MPEG video coding
A simple introduction

Dr. S.R. Ely (BBC)

The core element of all DVB systems is the MPEG-2 vision coding standard, which is based upon a flexible toolkit of techniques for bit-rate reduction.

The MPEG-2 specification only defines the bit-stream syntax and decoding process. The coding process is not specified, which means that compatible improvements in the picture quality will continue to be possible.

In this article, the author provides a simple introduction to the technicalities of the MPEG-2 video coding standard.

1. Introduction

The Moving Pictures Expert Group (MPEG) started in 1988 as Working Group 11, Subcommittee 29, of ISO/IEC JTC1[1] with the aim of defining the standards for digital compression of video and audio signals. It took as its basis the ITU-T[2] standard for video-conferencing and video-telephony [1], together with that of the Joint Photographic Experts Group (JPEG) which had initially been developed for compressing still images such as electronic photography [2].

The first goal of MPEG was to define a video coding algorithm for digital storage media, in particular CD-ROM. The resulting standard was published in 1993 as ISO/IEC 11172 [3] and comprises three parts, covering the *systems aspects* (multiplexing and synchronization), *video coding* and *audio coding*. This standard has been applied in the CD-i system to provide full motion video playback from CD, and is widely used in PC applications for which a range of hardware and software coders and decoders are available.

MPEG-1 is restricted to non-interlaced video formats and is primarily intended to support video coding at bit-rates up to about 1.5 Mbit/s.

1. Joint Technical Committee No. 1 of the International Organisation for Standardisation and the International Electrotechnical Commission.
2. International Telecommunication Union – Telecommunication Standardization Bureau.

Original language: English
Manuscript received 30/11/95.

In 1990, MPEG began work on a second standard which would be capable of coding interlaced pictures directly, originally to support high-quality applications at bit-rates in the range of about 5 to 10 Mbit/s. MPEG-2 now also supports high-definition formats at bit-rates in the range of about 15 to 30 Mbit/s. The MPEG-2 standard was first published in 1994 as ISO/IEC 13818, again comprising three parts – systems, video and audio. A second version of the standard was published in 1995 [4].

It is important to note that the MPEG standards specify only the syntax and semantics of the bit-streams and the *decoding* process. They do not specify the *coding* process: this is left mainly to the discretion of the coder designers, thus giving scope for improvement as coding techniques are refined and new techniques are developed.

2. Video coding principles

If we take a studio-quality 625-line component picture and digitize it according to ITU Recommendations BT.601 [5] and BT.656 [6] (i.e. if we use 4:2:2 sampling as shown in *Fig. 1*), a bit-stream of 216 Mbit/s is used to convey the luminance and the two chrominance components. For bandwidth-restricted media – such as terrestrial or satellite channels – a method is required to reduce the bit-rate needed to represent the digitized picture.

A video bit-rate reduction (compression) system operates by removing the redundant and less-important information from the signal prior to transmission, and by reconstructing an approximation of the image from the remaining information at the decoder. In video signals, three distinct kinds of redundancy can be identified:

1) Spatial and temporal redundancy

Here, use is made of the fact that the pixel values are not independent but are correlated with their neighbours, both within the same frame and across frames. So, to some extent, the value of a pixel is predictable, given the values of the neighbouring pixels.

2) Entropy redundancy

For any non-random digitized signal, some code values occur more frequently than others. This can be exploited by coding the more-frequently occurring values with shorter codes than the rarer ones. This same principle has long been exploited in Morse code where the most common letters in English, “E” and “T”, are represented by one dot and one dash, respectively, whereas the rarest letters, “X”, “Y” and “Z”, are each represented by a total of four dots and dashes.

3) Psycho-visual redundancy

This form of redundancy results from the way the eye and the brain work. In audio, we are familiar with the limited frequency response of the ear: in video, we have to consider two limits:

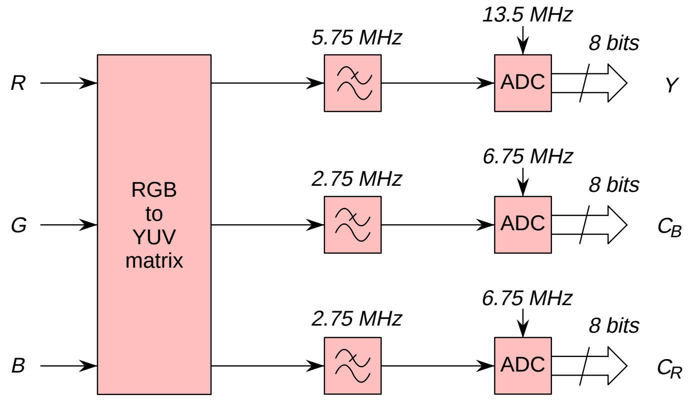
- the limit of *spatial resolution* (i.e. the fine detail which the eye can resolve);
- the limit of *temporal resolution* (i.e. the ability of the eye to track fast-moving images). Temporal resolution means, for example, that a change of picture (a shot-change) masks the fine detail on either side of the change.

3. MPEG video compression toolkit

Sample-rate reduction is a very effective method of reducing the bit-rate but, of course, it introduces irreversible loss of resolution. For very low bit-rate applications (e.g. in MPEG-1), alternate fields are discarded and the horizontal sampling-rate is reduced to around 360 pixels-per-line (giving about 3.3 MHz resolution). The sample rate for the chrominance is half that of the luminance, both horizontally and vertically. In this way, the bit-rate can be reduced to less than one fifth that of a conventional definition (4:2:2) sampled signal.

For “broadcast quality” at bit-rates in the range 3 to 10 Mbit/s, horizontal sample-rate reduction is not advisable for the luminance or chrominance signals, nor is temporal sub-sampling. However, for distribution and broadcast applications, sufficient chrominance resolution can be provided if the sampling frequency of the vertical chrominance is halved. Thus, for most MPEG-2 coding applications, 4:2:0 sampling is likely to be used rather than 4:2:2. However, 4:2:2 and 4:4:4 sampling are also supported by MPEG-2. It may be of interest to note that a conventional delay-line PAL decoder effectively yields the same vertical sub-

Figure 1
4:2:2 sampling.



$$
\begin{aligned}
Y &= 8 \times 13.5 = 108 \\
C_B &= 8 \times 6.75 = 54 \\
C_R &= 8 \times 6.75 = 54 \\
\text{Total} &= 216\ \text{Mbit/s}
\end{aligned}
$$

sampling of the chrominance signals as does 4:2:0 sampling.

Apart from sample-rate reduction, the MPEG toolkit includes two different kinds of tools to exploit redundancy in images:

1) Discrete Cosine Transform (DCT)

The purpose of using this orthogonal transform – which is similar to the *Discrete Fourier Transform (DFT)* – is to assist the processing which removes spatial redundancy, by concentrating the signal energy into relatively few coefficients.

2) Motion-compensated interframe prediction

This tool is used to remove temporal redundancy. It is based on techniques similar to the well-known differential pulse-code modulation (DPCM) principle.

3.1. Discrete cosine transform

The luminance signal of a 4:2:0-sampled digitized 625-line picture comprises about 704 pixels horizontally and about 576 lines vertically (see *Fig. 2*). In MPEG coding, spatial redundancy is removed by processing the digitized signals in 2-D blocks of 8 pixels by 8 lines (taken from either one field or two, depending on the mode of operation).

As *Fig. 3* illustrates, the DCT transform is a reversible process which maps between the normal 2-D presentation of the image and one which represents the same information in what may be thought of as the *frequency domain*. Each coefficient in the 8 x 8 DCT domain block indicates the contribution of a different DCT "basis" function to the original image block. The lowest frequency basis function (top-left in *Fig. 3*) is called the *DC coefficient* and may be thought of as representing the average brightness of the block.

The DCT does not directly reduce the number of bits required to represent the block. In fact, for an 8 x 8 image block of 8-bit pixels, the DCT produces an 8 x 8 block of at least 11-bit DCT coefficients, to allow for reversibility! The reduction in the number of bits follows from the fact that, for typical blocks of natural images, the distribution of coefficients is non-uniform – the transform tends to concentrate the energy into the low-frequency coefficients, and many of the other coefficients are near zero. The bit-rate reduction is achieved by not transmitting the near-zero coefficients, and by quantizing and coding the remaining coefficients as described below. The distribution of the non-uniform coefficients is a result of the spatial redundancy present in the original image block.

Many different forms of transformation have been investigated for bit-rate reduction. The best transforms are those which tend to concentrate the energy of a picture block into a few coefficients. The DCT is one of the best transforms in this respect and has the advantage that the DCT and its inverse are easy to implement in digital processing. The choice of an 8 x 8 block-size is a trade-off between the need to use a large picture area for the transform, so the energy compaction described above is most efficient, and the fact that the content and movement of the picture varies spatially, which would tend to support a smaller block-size. A large block-size would also emphasize variations from block-to-block in the decoded picture; it would also emphasize the effects of "windowing" by the block structure.

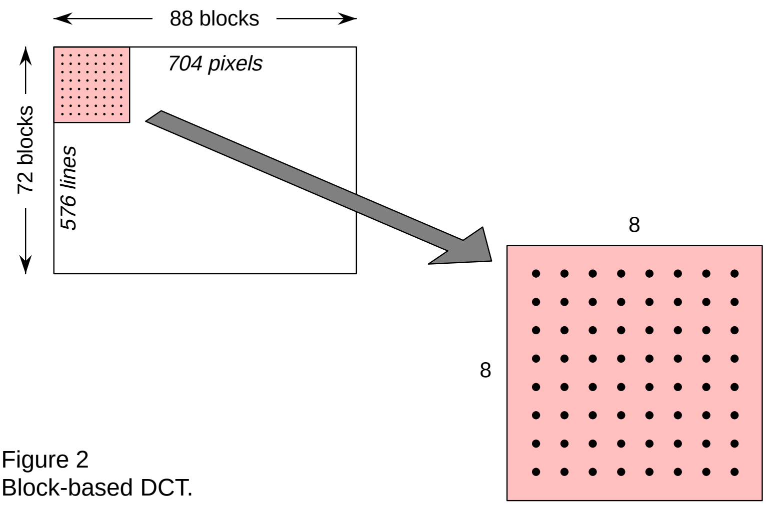


Figure 2
Block-based DCT.

3.2. Coefficient quantization

After a block has been transformed, the transform coefficients are quantized. Different quantization is applied to each coefficient depending on the spatial frequency within the block that it represents. The objective is to minimize the number of bits which must be transmitted to the decoder, so that it can perform the inverse transform and reconstruct the image: reduced quantization accuracy reduces the number of bits which need to be transmitted to represent a given DCT coefficient, but increases the possible quantization error for that coefficient. Note that the quantization noise introduced by the coder is not reversible in the decoder, so the coding and decoding process is "lossy".

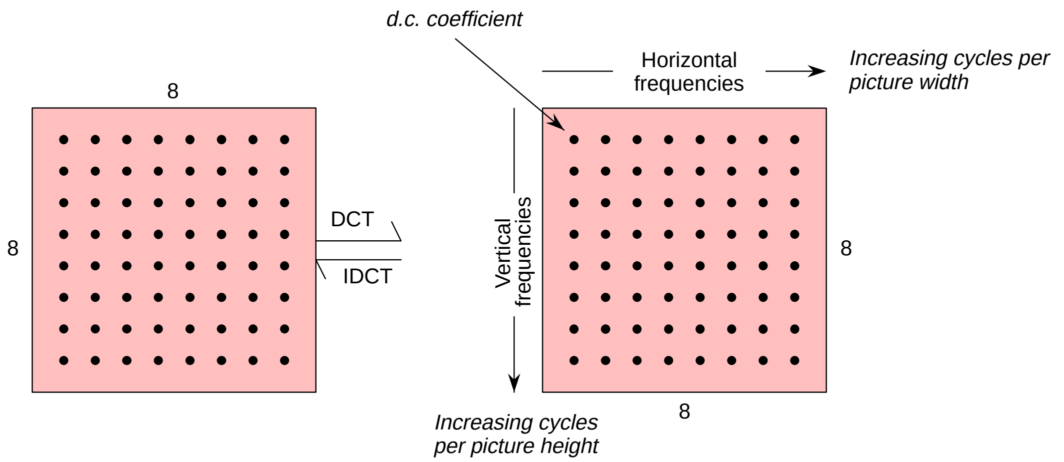


Figure 3
DCT transform pairs.

More quantization error can be tolerated in the high-frequency coefficients, because HF noise is less visible than LF quantization noise. Also, quantization noise is less visible in the chrominance components than in the luminance component. MPEG uses *weighting matrices* to define the relative accuracy of the quantization of the different coefficients. Different weighting matrices can be used for different frames, depending on the prediction mode used.

The weighted coefficients are then passed through a fixed quantization law, which is usually a linear law. However, for some prediction modes there is an increased threshold level (i.e. a dead-zone) around zero. The effect of this threshold is to maximize the number of coefficients which are quantized to zero: in practice, it is found that small deviations around zero are usually caused by noise in the signal, so suppressing these values actually "improves" the subjective picture quality.

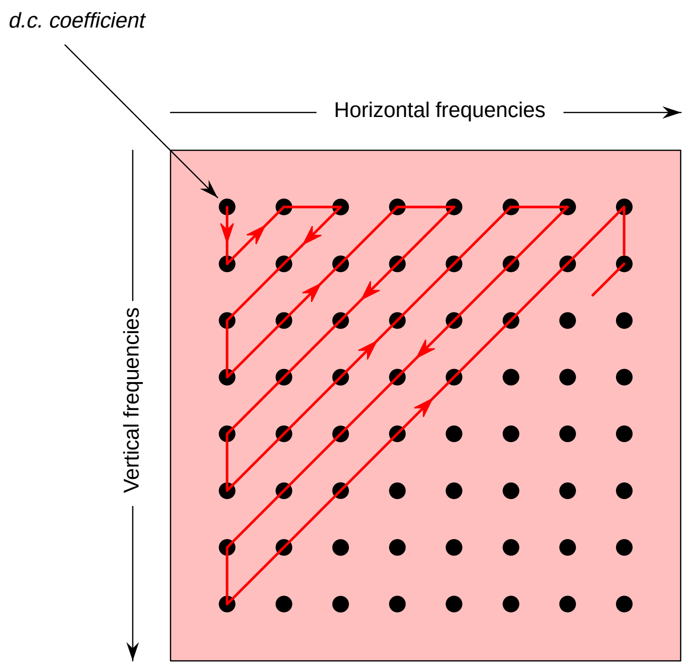
Quantization noise is more visible in some blocks than in others; for example, in blocks which contain a high-contrast edge between two plain areas. In such blocks, the quantization parameters can be modified to limit the maximum quantization error, particularly in the high-frequency coefficients.

3.3. *Zigzag coefficient scanning, run-length coding, and variable-length coding*

After quantization, the 8 x 8 blocks of DCT coefficients are scanned in a *zigzag* pattern (see *Fig. 4*) to turn the 2-D array into a serial string of quantized coefficients. Two scan patterns are defined. The first is usually preferable for picture material which has strong vertical frequency components due to, perhaps, the interlace picture structure. In this scan pattern, there is a bias to scan vertical coefficients first. In the second scan pattern, which is preferable for pictures without a strong vertical structure, there is no bias and the scan proceeds diagonally from top left to bottom right, as illustrated in *Fig. 4*. The coder indicates its choice of scan pattern to the decoder.

The strings of coefficients produced by the zigzag scanning are coded by counting the number of zero coefficients preceding a non-zero coefficient, i.e. *run-length coded*. The run-length value, and the value of the non-zero coefficient which the run of zero coefficients precedes, are then combined and coded using a *variable-length code* (VLC). The VLC exploits the fact that short runs of zeros are more likely than long ones, and small coefficients

Figure 4
Scanning of DCT blocks and run-length coding with variable-length codes (Entropy coding).



Note 1: Zigzag scanning.

Note 2: Run/amplitude coding: the run of zeros and the amplitude of the DCT coefficient are given one Variable Length Code (VLC) (Huffman Code).

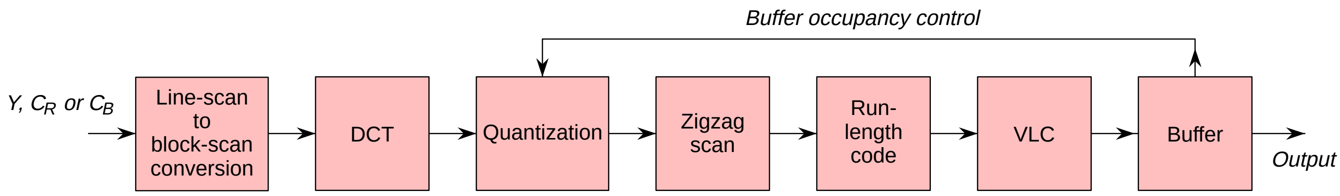


Figure 5
Basic DCT coder.

are more likely than large ones. The VLC allocates codes which have different lengths, depending upon the expected frequency of occurrence of each zero-run-length / non-zero coefficient value combination. Common combinations use short code words; less common combinations use long code words. All other combinations are coded by the combination of an escape code and two fixed-length codes, one 6-bit word to indicate the run length, and one 12-bit word to indicate the coefficient value.

One VLC code table is used in most circumstances. However, a second VLC code table is used for some special pictures. The DC coefficient is treated differently in some modes. However, all the VLCs are designed such that no complete codeword is the prefix of any other codeword: they are similar to the well-known *Huffman code*. Thus the decoder can identify where one variable-length codeword ends and another starts, when operating within the correct codebook. No VLC or combination of codes is allowed to produce a sequence of 23 contiguous zeros – this particular sequence is used for synchronization purposes.

DC coefficients in blocks contained within *intra macroblocks* (see *Section 3.7.*) are differentially encoded before variable-length coding.

3.4. Buffering and feedback

The DCT coefficient quantization, the run-length coding and the variable-length coding processes produce a varying bit-rate which depends upon the complexity of the picture information and the amount and type of motion in the picture. To produce the constant bit-rate needed for transmission over a fixed bit-rate system, a buffer is needed to smooth out the variations in bit-rate. To prevent overflow or underflow of this buffer, its occupancy is monitored and feedback is applied to the coding processes to control the input to the buffer. The DCT quantization process is often used to provide direct control of the input to the buffer: as the buffer becomes full, the quantizer is made coarser to reduce the number of bits used to code each DCT coefficient: as the buffer empties, the DCT quantization is made finer. Other means of controlling the buffer occupancy may be used as well as, or instead of, the control of DCT coefficient quantization.

Fig. 5 shows a block diagram of a basic DCT codec with, in this example, the buffer occupancy controlled by feedback to the DCT coefficient quantization.

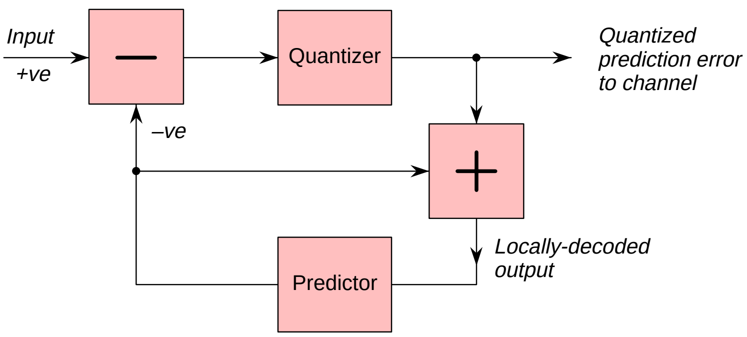
It is important to note that the final bit-rate at the output of an MPEG video encoder can be freely varied: if the output bit-rate is reduced, the buffer will empty more slowly and the coder will automatically compensate by, for example, making the DCT coefficient quantization coarser. But, of course, reducing the output bit-rate reduces the quality of the decoded pictures. There is no need to lock input sampling rates to channel bit-rates, or vice-versa.

3.5. Reduction of temporal redundancy: interframe prediction

In order to exploit the fact that pictures often change little from one frame to the next, MPEG includes temporal prediction modes: that is, we attempt to predict one frame to be coded from a previous "reference" frame.

Fig. 6 illustrates a basic differential pulse code modulation (DPCM) coder in which we quantize and transmit only the differences between the input and a prediction based on the previous locally-decoded output. Note that the prediction cannot be

Figure 6
Basic DPCM coder.



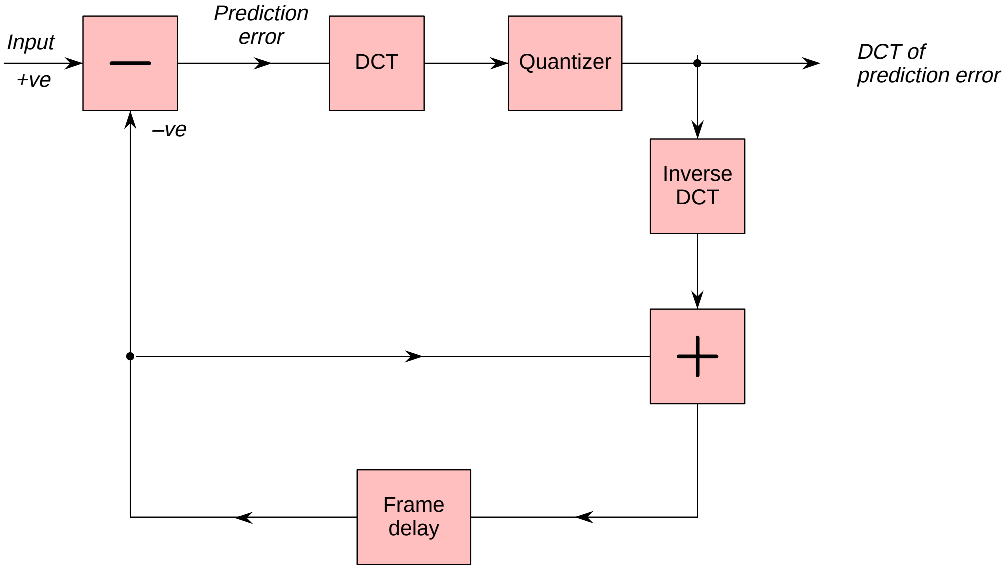


Figure 7
DCT with interframe prediction coder.

based on previous source pictures, because the prediction has to be repeatable in the decoder (where the source pictures are not available). Consequently, the coder contains a local decoder which reconstructs pictures exactly as they would be in the actual decoder. The locally-decoded output then forms the input to the predictor. In interframe prediction, samples from one frame are used in the prediction of samples in other "reference" frames.

In MPEG coding, we combine interframe prediction (which reduces temporal redundancy) with the DCT and variable-length coding tools that were described in *Section 3.3.* (which reduce spatial redundancy), as shown in *Fig. 7*. The coder subtracts the prediction from the input to form a *prediction-error* picture. The prediction error is transformed with the DCT, the coefficients are quantized, and these quantized values are coded using a VLC.

The simplest interframe prediction is to predict a block of samples from the co-sited (i.e. the same spatial position) block in the reference frame. In this case the "predictor" would comprise simply a delay of exactly one frame, as shown in *Fig. 7*. This makes a good prediction for stationary regions of the image but is poor in moving areas.

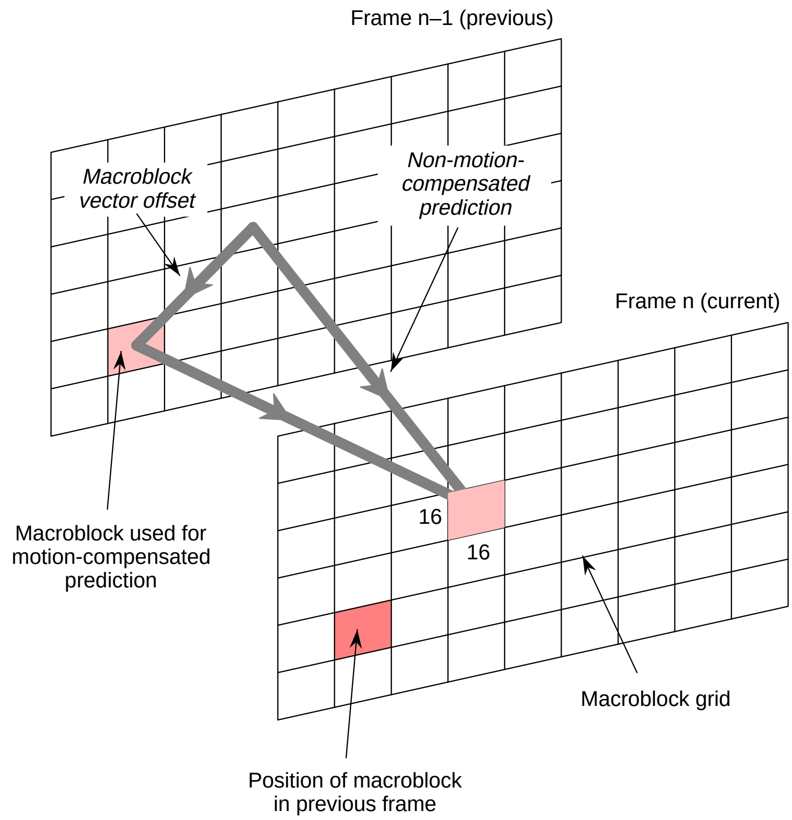
3.6. Motion-compensated interframe prediction

A more sophisticated prediction method, known as *motion-compensated interframe prediction*, offsets any translational motion which has occurred between the block being coded and the reference frame, and uses a shifted block from the reference frame as the prediction (see *Fig. 8*).

One method of determining the motion that has occurred between the block being coded and the reference frame is a "block-matching" search in which a large number of trial offsets are tested in the coder (see *Fig. 9*).

The "best" offset is selected on the basis of a measurement of the minimum error between the block being coded and the prediction. Since MPEG de-

Figure 8
Motion-compensated interframe predictiom



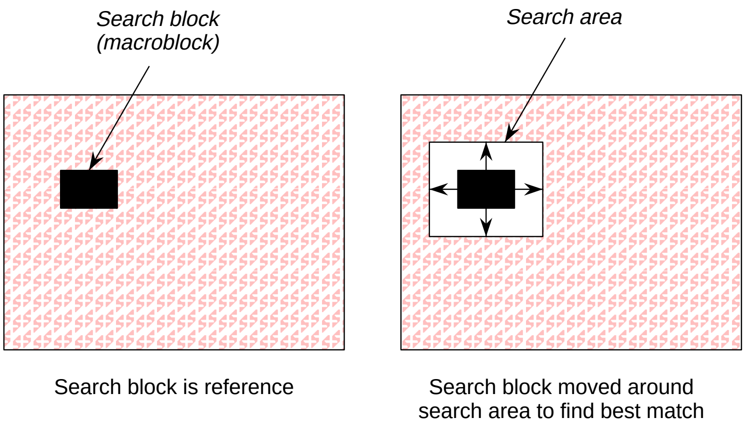


Figure 9
Principle of
block-matching
motion.

fines only the decoding process, not the coder, the choice of motion measurement algorithm is left to the designer of the coder and is an area where considerable difference in performance occurs between different algorithms and different implementations. A major requirement is to have a search area large enough to cover any motion present from frame to frame. However, increasing the size of the search area greatly increases the processing needed to find the best match: various techniques such as *hierarchical block matching* are used to try to overcome this dilemma.

Bi-directional prediction (see *Fig. 10*) consists of forming a prediction from both the previous frame and the following frame, by a linear combination of these, shifted according to suitable motion estimates.

Bi-directional prediction is particularly useful where motion uncovers areas of detail. However, to enable backward prediction from a future frame, the coder re-orders the pictures so that they are transmitted in a different order from that in which they are displayed. This process, and the process of re-ordering to the correct display sequence in the decoder, introduces considerable end-to-end processing delay which may be a problem in some applications. To overcome this, MPEG defines a profile (see *Section 4*) which does not use bi-directional prediction.

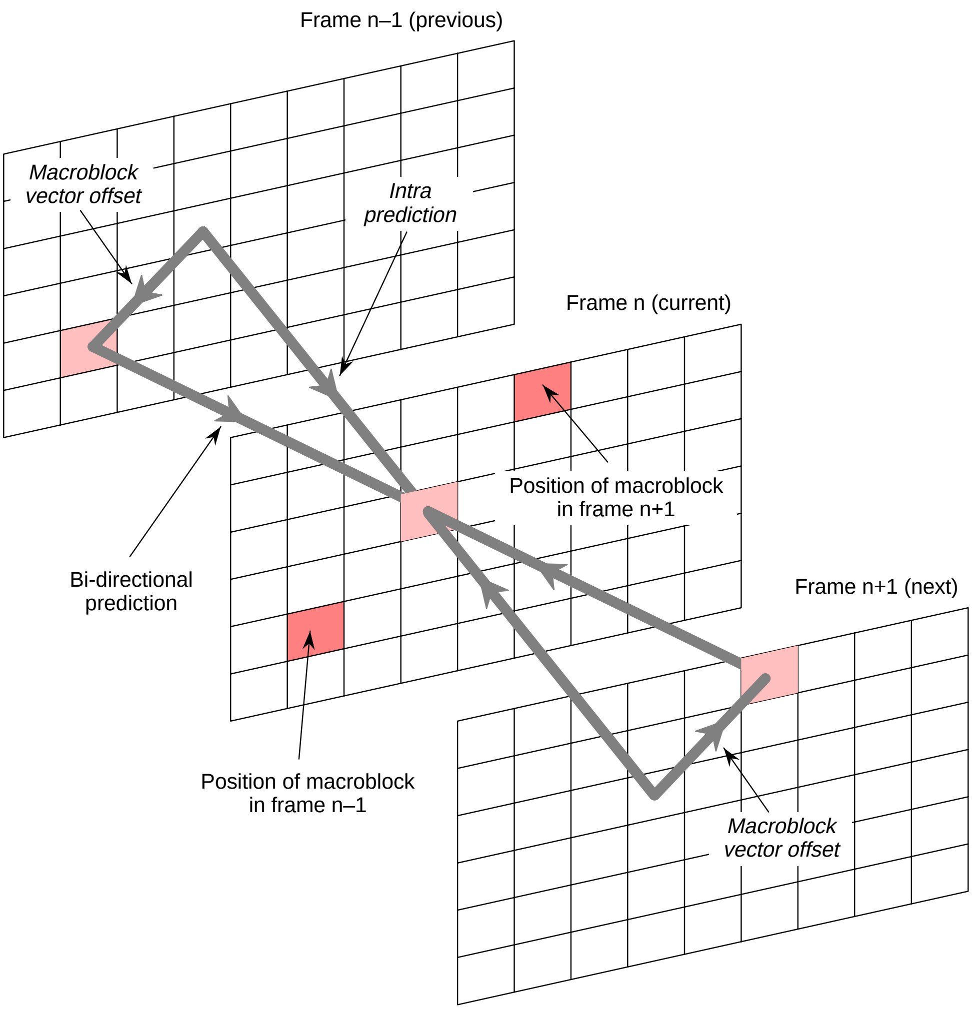


Figure 10
Motion-compensated
bi-directional
prediction.

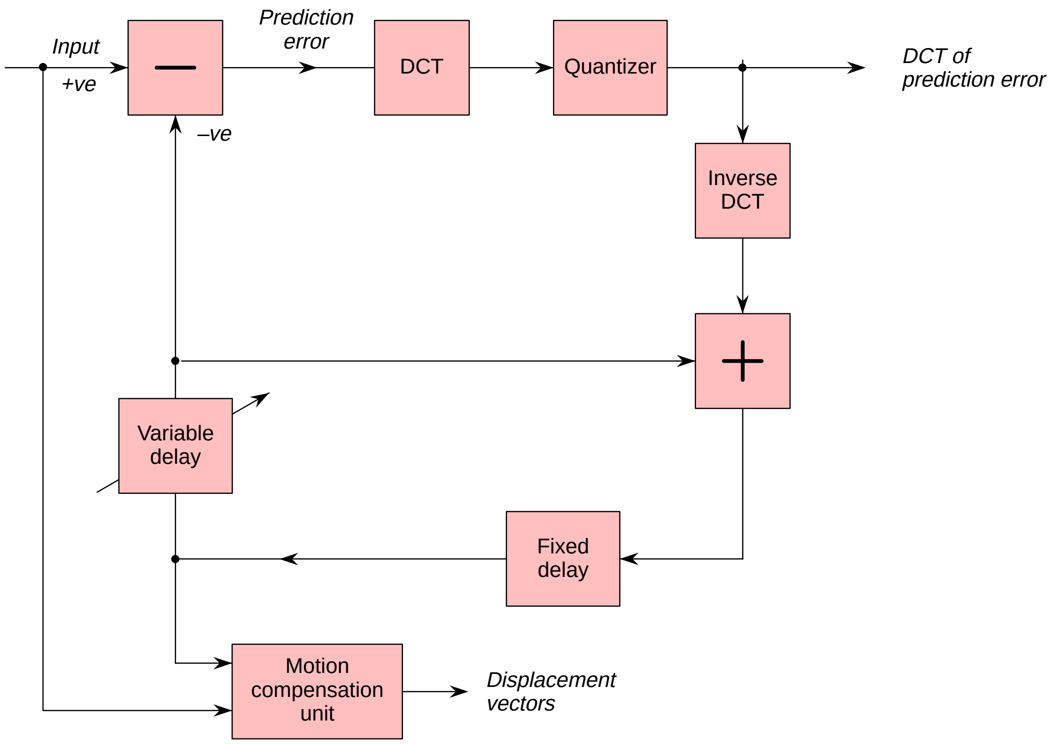


Figure 11
Motion-compensated interframe prediction DCT.

Whereas the basic coding unit for spatial redundancy reduction in MPEG is based on an 8 x 8 block, motion-compensation is usually based on a 16 pixel by 16 line *macroblock*. The size of the macroblock is a trade-off between the need to minimize the bit-rate required to transmit the motion representation (known as *motion vectors*) to the decoder, which supports the case for a large macroblock size, and the need to vary the prediction process locally within the picture content and movement, which supports the case for a small macroblock size.

To minimize the bit-rate needed to transmit the motion vectors, they are differentially-encoded with reference to previous motion vectors. The motion vector value *prediction error* is then variable-length coded using another VLC table.

Fig. 11 shows a conceptual motion-compensated inter-frame DCT coder in which, for simplicity, we illustrate implementing the process of motion-compensated prediction by suggesting a "variable delay". In practical implementations, of course, the motion-compensated prediction is implemented in other ways.

3.7. Prediction modes

In an MPEG-2 coder, the motion-compensated predictor supports many methods for generating a prediction. For example, a macroblock may be "forward predicted" from a past picture, "backward predicted" from a future picture, or "interpolated" by averaging a forward and backward prediction. Another option is to make a zero-value prediction, such that the source image block rather than the prediction error-block is DCT-coded. Such macroblocks are known as *intra-* or *I-coded*.

Although no prediction information is needed for intra-macroblocks, they can carry motion vector information. In normal circumstances, the motion vector information for an I-coded macroblock is not used, but its function is to provide a means of concealing the decoding errors when data errors in the bit-stream make it impossible to decode the data for that macroblock.

Fields of a frame may be predicted separately from their own motion vector (*field prediction coding*), or together using a common motion vector (*frame prediction coding*). Generally, in the case of image sequences where the motion is slow, frame prediction coding is more efficient. However, when motion speed increases, field prediction coding becomes more efficient.

In addition to the two basic modes of field and frame prediction, two further modes have been defined:

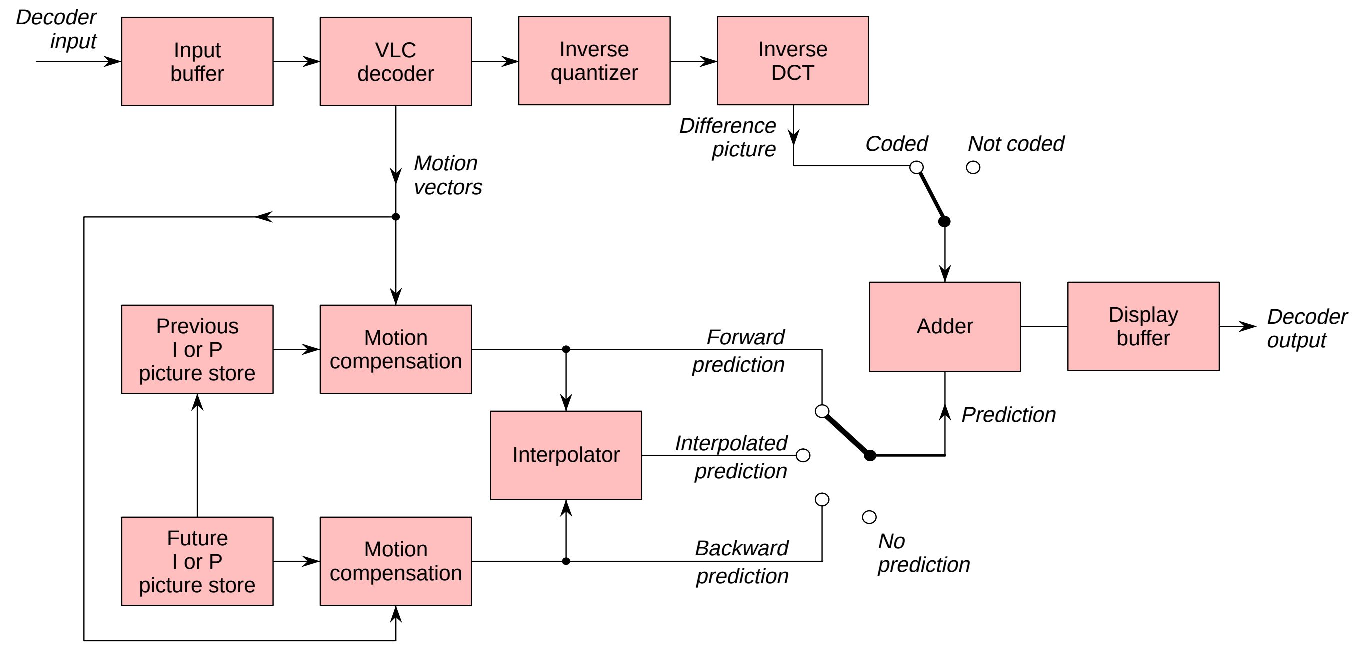


Figure 12
Decoding a "B" macroblock.

1) 16 x 8 motion compensation

This mode uses at least two motion vectors for each macroblock: one vector is used for the upper 16 x 8 region and one for the lower half. (In the case of B-pictures (see *Section 3.8*), a total of four motion vectors are used for each macroblock in this mode, since both the upper and the lower regions may each have motion vectors referring to past and future pictures.):

The 16 x 16 motion compensation mode is permitted only in field-structured pictures and is intended to allow that, in such cases, the spatial area covered by each motion vector is approximately equal to that of a 16 x 16 macroblock in a frame structure picture.

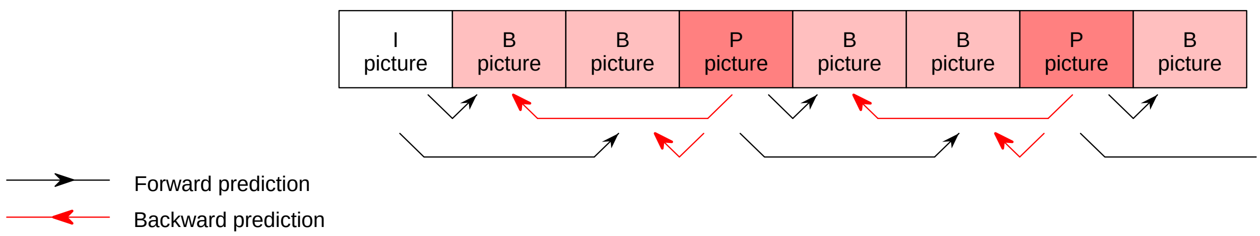
2) Dual prime mode

This mode may be used in both field- and frame-structured coding but is only permitted in P-pictures (see *Section 3.8*) when there have been no B-pictures between the P-picture and its reference frame. In this case, a motion vector and a differential-offset motion vector are transmitted.

For field pictures, two motion vectors are derived from this data and are used to form two predictions from two reference fields. These two predictions are combined to form the final prediction.

For frame pictures, this process is repeated for each of the two fields: each field is predicted separately, giving rise to a total of four field predictions which are combined to form the final two predictions. Dual prime mode is used as an alternative to bi-directional prediction, where low delay is required: it avoids the frame re-ordering needed for bi-directional prediction but achieves similar coding efficiency.

Figure 13
MPEG picture types.



Note 1: An intra-coded (I) picture is coded using information only from itself.

Note 2: Predictive-coded (P) pictures are coded with reference to a previous I or P picture.

Note 3: Bidirectionally-predictive (B) pictures are coded with reference to both the previous I or P picture and the next (future) I or P picture.

For each macroblock to be coded, the coder chooses between these prediction modes, trying to minimize the distortions on the decoded picture within the constraints of the available channel bit-rate. The choice of prediction mode is transmitted to the decoder, together with the prediction error, so that it can regenerate the correct prediction.

Fig. 12 illustrates how a bi-directionally coded macroblock (a *B-macroblock*) is decoded. The switches illustrate the various prediction modes available for such a macroblock. Note that the coder has the option not to code some macroblocks: no DCT coefficient information is transmitted for those blocks and the macroblock address counter skips to the next coded macroblock. The decoder output for the uncoded macroblocks simply comprises the predictor output.

3.8. Picture Types

In MPEG-2, three "picture types" are defined (see *Fig. 13*). The picture type defines which prediction modes may be used to code each macroblock:

1) Intra pictures (I-pictures)

These are coded without reference to other pictures. Moderate compression is achieved by reducing spatial redundancy but not temporal redundancy. They are important as they provide access points in the bit-stream where decoding can begin without reference to previous pictures.

2) Predictive pictures (P-pictures)

These are coded using motion-compensated prediction from a past I- or P-picture and may be used as a reference for further prediction. By reducing spatial and temporal redundancy, P-pictures offer increased compression compared to I-pictures.

3) Bi-directionally-predictive pictures (B-pictures)

These use both past and future I- or P-pictures for motion compensation, and offer the highest degree of compression. As noted above, to enable backward prediction from a future frame, the coder re-orders the pictures from the natural display order to a "transmission" (or "bit-stream") order so that the B-picture is transmitted after the past and future pictures which it references (see *Fig. 14*). This introduces a delay which depends upon the number of consecutive B-pictures.

3.9. Group of pictures

The different picture types typically occur in a repeating sequence termed a *Group of Pictures* or GOP. A typical GOP is illustrated in display order in *Fig. 14(a)* and in transmission order in *Fig. 14(b)*.

A regular GOP structure can be described with two parameters:

– N (the number of pictures in the GOP);

– M (the spacing of the P-pictures).

The GOP illustrated in *Fig. 14* is described as $N = 9$ and $M = 3$.

For a given decoded picture quality, coding – using each picture type – produces a different number of bits. In a typical sequence, a coded I-picture needs three times more bits than a coded P-picture, which itself occupied 50% more bits than a coded B-picture.

4. MPEG profiles and levels

MPEG-2 is intended to be generic, supporting a diverse range of applications. Different algorithmic elements or "tools", developed for many applica-

Dr. Bob Ely is an R&D manager at BBC Research and Development Department, Kingswood Warren, Surrey, UK.

Currently, he is working with the BBC's Digital Broadcasting Project which aims to investigate the technical and commercial feasibility of digital terrestrial broadcasting and to implement technical field-trials and demonstrations.

After completing his PhD in computer communications systems at Daresbury Nuclear Physics Laboratory, Bob Ely joined BBC Research Department to work on RDS and related data transmission systems. He later led the BBC team which developed the Nicam digital stereo-sound-with-television system. For many years, he was Chairman of the EBU Specialist Group on RDS, a Vice-Chairman of Working Party R and has also been a member of EBU Groups on conditional access systems.

tions, have been integrated into a single bit-stream syntax.

To implement the full syntax in all decoders is unnecessarily complex, so a small number of subsets or *profiles* of the full syntax have been defined. Also, within a given profile, a “level” is defined which describes a set of constraints such as maximum sampling density, on parameters within the profile.

The profiles defined to date fit together such that a higher profile is a superset of a lower one. A decoder which supports a particular profile and level is only required to support the corresponding subset of the full syntax and a set of parameter constraints. To restrict the number of options which must be supported, only selected combinations of profile and level are defined as *conformance points* (see *Table 1*). These are:

1) Simple profile

This uses no B-frames and, hence, no backward or interpolated prediction. Consequently, no picture re-ordering is required which makes this profile suitable for low-delay applications such as video conferencing.

2) Main profile

This adds support for B-pictures which improves the picture quality for a given bit-rate but increases the delay. Currently, most MPEG-2 video decoder chip-sets support main profile.

3) SNR profile

This adds support for enhancement layers of DCT coefficient refinement, using signal-to-noise ratio (SNR) scalability.

4) Spatial profile

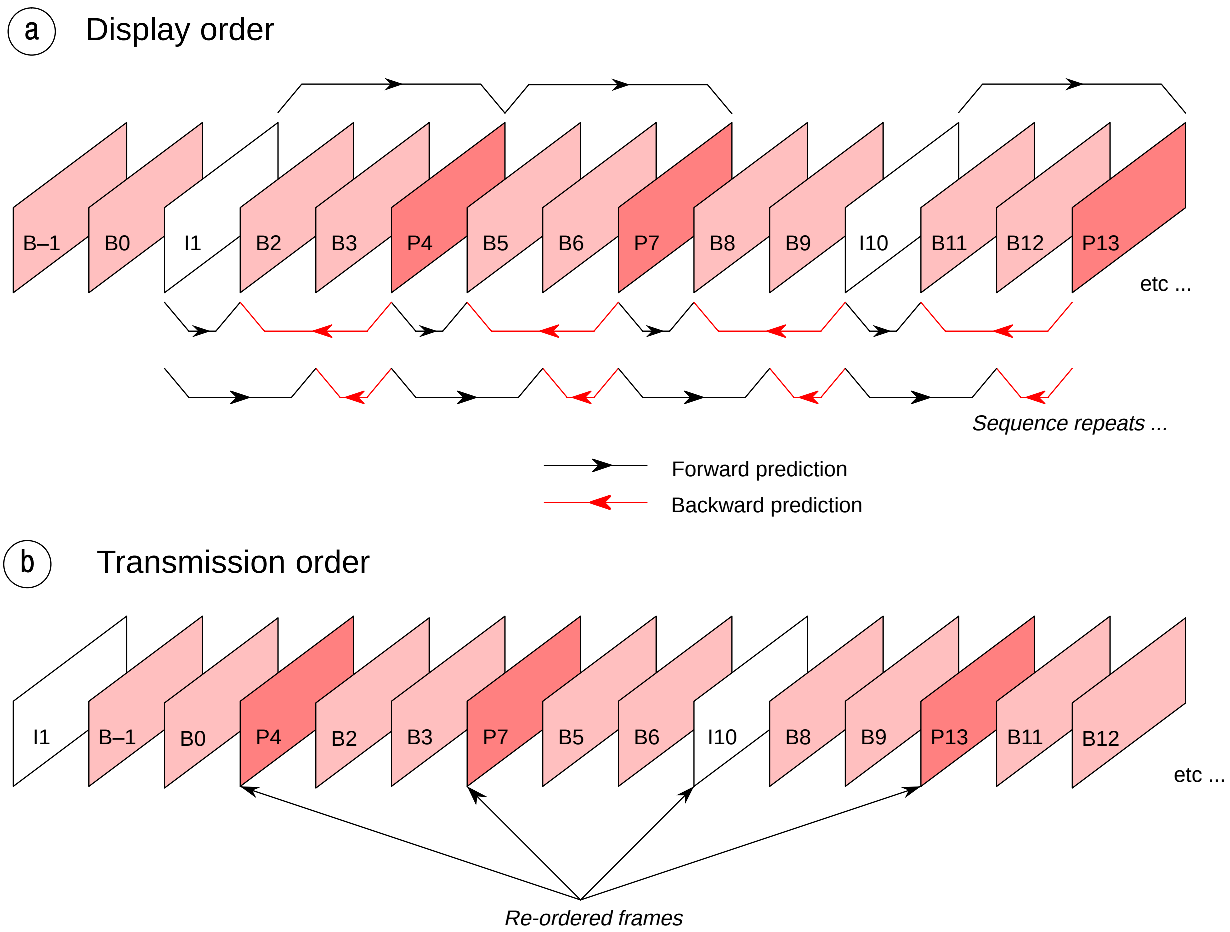
This adds support for enhancement layers carrying the image at different resolutions, using the spatial scalability tool.

5) High profile

This adds support for 4:2:2-sampled video.

All MPEG-2 decoders will also decode MPEG-1 pictures (but not vice-versa).

Figure 14
Example Group of Pictures (GOP).



	Profile and maximum total bit-rate (Mbit/s)					
	Maximum sampling density *(Hor/Vert/Freq)*	**Simple profile (SP)**	**Main profile (MP)**	**SNR profile (scalable)**	**Spacial profile (scalable)**	**High profile (HP)**
Level	**High level (HL) *(1920/1152/60)***	–	MP@HL *80 Mbit/s*	–	–	HP@HL *100 Mbit/s + lower layers*
	High–1440 *(1440/1152/60)*	–	MP@H–14 *60 Mbit/s*	–	Spt@H–14 *60 Mbit/s + lower layers*	HP@H–14 *80 Mbit/s + lower layers*
	Main level (ML) *(720/576/30)*	SP@ML *15 Mbit/s*	MP@ML *15 Mbit/s*	SNR@ML *15 Mbit/s + lower layers*	–	HP@ML *20 Mbit/s + lower layers*
	Low level (LL) *(352/280/30)*	–	MP@LL *4 Mbit/s*	SNR@LL *4 Mbit/s*	–	–
	ISO 11172 (MPEG-1) *1.856 Mbit/s*	–	–	–	–	–

Note 1: All decoders shall be able to decode ISO/IEC 11172 bitstreams.
Note 2: SP@ML decoders are required to decode MP@LL bitstreams.

Table 1
MPEG profiles and levels.

5. Conclusions

MPEG has been outstandingly successful in defining the standards for video compression coding, serving a wide range of applications, bit-rates, qualities and services. The standards are based upon a flexible toolkit of techniques of bit-rate reduction. The specification only defines the bitstream syntax and decoding process: the coding process is not specified and the performance of a coder will vary depending upon, for example, the quality of the motion-vector measurement, and the processes used for prediction mode selection.

The picture quality obtained through an MPEG codec depends strongly upon the picture content, but as experience with MPEG coding grows, the bit-rate needed for a given picture quality is likely to reduce.

Acknowledgements

The author gratefully acknowledges the contributions of his BBC colleagues Nick Wells, Phil Tudor and Paul Burfield in writing this article, which is based on published and unpublished work by these three MPEG video experts.

Bibliography

[1] ITU-T Standard H.261: **Video codec for audiovisual services at P x 64 kbit/s.**

[2] ISO/IEC 10918-1: **Information Technology – Digital compression and coding of continuous-tone still images.**

[3] ISO/IEC 11172: **Information technology coding of moving pictures and associated audio for digital storage media up to about 1.5 Mbit/s. Part 1: system.**

[4] ISO/IEC 13818-2:1995: **Information Technology – Generic coding of moving pictures and associated audio information: Video.**

[5] ITU-R Recommendation BT.601-3: **Encoding parameters of digital television for studios.**

[6] ITU-R Recommendation BT.656-1: **Interfaces for digital component video signals in 525-line and 625-line television systems operating at the 4:2:2 level of Recommendation 601.**

Further reading:

Tudor, P.: **MPEG-2: What it is and what it isn't.**
IEE Colloquium on Video Compression, January 1995.

Wells, N. et al.: **DCT source coding and current implementations for HDTV**.
EBU Technical Review No. 251, Spring 1992.

Wells, N.: **Bit-rate reduction of digital television for transmission: an introductory review.**
BBC R&D Department Report No. 1994/2.

Sandbank, C.P. (editor): **Digital Television**.
John Wiley, Chichester, 1990.
ISBN 0 471 92360 5.