s and 80 kbits/s are only allowed for a single channel.
- 64 kbits/s, 96 kbits/s, 112 kbits/s, 128 kbits/s, 160 kbits/s and 192 kbits/s are allowed for all modes.
- 224 kbits/s, 256 kbits/s, 320 kbits/s, 384 kbits/s are allowed for the modes "stereo", "joint stereo", and "dual channel."

7.3 *Data stream*

MPEG specifies a syntax for the interleaved audio and video data stream. An *audio data stream* consists of frames, which are divided into audio access units. Each audio access unit is composed of slots. At the lowest complexity (layer 1) a slot consists of four bytes. In any other layer it consists of one byte. A frame always consists of a fixed number of samples. Most important is the term "audio access unit," which describes the smallest possible audio sequence of compressed data that can be completely decoded, being independent of all other data. The audio access units of one frame lead to a playing time of 8 ms at 48 kHz, of 8,7 ms at 44,1 kHz, and 12 ms at 32 kHz. In the case of stereo signals, data from both channels are included in one frame.

A *video data stream* comprises six layers:

1. At the highest level, the *sequence layer*, the data buffering is handled. A data stream should have low requirements in terms of storage capacity. For this reason, at the beginning of the sequence layer there are the following two entries: the constant bit rate of a sequence and the storage capacity that is needed for decoding. In the processing scheme after the quantizer a *video buffer verifier* is inserted. The resulting data rate is used to verify the delay caused by decoding. The

video buffer verifier influences the quantizer and forms a kind of control loop. Several successive sequences could have a varying data rate. During the decoding of several consecutive sequences, there is no direct relationship between the end of one sequence and the beginning of the next. The basic parameters of the decoder are reset and the decoder is initialized at this time.

2. The *group of pictures layer* is the next layer. This layer consists of a minimum of one I frame, which is the first frame. Random access to this image is always possible. At this layer it is possible to distinguish the order of images in a data stream and during display. The first image of a data stream must always be an I frame. Therefore the decoder decodes and stores the reference frame first. In the display order a B frame can occur before an I frame.

display order												
type of frame	B	B	I	B	B	P	B	B	P	B	B	P
no. of frame	0	1	2	3	4	5	6	7	8	9	10	11

order during decoding, i.e., within the data stream

type of frame	I	B	B	P	B	B	P	B	B	P	B	B
no. of frame	2	0	1	5	3	4	8	6	7	11	9	10.

3. The *picture layer* contains a whole picture. The temporal reference is defined by an image number. This number is written underneath the corresponding image in the above shown example of (MPEG 1993a). Note that data fields defined in this layer are not yet used in MPEG. The decoder is not allowed to use these data fields, as they are designated for future extensions.

4. The following layer is the *slice layer*. Each slice consists of a number of macro blocks that may vary from one image to the next. Additionally the DCT quantization of all macro blocks of a slice is specified.

5. The fifth layer is the *macro block layer*. It contains the sum of the features of each macro block as previously described.

6. The lowest layer is the *block layer* already described.

MPEG defines the combination of data streams to a single data stream in the *system definition*. The same idea was pursued in DVI to define the Audio/Video Support System (AVSS) data format. The most important task of this process is the multiplexing. It includes the coordination of input data streams, output data streams, the adjustment of clocks, and the buffer management. Therefore the data stream defined by ISO 11172 is divided into single *packs*. The decoder gets the information necessary for its resource reservation from this multiplexed data stream. The maximal data rate is included in the first pack at the beginning of each *ISO 11172 data stream*. The definition of this data stream makes the following implicit assumptions. For data stored on a secondary storage medium it is possible to read such a header first (if necessary by random access). In a dialogue service like telephone or videophone applications using communication networks, the user will always get the header information first. This makes the use of an MPEG stream inconvenient in a conferencing application, as a new user may want to join an existing conference after the data stream have already been set up. The necessary header information would not be directly available for her/him, because in an ISO 11172 data stream this information is only transferred at the beginning.

For a data stream generated according to ISO 11172, MPEG provides time stamps that are necessary for synchronization. They refer to the relationship between multiplexed data streams, but not between other existing ISO 11172 data streams.

It should be mentioned that MPEG does not prescribe compression in real time. MPEG defines the process of decoding, but not the decoder itself. Different systems have been developed. Public domain software has been available in 1993, but only a few products are commercially available although many were announced. One example is CD-I.

7.4 MPEG-2

The quality of a video sequence compressed in compliance with the MPEG standard is near the optimal possible for a target maximum data rate of about 1.5 Mbit/s. This optimum is in quality and not in performance. Further developments in the area of video coding techniques are based on a target rate of up to 100 Mbits/s. This is known as *MPEG-2* (MPEG 1993b). MPEG-2 strives for a higher resolution, similar to the digital video studio standard CCIR 601 and leading to HDTV. Note that the author gleaned most of the following information on MPEG-2 and MPEG-4 from press releases and many personal communications from P. Liu and other members of the MPEG.

In order to ensure that a harmonious solution for the widest range of applications is achieved, the work group designated ISO/IEC JTC1/SC29/WG11 has been working jointly with the ITU-TS Study Group 15 "Experts Group for ATM Video Coding." MPEG-2 also collaborates with representatives from other parts of International Telecommunication Union – Telecommunication Sector (ITU-TS), International Telecommunication Union – Radiocommunication Sector (ITU-RS), European Broadcasting Union (EBU), Society of Motion Pictures and Television Engineers (SMPTE), and the North American HDTV community.

MPEG developed the *MPEG-2 Video* Standard, which specifies the coded bit stream for high-quality digital video. As a compatible extension, MPEG-2 Video builds upon the completed MPEG-1 standard by supporting interlaced video formats and a number of other advanced features including those supporting HDTV.

As a generic International Standard, MPEG-2 Video was defined in terms of extensible profiles, each of which will support the features needed by an important class of applications. The MPEG-2 Main Profile was defined to support digital video transmission in the range of about 2 to 80 Mbits/s over cable, satellite and other broadcast channels as well as for digital storage media and other communications applications. Parameters of the Main Profile and High Profile (note: the name of this "Next Profile" is preliminary) are suitable for supporting HDTV formats.

Table 4. MPEG-2 Profiles and levels with the most important characteristics; note that cells in the table without entries are not defined as compliance points [adapted from Schäfer (1993)]

High level 1920 pixels/line 1152 lines		≤ 80 Mbits/s			≤ 100 Mbits/s
High-1440 level 1440 pixels/line 1152 lines		≤ 60 Mbits/s		≤ 60 Mbits/s	≤ 80 Mbits/s
Main level 720 pixels/line 576 lines	≤ 15 Mbits/s	≤ 15 Mbits/s	≤ 15 Mbits/s		≤ 20 Mbits/s
Low level 352 pixels/line 288 lines		≤ 4 Mbits/s	≤ 4 Mbits/s		
Layers and Profiles	**Simple profile** No B frames 4:2:0 Not scalable	**Main profile** B frames 4:2:0 Not scalable	**SNR scalable** profile B frames 4:2:0 SNR scalable	**Spatially scalable profile** B frames 4:2:0 SNR scalable or spatial scalable	**High profile** B frames 4:2:0 or 4:2:2 SNR scalable or spatial scalable

The MPEG experts also extended the features of the Main Profile by defining a hierarchical/scalable profile. This profile aims to support applications such as compatible terrestrial TV/HDTV, packet-network video systems, backward compatibility with existing standards (MPEG-1 and H.261), and other applications for which multilevel coding is required. For example, such a system could give the consumer the option of using either a small portable receiver to decode standard definition TV, or a larger fixed receiver to decode HDTV from the same broadcast signal.

All profiles are arranged in a 5 × 4 matrix as shown in Table 4. The horizontal axis denotes profiles with an increasing number of operations to be supported. The vertical axis indicates levels with increased parameters. Smaller and larger frame sizes are examples for these parameters. For example, the *main profile* in the *low level* defines 352 pixels/line, 28 lines/frame, 30 frames/s, B frames are allowed to occur, and the data rate shall not exceed 4 Mbits/s; the *main profile* in the *high level* specifies 1920 pixels/line 1152 lines/frame and 60 frames/s with a data rate to be less than 80 Mbits/s. MPEG-2 considers a structure similar to that of the hierarchical mode of JPEG, a scaling of the compressed motion images (personal communication from E. Viscito) where video is encoded at different "qualities" (Lippman 1991])

The scaling may act on different parameters:

- A spatial scaling facilitates decompression of image sequences with dissimilar horizontal and vertical resolutions. A single data stream could include images with 352 × 288 pixels (H.261 CIF format), 360 × 240 pixels, 704 × 576 pixels (a format according to CCIR 601) and, for example, with 1250 lines at an aspect ratio of 16:9 (European HDTV). These resolutions refer to the luminance component, the chrominance components are subsampled with ratio 1:2. This can be implemented using a pyramid for the level of the DCT coefficients. Thereby a 8 × 8 DCT, 7 × 7 DCT, 6 × 6 DCT, and other transformations can be performed. From the technical point of view, only steps with the factor two are useful.
- Scaling of the data rate allows for a playback with a lower frame rate or for a fast forward mode with a constant frame rate. In MPEG-1 this is defined with D frames. D frames are not allowed in MPEG-2. If there is a suitable distribution of I frames within the data stream, they can be used to scale the data rate. This distribution must apply to the whole video clip and not only to a group of pictures.
- The scaling in amplitude can be interpreted as a different resolution of different pixels or a different quantization of the DCT coefficients. This leads to layered coding and to the possibility of progressive image presentation. Progressive coding is not at all important for the presentation of video data. However, it should be possible to extract certain still images from a sequence in the data stream, in which case progressive coding may be of interest. Layered coding can be used to partition data for the transmission of the more important data with better error correction.

Scaling is an essential extension of MPEG-1 to MPEG-2. MPEG-2 considers the current developments in the broadband ISDN (BISDN) world. The asynchronous transfer mode (ATM) is the realization of BISDN based on the transfer of a small packet known as cells. A potential loss of single ATM cells containing MPEG-2 encoded data is taken into account in the MPEG-2 development. In such a case effects on other images and other parts of the video data stream must be minimized. It should also be possible to define sequences of the different types of images (I, P, B) that allow the minimizing of the end- to-end delay for a given target data rate.

MPEG developed the *MPEG-2 Audio Standard* for low-bit-rate coding of multichannel audio. MPEG-2 audio coding supplies up to five full bandwidth channels (left, right, center, and two surround channels), plus an additional low-

frequency enhancement channel, and/or up to seven commentary/multilingual channels. The MPEG-2 Audio Standard also extends stereo and mono coding of the MPEG-1 Audio Standard to half sampling rates (16 kHz, 22.05 kHz, and 24 kHz) for improved quality of bit rates at or below 64 kbits/s per channel.

The MPEG-2 Audio Multichannel Coding Standard provides backward compatibility with the existing MPEG-1 Audio Standard. MPEG organized formal subjective testing of the proposed MPEG-2 multichannel audio codecs and up to three nonbackward compatible codecs. These coders work with rates ranging from 256 to 448 kbits/s.

Note that in order to provide a very accurate description, the following text was written with extensive use of the notation and terminology defined by the original MPEG-2 specification. As MPEG-2 addresses video as well as associated audio, it provides the MPEG-2 system specification to define how audio, video, and other data are combined as single or multiple streams that are suitable for storage and transmission. Therefore it imposes syntactical and semantical rules, which are necessary for and capable of synchronizing the decoding and presentation of the video and audio information, while ensuring that coded data buffers in the decoder do not overflow or underflow. The streams include time stamps for the decoding, the presentation, and the delivery of this data.

In the first step, the basic multiplexing approach adds information of the system level to each individual stream, which is packetized to produce the packetized elementary stream (PES). In the subsequent step the PESs are combined as a *program* or a *transport* stream. Both program and transport streams are designed to support a large number of known and anticipated applications, and they retain a significant amount of flexibility, such as may be required for such applications, while providing interoperability between different device implementations.

- The *program stream* is similar to the MPEG-1 stream; it is aimed at a relatively error-free environment. The program stream's packets may be of variable length. The timing information in this stream can be used to implement a constant end-to-end delay (covering the path from the input of the encoder to the output of the decoder).
- The *transport stream* combines the PESs with one or several independent time bases into one single stream. The transport stream is designed for use in lossy or noisy media. The respective packets are 188 bytes long, including the 4-byte header. The transport stream is well-suited for the transmission of digital television and video telephony over fiber, satellite, cable, ISDN, ATM, and other networks, and also for storage on digital video tape and other devices.

A conversion between the program and the transport stream is possible and reasonable. Note that the definition of MPEG-2 in its buffer management section does constrain the end-to-end delay below 1 s for audio and video data, a value that is too high, i.e., not humanly acceptable, for applications in the dialogue mode.

A typical MPEG-2 video stream has variable bit rate. With the use of a video buffer defined in this standard it is also possible to enforce a constant bit rate leading to a varying quality.

Typically a standard like MPEG-2 – at the committee draft (CD) status in late 1993 – requires 3 months to become draft international standard (DIS) and then a 6-month ballot period before becoming an international standard (IS).

Originally there were plans to specify an MPEG-3 standard approaching HDTV. During development, MPEG-2 proved adequate when scaled up to meet HDTV requirements. Subsequently MPEG-3 was dropped.

7.5 MPEG-4

Work on another MPEG initiative for very-low-bit rate coding of audiovisual programs started in September 1993 at ISO/IEC JTC1. It is scheduled to result in the CD status in 1995 or 1996.

This work will require the development of fundamentally new algorithmic techniques, including model-based image coding for human interaction with multimedia environments and low-bit-rate speech coding for use in environments like in the European mobile telephony system.

8 DVI

Digital Video Interactive (DVI) is a technology that includes coding algorithms. The fundamental components are a VLSI chip set for the video subsystem, a well-specified data format for audio and video files, an application user interface for the audio-visual kernel (AVK), the kernel software interface for the DVI hardware, and compression as well as decompression algorithms (Harney et al. 1991; Luther 1991; Ripley 1989). In this chapter we concentrate mainly on compression and decompression. DVI can process data, text, graphics, still images, video and audio. The original essential characteristic was the asymmetric technique of video compression and decompression known as presentation level video (PLV).

DVI has a very interesting history, which helps explain some of the features of this system today. A project was started at David Sarnoff Research Center of the RCA company in Princeton in 1984. At that time the major goals – to compress video and audio to the data rate appropriate for a CD and to decompress it in real time – were defined. In 1986 the prototype of a DVI-specific chip using a silicon compiler was developed. In 1986 General Electrics (GE) took over this technology, the DVI development team became employees of GE, and the project continued. The first public presentation took place at the second Microsoft CD-ROM Conference in Seattle in March 1987. For the first time the real-time playback of video stored on a CDROM was presented. In 1989, at the fourth Microsoft CD-ROM Conference, IBM and Intel announced their cooperation concerning DVI. The DVI team was later taken over by Intel. The first generation of PS/2 boards were introduced as ActionMedia 750. In April 1992, the second generation of these boards for Microchannel and ISA bus machines (ActionMedia II) became available. In 1993

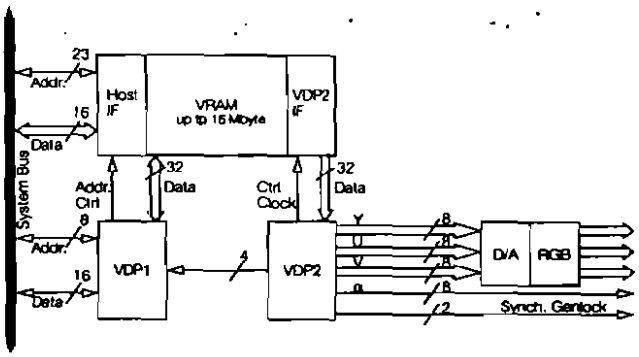


Fig. 18. DVI video processor according to Ripley (1989)

the software-only decoder became a product; it is known as Indeo.

Concerning audio, the demand for a hardware solution implemented at a reasonable price is met using a standard signal processor. Still images and video are processed by a video processor. The video hardware of a DVI board is shown in Fig. 18. It consists of two VLSI chips containing more than 265,000 transistors each. This video display processor (VDP) consists of the pixel processor VDP1 and the display processor VDP2. The VDP1 processes bitmaps and is programmed in microcode, the VDP2 generates analogue RGB signals from the various bitmap formats, and its configuration is also programmable. The processors are coupled by the video RAM (VRAM). An important characteristic is the ability to microprogram. It allows one to change and adapt the compression and decompression algorithms to new developments without new hardware investments.

8.1 Audio and still image encoding

Audio signals are digitized with 16 bits per sample and are either PCM encoded or compressed with the Adaptive Differential Pulse Code Modulation (ADPCM) technique. Thereby a reduction to about 4 bits per sample is achieved at a quality corresponding to stereo broadcasting. Different sampling frequencies are supported (11,025 Hz, 22,050 Hz and 44,100 Hz for one or two PCM-coded channels; 8,268 Hz, 31,129 Hz and 33,075 Hz for ADPCM at one channel).

Various video input formats can be used to encode still images. These can be composite, as well as component, video signals like RGB. In case of an RGB signal, the color of each pixel is split into portions of the three colors of the spectrum, red, green, and blue, and each color is processed separately.

For *image preparation*, DVI assumes an internal digital YUV format, i.e., any video input signal must first be transformed into this format. Note that by DVI we mean the Action Media II version. The color of each pixel is split up into a luminance component Y and the two chrominance components U and V. The luminance represents a gray-scale image. White is not a basic color, but a mixture of colors. In the case of a RGB signal, a pure white pixel consists of about 30% red, 59% green, and 11% blue. Starting with a RGB signal, DVI computes the YUV signal using the following relationship:

$$Y = 0.30R + 0.59G + 0.11B, U = B - Y, V = R - Y$$

leading to:

$$U = -0.30R - 0.59G + 0.89B$$
$$V = 0,70R - 0,59G - 0,11B .$$

Therefore DVI determines the components YUV according to the following relation:

$$Y = 0.299R + 0.587G + 0.144B + 16$$
$$U = 0.577B - 0.577Y + 137.23$$
$$V = 0.730R - 0.730Y + 139.67 .$$

This is realized in software with fixed point arithmetic based on:

$$109Y = 32R + 64G + 16B + 1744 \text{ with } 0 \leq R, G, B \leq 255$$
$$111U = 64B - 64Y + 15216 \text{ with } 16 \leq Y \leq 235$$
$$88V = 64R - 64Y + 12253 \text{ with } 16 \leq U, V \leq 240 .$$

DVI always combines all chrominance components of 4×4 blocks of pixels in a single value. The chrominance component of the top left pixel of such a block is used as the reference value for the 16 pixels. Therefore each pixel has an 8-bit value for the luminance Y, and for 16 related pixels a single 8-bit chrominance value, U, and another 8 bits defining the value for V. The result is a *9-bit YUV format*.

In order to increase the image quality during presentation, the chrominance values of adjacent blocks are interpolated. Note that this is the reason for the recognizable color distortion at the right and at the bottom edges of the images. Additionally, DVI is able to process images in the *16-bit YUV format* and the *24-bit YUV format*. The 24-bit format uses 8 bits for each component. The 16-bit YUV format codes the Y component of each pixel with 6 bits and the color difference components with 5 bits each. This is the reason for two different bitmap formats, *planar* and *packed*. For the planar format, all the data of the Y component are stored first, followed by all the U component values, then all V values (9-bit YUV format or 24-bit YUV format). For the packed bitmap format, the Y, U, and V information of each pixel is stored together, followed by the data of the next pixel in the 16-bit YUV format.

Single images in 24-bit format can be stored immediately or transmitted. Images in the 16-bit format can be compressed with a lossless algorithm that is known as PIC 1.0. In 9-bit format there is a choice between a lossy algorithm and the JPEG baseline mode; for backward compatibility to the previous DVI algorithms, two additional compression schemes can be applied.

8.2 Video encoding

For *motion video encoding* DVI distinguishes two techniques with different resolutions and dissimilar goals:

- Presentation level video (PLV) is characterized by its better quality. This is achieved at the expense of a very time-consuming asymmetric compression performed by specialized compression facilities. In the early stage of the DVI technology, PLV compression required, for example, a Meico Transputer System with more than 60 transputers to compress one image in 3 s, which corresponds to a 90-fold increase in the time needed for such an operation compared to the real time constraints. PLV is suitable for applications distributed on CD-ROMs. The development of such DVI applications in the PLV mode follows the process shown in Fig. 19.
- Real-time video (RTV) is a symmetric compression technique that works with hardware and software in real time (RTV version 2.0 uses the i750PB chip). Known as Indeo, it can also run on processors such as Intel 386/486 in real time with certain limitations, especially reduced quality of the images. In previous versions RTV was known as edit level video (ELV). ELV was conceived to enable the developer of DVI applications to see his/her video sequences with reduced quality during the construction phase. Afterwards, he/she sends the videotapes to be compressed in the PLV mode to a DVI compression facility and gets compressed video sequences of higher quality than RTV back. Today, RTV is most often used for interactive communication in the same manner as px64.

With the aid of our reference processing model shown in Fig. 1 we can again distinguish various steps:

The *image preparation* phase of RTV distinguishes three components of an image, where all pixels are coded with 8 bits each. As subsampling is used, the luminance has a higher resolution than the chrominance components. For each 16 pixels of luminance there is one pixel in each of the chrominance components U and V. Consequently, the luminance component consists of four times as many lines and columns as the other two components. In a block of 16 Y pixels, 1 U pixel and 1 V pixel are encoded together. The result is the 9-bit YUV format of RTV (for each 8 Y bits there is one U bit or one V bit).

It should be mentioned that the RTV algorithms described in the following may also make use of other image preparation schemes that would have to be supported by the AVK. The following processing in the RTV algorithm treats all three components with the same scheme.

The *image processing* of RTV distinguishes between an interframe coding and an intraframe coding.

- intraframe coding is based on individual images. The difference between the value of each pixel and the adjacent pixel above is calculated. For the first line, a fictitious line above is used, which has a constant value. This calculation is performed for all components, and it results in many zero values, which is excellent for the consecutive entropy coding steps.

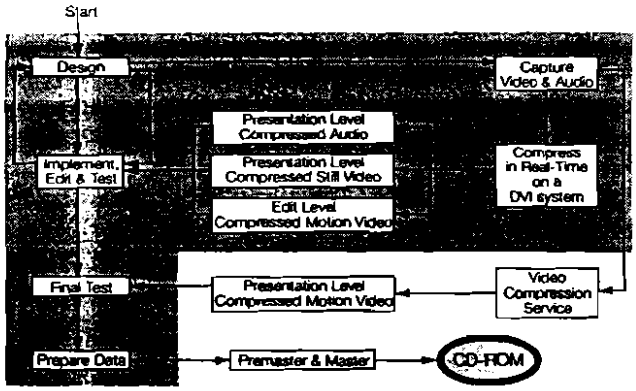


Fig. 19. DVI generation process of a PLV coded video sequence as shown in Ripley (1989)

- interframe coding determines the difference between the value of a pixel in the current image and the value of the pixel located at the same place in the preceding image. This is done for all of the three components – normally the differences also consist of many zero values.

A quantization is not necessary because of the simple subtraction operations mentioned. The *entropy encoding* is based on a linear data stream, and follows the calculation of differences immediately. It is used for both interframe and intraframe coding. A distinction between run length encoded databytes (these are the zero bytes) and the remaining vector is made:

- Sequences of existing zero bytes are compressed with run-length coding.
- All other bytes are compressed with a two-dimensional vector encoding technique. An index in one of the eight available tables that corresponds to two adjacent pixels to be compressed is determined. The various components of an image are usually encoded via different tables.

In a last step, the run-length encoded values and the indices determined previously are transformed with another table and subsequently Huffman encoded. It is possible to select a new Huffman table that is adapted to the specific content for each image.

PLV is an asymmetric compression technique that is proprietary to Intel and not published in detail (Harney et al. 1991). However, the principle is known. Each picture is divided into rectangular blocks and motion compensated. For each block, a prediction in form of a block of the previous image is made. If its position has changed, its motion is recorded in a motion vector. An exceptional feature should be mentioned; the motion vector is measured in terms of pixels but its values can be real numbers. The interpolation between the values of pixels can result in a motion vector with values of fractions of pixel widths and heights, leading to a better resolution. However, the disadvantage is the penalty caused by processing real numbers instead of integers.

The difference of the predicted block and the actual block of the previous picture is coded as in the previously described

RTV technique (two-dimensional vector encoding and Huffman encoding).

8.3 Data stream

Besides the actual compression technique, DVI defines a *data stream*. As an example, when a data stream including audio and PLV-encoded video is used, a subdivision into single images is the first step. In addition to the actual image data, the following information is included:

- version label
- information on the choice of interframe or intraframe encoding
- height and width of the image (number of pixels)
- which of the eight tables must be used for the twodimensional vector encoding of this image
- Huffman tables
- whether half resolution in vertical and/or horizontal dimension is used

Additionally, there is the PCM or ADPCM encoded audio data in the stream.

9 Conclusion

All of the important compression techniques used in multimedia systems turn out to be combinations of various known algorithms.

JPEG must be considered as the future standard for coding of still images. It incorporates a remarkable variety of alternative modes with a high degree of freedom. For example, there could be up to 255 components or planes, an image may comprise as many as 65,535 lines, each line can have as many as 65,525 pixels. As a measure of efficiency, the required bits per pixel can be calculated, i.e., the average value determined by the ratio of the number of encoded bits and the number of pixels of the image. The following statements apply to DCT encoded still images (Wallace 1991):

- 0.25 to 0.50 bits/pixel: moderate to good quality, sufficient for some applications
- 0.50 to 0.75 bits/pixel: good to very good quality, sufficient for many applications
- 0.75 to 1,50 bits/pixel: excellent quality suitable for most applications
- 1,50 to 2,00 bits/pixel: in most cases not distinguishable from the original, sufficient for almost all applications, even for highest quality requirements.

In the lossless mode a compression ratio of 2:1 is reached despite the remarkable simplicity of the technique. Today JPEG is available both as a software and as a hardware solution; however, in most cases only the baseline mode with a predefined image format is supported. JPEG is often used in multimedia applications that require high quality. The primary domain is image. However it is also used as Motion JPEG for video compression; medical imaging is an example of this application area.

H.261 is an already established standard that is strongly supported by telecommunication operators. Due to a very restricted resolution in the QCIF format and reduced frame rates, the implementation of H.261 coders and decoders is possible without any technical problems. This is certainly true if the motion compensation and the optical low-pass filter are not components of the implementation, though the quality is not always satisfactory in this case. If the image is encoded in CIF format at 25 or 30 images/s with motion compensation, the quality is acceptable. The major application domain is the dialog mode in a networked environment: video telephoning and conferencing. The resulting continuous bit rate is eminently suitable for today's wide area networks operation with ISDN and leased lines.

MPEG is the most promising standard for future compressed digital video and audio. Although the JPEG group has a system that can also be used for video, it is too oriented towards "animating" stills rather than towards the properties of motion pictures. Currently, the quality of MPEG video can be compared to VHS video recordings, with a data rate of 1.2 Mbits/s, an appropriate rate for CD-ROM drives. Referring to the compression, the algorithm is very good for a resolution of 360 × 240 pixels. Obviously higher resolutions can also be decoded, for example, a resolution of 625 lines, but the quality is affected. The future of MPEG points towards MPEG-2, which defines a data stream that is compatible with MPEG-1, but providing data rates up to 100 Mbits/s. This substantially improves the currently available quality of MPEG coded data. MPEG also defines an audio stream, with various sampling rates up to DAT quality at 16 bits/sample. One other important part of the MPEG group's work is the definition of a syntax of a data stream, which has proved to be relatively successful for the DVI technology. MPEG was optimized for multimedia applications with the use of the retrieval model. CD-ROM based tutoring systems and interactive TV are such typical application areas. MPEG-2 will allow for TV and HDTV quality at the expense of a higher data rate. MPEG-4 will provide a very high compression ratio for the coding of video and the associated audio.

Intel now owns *DVI*, which, unlike the other systems, is a proprietary invention. It defines two quality encoding variants, one for real-time compression given the appropriate hardware and the other for off-line compression, but allowing decompression with the same hardware. The resolutions are 512×480 (interpolated from 256 × 240) and 256 × 240 respectively. As already mentioned, DVI also features a file syntax known as AVSS. Due to the standardization of the other formats mentioned, as well as the demand for interchangeable formats, it can be expected that RTV, PLV, or both will incorporate more features from these standards in order to provide compatibility. DVI also defines audio and still image compression of very good quality. For still images a certain configuration of the JPEG format is supported. The video quality in PLV mode is very good and it allows for use in retrieval mode applications similar to MPEG. The RTV mode is good and quite convincing for many applications. As described in this paper, RTV accommodates dialogue mode applications. However, many

available implementations suffer from the considerable compression/decompression delay longer than 150 ms.

JPEG, H.261, MPEG, and DVI are not alternative techniques for data compression. Their goals are different and partly complementary. Most of the algorithms used are very similar but not the same. The technical quality as well as the availability on the market determine the techniques that will be used in future multimedia systems. In the author's opinion, this will lead to a cooperation and a convergence of the techniques. For instance, a future multimedia computer might generate still images in JPEG, use H.261 for a video conference and MPEG-2 as well as DVI PLV for the retrieval of stored multimedia information. (Note that this is a purely hypothetical statement and it does not prejudice any type of future development or strategy in this field.)

Acknowledgements. The author gratefully acknowledges the work of Sven Dryoff and Stefan Mengler in the DVI technology section. Doris Meschzan and Ian Marsh helped with many valuable details. The anonymous reviewers and Ralf Guido Herrtwich provided many extra refinements for the later editions of this paper. Mike Salmony spent many hours improving my English and checking many technical details. Thank you.

Note. There are several ways to order the mentioned Standards of ISO. In the US, for example, one may call ANSI (212-642-4900) or Global Engineering Documents, Washington, DC (800-854-7179)

References

ACM (1989) Special Section on Digital Multimedia Systems. Commun. ACM 32 (July): 794–889

ACM (1991) Special Section on Interactive Technology. Commun. ACM 34 (April): 26–119

Ahmed N, Natarajan T, Rao KR (1974) Discrete Cosine Transform. IEEE Trans. Comput 23:90–93

Aravind R, Cash GL, Duttweiler DL, Hang HM, Haskell BG, Puri A (1993) Image and video coding standards. AT&T Technical J 72 (January/February): 67–89

Atal BS, Cuperman V, Gersho A (eds) (1993) Speech and Audio Coding for Wireless and Network Applications. Kluwer, Dordrecht

Barnsley MF, Hurd LP (1993) Fractal image compression. AK Peters, Wellesley, Mass.

Blair G, Hutchison D, Shepherd D (1991) Multimedia systems tutorial. Proc. 3rd IFIP Conference on High-Speed Networking, Berlin

CCIR (1982) (International Radio Consultative Committee) Encoding Parameters of Digital Television for Studios, Recommendation 601

CCITT (1990) Line transmission on non-telephone signals: video codec for audiovisual services at px64 kbit/s. International Telecommunication Union, the International Telegraph and Telephone Consultative Committee (CCITT) Recommendation H.261, Geneva

Duhamel P, Guillemot C (1990) Polynomial transform computation of the 2-D DCT. Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, Albuquerque, N.M., pp 1515–1518

Feig E (1990) A fast scaled DCT algorithm. In: Pennington KS, Moorhead RJ II (eds) Image Processing Algorithms and Techniques, Proc. SPIE 1244:2–13, Santa Clara, Calif.

Gonzalez RC, Woods RE (1992) Digital Image Processing. Addison-Wesley, Reading

Gray RM (1984) Vector quantization. IEEE ASSP Magazine 1 (April): 4–29

Harney K, Keith M, Lavelle G, Ryan LD, Stark DJ (1991) The i750 video processor: a total multimedia solution. Commun ACM 34 (April): 64–78

Herrtwich RG, Steinmetz R (1991) Towards integrated multimedia systems: why and how. Informatik-Fachberichte, no. 293, Springer, Berlin Heidelberg New York, pp 327–342

Hou HS (1987) A fast recursive algorithm for computing the discrete cosine transform. IEEE Trans Acoustics Speech Signal Processing 35:1455–1461

Hudson G, Yasuda H, Sebestyen I (1988) The international standardization of a still picture compression technique. Proc IEEE Global Telecommunications Conference, pp 1016–1021

Huffman DA (1952) A method for the construction of minimum redundancy codes. Proc IRE 40, pp 1098–1101

IEEE (1992a) Special Section on Signal Processing and Coding for Recording Channels. IEEE Journal on Selected Areas in Communication 10 (January): 1–299

IEEE (1992b) Special Section on Speech and Image Coding. IEEE Journal on Selected Areas in Communication 10 (June): 793–975

IS&T/SPIE (1994) Symposium on Electronic Imaging: Conference on Digital Video Compression on Personal Computers: Algorithms and Technologies, February 7–8, Proceedings SPIE/IS&T 2187

Jayant NS, Noll P (1984) Digital coding of waveforms. Prentice-Hall, Englewood Cliffs, New Jersey

JPEG (1993) ISO IEC JTC 1 Information Technology – Digital Compression and Coding of Continuous-Tone Still Images. International Standard ISO/IEC IS 10918

Lamparter B, Effelsberg W (1991) X-MOVIE: transmission and presentation of digital movies under X. 2nd International Workshop on Network and Operating System Support for Digital Audio and Video, Heidelberg, Lecture Notes in Computer Science, no. 614, Springer-Verlag, pp 328–340

Lamparter B, Effelsberg W, Michl N (1992) MTP: a movie transmission protocol for multimedia applications. Proc of the 4th IEEE ComSoc International Workshop on Multimedia Communications, Monterey, Calif., pp 260–270

Langdon G (1984) An introduction to arithmetic coding. IBM J Res Devel 28 (March): 135–149

Le Gall D (1991) MPEG: A video compression standard for multimedia applications. Commun ACM 34 (April): 46–58

Lee BG (1984) A new algorithm to compute the discrete cosine transform. IEEE Trans Acoustics Speech and Signal Processing 32:1243–1245

Leger A, Mitchell J, Yamazaki Y (1988) Still picture compression algorithm evaluated for international standardization. Proc IEEE Global Telecommunications Conference, pp 1028–1032

Leger A, Omachi T, Wallace GK (1991) JPEG still picture compression algorithm. Optical Eng 30:947–954

Linzer EN, Feig E (1991) New DCT and scaled DCT algorithms for fused multiply/add architectures. Proc IEEE International Conference on Acoustics, Speech and Signal Processing, Toronto, Canada, pp 2201–2204

Liou M (1991) An overview of the px64 kbit/s video coding standard. Commun ACM 34 (April): 59–63

Lippman A (1991) Feature sets for interactive images. Commun ACM 34 (April): 92–101

Lu G (1993) Advances in digital image compression techniques. Comput Commun 16:202–214

Luther (AC 1991) Digital video in the PC environment. Intertext Publications, McGraw-Hill, New York

Mitchell JL, Pennebaker WB (1991) Evolving JPEG color data compression standard. In: Nier M, Courtot ME (eds) Standards for Electronic Imaging systems. SPIE CR37:68–97

MPEG (1993a) ISO IEC JTC 1 (1993) Information Technology – Coding of moving pictures and associated audio for digital storage media up to about 1.5 Mbit/s. ISO/IEC IS 11 172

MPEG (1993b) ISO IEC JTC 1 (1993) Information Technology – Coding of moving pictures and associated audio for digital storage media, test model 4. Draft, MPEG 93/255b

Musmann HG (1990) The ISO audio coding standard. IEEE Globecom 90, San Diego, Calif. pp 511–517

Narasinha NJ, Peterson AM (1978) On the computation of the discrete cosine transform. IEEE Trans Commun 26:966–968

Netravali AN, Haskell BG (1988) Digital pictures: representation and compression. Plenum Press, New York

Pennebaker WB, Mitchell JL, Langdon G Jr, Arps RB (1988) An overview of the basic principles of the Q-coder binary arithmetic coder. IBM J Res Devel 32:717–726

Pennebaker WB, Mitchell JL (1993) JPEG still image data compression. Van Nostrand Reinhold, New York

Puri A, Aravind R (1991) Motion compensated video coding with adaptive perceptual quantization. IEEE Trans Circuits Syst Video Technol 1:351

Rabbani M, Jones P (1991) Digital Image Compression Techniques. Tutorial Texts in Optical Engineering 7, SPIE

Ripley GD (1989) DVI – a digital multimedia technology. Commun ACM 32:811–822

Schäfer R (1993) Source coding for television – European approaches; digital television – digital radio technologies of tomorrow. "Münchner Kreis" Congress, Munich, Germany

Steinmetz R (1993) Multimedia Technology: Fundamentals and Introduction (in German). Springer, Berlin Heidelberg New York

Storer JA (1988) Data Compression Methods and Theory. Computer Science Press, Rockville, Md.

Suehiro N, Hatori M (1986) Fast algorithms for the DFT and other sinusoidal transforms. IEEE Trans Acoustics Speech Signal Processing 34:642–644

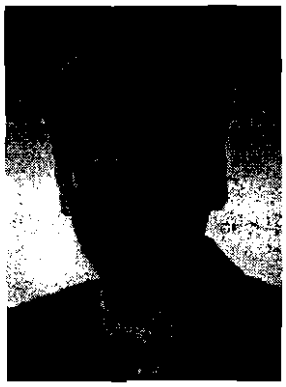
Vetterli M (1985) Fast 2-D discrete cosine transform. Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, pp 1538–1541

Vetterli M, Nussbaumer HJ (1984) Simple FFT and DCT algorithms with reduced number of operations. Signal Processing (August)

Viscito E, Gonzales C (1991) A video compression algorithm with adaptive bit allocation and quantization. Proc. SPIE Visual Communications and Image, Boston, Mass.

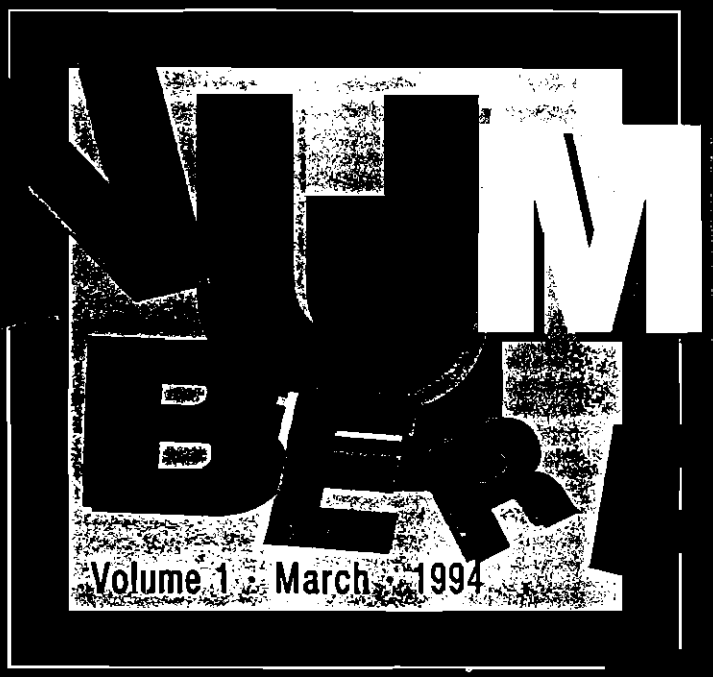
Wallace GK (1991) The JPEG still picture compression standard. Commun ACM 34 (April): 30–44

Wallace G, Vivian R, Poulsen H (1988) Subjective testing results for still picture compression algorithms for international standardization. Proc. IEEE Global Telecommunications Conference, pp 1022–1027



RALF STEINMETZ studied electrical engineering specializing in communications at the University of Salford, England, and at the Technical University of Darmstadt, Germany, where he received an M.Sc. degree in engineering (1982) and a Ph.D. degree in engineering (1986). Thereafter, he focused his work on concurrency in programming languages and OCCAM, and he wrote the first book on OCCAM-2 in German. In 1987 and 1988 he worked for Philips in the area of ISDN and multimedia user interfaces. Since 1988 he has worked at the IBM European Networking Center in Heidelberg, Germany, where he has been involved in various multimedia communication activities and has acted as a key technical coordinator for several projects. He lectures at the University of Frankfurt on "distributed multimedia systems". He is coeditor and coauthor of a multimedia course that reflects the major issues of his in-depth technical book on multimedia technology published in 1993 (in German). He is an editor of Computer Communications published by Butterworths-Heinemann and of the journal Distributed Systems Engineering. He is the associate editor-in-chief of the IEEE Multimedia Magazine. He has served as chairman, vice-chairman and member of various program and organizing committees of multimedia workshops and conferences. He is a member of ACM, GI, ITG, and a senior member of IEEE.

Multimedia SYSTEMS



acm PRESS Association for Computing Machinery

Springer International