Chapter 2
MPEG Video Compression Basics

B.G. Haskell and A. Puri

2.1 Video Coding Basics

Video signals differ from image signals in several important characteristics. Of course the most important difference is that video signals have a camera frame rate of anywhere from 15 to 60 frames/s, which provides the illusion of smooth motion in the displayed signal. Another difference between images and video is the ability to exploit temporal redundancy as well as spatial redundancy in designing compression methods for video. For example, we can take advantage of the fact that objects in video sequences tend to move in predictable patterns, and can therefore be motion-compensated from frame-to-frame if we can detect the object and its motion trajectory over time.

Historically, there have been five major initiatives in video coding [1–5] that have led to a range of video standards.

- Video coding for ISDN video teleconferencing, which has led to the ITU video coding standard called H.261 [6]. H.261 is also the baseline video mode for most multimedia conferencing systems.
- Video coding for low bitrate video telephony over POTS2 networks with as little as 10 kbits/s allocated to video and as little as 5.3 kbits/s allocated to voice coding, which led to the ITU video coding standard called H.263 [7]. The H.263 low bitrate video codec is used at modem rates of from 14.4 to 56 kbits/s, where the modem rate includes video coding, speech coding, control information, and other logical channels for data.

---

1 If the camera rate, chosen to portray motion, is below the display rate, chosen to avoid flicker, then some camera frames will have to be repeated.

2 Plain Old Telephone Service.

B.G. Haskell
Apple Computer, 1 Infinite Loop, Cupertino, CA 95014, USA
e-mail: BGHaskell@comcast.net
• Video coding for storing movies on CD-ROM with on the order of 1.2 Mbits/s allocated to video coding and 256 kbits/s allocated to audio coding, which led to the initial ISO MPEG-1 (Motion Picture Experts Group) standard [8].
• Video coding for broadband ISDN, broadcast and for storing video on DVD (Digital Video Disks) with on the order of 2–400 Mbits/s allocated to video and audio coding, which led to the ISO MPEG-2 video coding standard [9]. The ITU has given this standard the number H.262.
• Video coding for object-based coding at rates as low as 8 kbits/s, and as high as 1 Mbits/s, or higher, which led to the ISO MPEG-4 video coding standard [10]. Key aspects of this standard include independent coding of objects in a picture; the ability to interactively composite these objects into a scene at the display; the ability to combine graphics, animated objects, and natural objects in the scene; and finally the ability to transmit scenes in higher dimensionality formats (e.g., 3D).

Before delving into details of standards, a few general remarks are in order. It is important to note that standards specify syntax and semantics of the compressed bit stream produced by the video encoder, and how this bit stream is to be parsed and decoded (i.e., decoding procedure) to produce a decompressed video signal. However, many algorithms and parameter choices in the encoding are not specified (such as motion estimation, selection of coding modes, allocation of bits to different parts of the picture, etc.) and are left open and depend greatly on encoder implementation. However it is a requirement that resulting bit stream from encoding be compliant to the specified syntax. The result is that the quality of standards based video codecs, even at a given bitrate, depends greatly on the encoder implementation. This explains why some implementations appear to yield better video quality than others.

In the following sections, we provide brief summaries of each of these video standards, with the goal of describing the basic coding algorithms as well as the features that support use of the video coding in multimedia applications.

### 2.1.1 Basics of Interframe Video Coding

A video scene captured as a sequence of frames can be efficiently coded by estimating and compensating for motion between frames prior to generating interframe difference signal for coding. Since motion compensation is a key element in most video coders, it is worthwhile understanding the basic concepts in this processing step.

For the ease of processing, each frame of video is uniformly partitioned into smaller units called Macroblocks (MBs, formally defined a bit later) where each macroblock consists of a 16×16 block of luma, and corresponding chroma blocks.

The way that the motion estimator works is illustrated in Fig. 2.1. Each block of pixels (say 16×16 luma block of a MB) in the current frame is compared with a set of candidate blocks of same size in the previous frame to determine the one that best predicts the current block. The set of blocks includes those within a search region in previous frame centered on the position of current block in the current frame.
When the best matching block is found, a motion vector is determined, which specifies the reference block.

Figure 2.2 shows a block diagram of a motion-compensated image codec. The key idea is to combine transform coding (in the form of the Discrete Cosine Transform (DCT) of 8×8 pixel blocks) with predictive coding (in the form of...
differential Pulse Code Modulation (PCM)) in order to reduce storage and computation of the compressed image, and at the same time to give a high degree of compression and adaptability.

Since motion compensation is difficult to perform in the transform domain, the first step in the interframe coder is to create a motion compensated prediction error in the pixel domain. For each block of current frame, a prediction block in the reference frame is found using motion vector found during motion estimation, and differentiated to generate prediction error signal. This computation requires only a single frame store in the encoder and decoder. The resulting error signal is transformed using 2D DCT, quantized by an adaptive quantizer, entropy encoded using a Variable Length Coder (VLC) and buffered for transmission over a fixed rate channel.

We now discuss how various MPEG standards are built using principles and building blocks discussed so far.

### 2.2 The MPEG-1 Video Coding Standard

The MPEG-1 standard is the first true multimedia standard with specifications for coding, compression, and transmission of audio, video, and data streams in a series of synchronized, mixed Packets. The driving focus of the standard was storage of multimedia content on a standard CDROM, which supported data transfer rates of 1.4 Mb/s and a total storage capability of about 600 MB. MPEG-1 was intended to provide VHS VCR-like video and audio quality, along with VCR-like controls. MPEG-1 is formally called ISO/IEC 11172.

#### 2.2.1 Requirements of the MPEG-1 Video Standard

Uncompressed digital video of full component TV resolution requires a very high transmission bandwidth, while VHS VCR-grade equivalent raw digital video requires transmission bandwidth of around 30 Mbits/s, with compression still necessary to reduce the bit-rate to suit most applications. The required degree of compression is achieved by exploiting the spatial and temporal redundancy present in a video signal. However, the compression process is inherently lossy, and the signal reconstructed from the compressed bit stream is not identical to the input video signal. Compression typically introduces some artifacts into the decoded signal.

The primary requirement of the MPEG-1 video standard was that it should achieve the high quality of the decoded motion video at a given bit-rate. In addition to picture quality under normal play conditions, different applications have additional requirements. For instance, multimedia applications may require the ability to randomly access and decode any single video picture in the bitstream. Also, the ability to perform fast

---

3 Frames and pictures are synonymous in MPEG-1.
search directly on the bit stream, both forward and backward, is extremely desirable if the storage medium has “seek” capabilities. It is also useful to be able to edit compressed bit streams directly while maintaining decodability. And finally, a variety of video formats were needed to be supported.

2.2.2 H.261 Coding Concepts as Applicable to MPEG-1 Video

The H.261 standard employs interframe video coding that was described earlier. H.261 codes video frames using a DCT on blocks of size 8×8 pixels, much the same as used for the original JPEG coder for still images. An initial frame (called an INTRA frame) is coded and transmitted as an independent frame. Subsequent frames, which are modeled as changing slowly due to small motions of objects in the scene, are coded efficiently in the INTER mode using a technique called Motion Compensation (MC) in which the displacement of groups of pixels from their position in the previous frame (as represented by so-called motion vectors) are transmitted together with the DCT coded difference between the predicted and original images.

2.2.2.1 H.261 Bitstream Data Hierarchy

We will first explain briefly the data structure in an H.261 video bit stream and then the functional elements in an H.261 decoder.

Only two picture formats, common intermediate format (CIF) and quarter-CIF (QCIF), are allowed. CIF pictures are made of three components: luminance Y and color differences Cb and Cr, as defined in ITU-R Recommendation BT601. The CIF picture size for Y is 352 pels\(^4\) per line by 288 lines per frame. The two color difference signals are subsampled to 176 pels per line and 144 lines per frame. The image aspect ratio is 4(horizontal):3(vertical), and the picture rate is 29.97 non-interlaced frames per second. All H.261 standard codecs must be able to operate with QCIF; CIF is optional. A picture frame is partitioned into 8 line × 8 pel image blocks. A Macroblock (MB) is defined as four 8×8 (or one 16×16) Y block/s, one Cb block, and one Cr block at the same location.

The compressed H.261 video bit stream contains several layers. They are picture layer, group of blocks (GOB) layer, Macroblock (MB) layer, and block layer. The higher layer consists of its own header followed by a number of lower layers.

Picture Layer

In a compressed video bit stream, we start with the picture layer. Its header contains: Picture start code (PSC) a 20-bit pattern.

\(^4\) Abbreviation of pixel.
Temporal reference (TR) a 5-bit input frame number.
Type information (PTYPE) such as CIF/QCIF selection.
Spare bits to be defined in later versions.

GOB Layer

At the GOB layer, a GOB header contains:

- Group of blocks start code (GBSC) a 16-bit pattern.
- Group number (GN) a 4-bit GOB address.
- Quantizer information (GQUANT) initial quantizer step size normalized to the range 1–31. At the start of a GOB, we set QUANT=GQUANT.
- Spare bits to be defined in later versions of the standard.

Next, comes the MB layer. An 11-bit stuffing pattern can be inserted repetitively right after a GOB header or after a transmitted Macroblock.

Macroblock (MB) Layer

At the MB layer, the header contains:

- Macroblock address (MBA) location of this MB relative to the previously coded MB inside the GOB. MBA equals one plus the number of skipped MBs preceding the current MB in the GOB.
- Type information (MTYPE) 10 types in total.
- Quantizer (MQUANT) normalized quantizer step size to be used until the next MQUANT or GQUANT. If MQUANT is received we set QUANT=MQUANT. Range is 1–31.
- Motion vector data (MVD) differential displacement vector.
- Coded block pattern (CBP) indicates which blocks in the MB are coded.
  Blocks not coded are assumed to contain all zero coefficients.

Block Layer

The lowest layer is the block layer, consisting of quantized transform coefficients (TCOEFF), followed by the end of block (EOB) symbol. All coded blocks have the EOB symbol.

Not all header information need be present. For example, at the MB layer, if an MB is not Inter motion-compensated (as indicated by MTYPE), MVD does not exist. Also, MQUANT is optional. Most of the header information is coded using Variable Length Codewords.

There are essentially four types of coded MBs as indicated by MTYPE:

- Intra – original pels are transform-coded.
- Inter – frame difference pels (with zero-motion vectors) are coded. Skipped MBs are considered inter by default.
• Inter_MC – displaced (nonzero-motion vectors) frame differences are coded.
• Inter_MC_with_filter – the displaced blocks are filtered by a predefined loop filter, which may help reduce visible coding artifacts at very low bit rates.

2.2.2.2  H.261 Coding Semantics

A single-motion vector (horizontal and vertical displacement) is transmitted for one Inter_MC MB. That is, the four Y blocks, one Cb, and one Cr block all share the same motion vector. The range of motion vectors is $\pm 15$ Y pels with integer values. For color blocks, the motion vector is obtained by halving the transmitted vector and truncating the magnitude to an integer value.

Motion vectors are differentially coded using, in most cases, the motion vector of the MB to the left as a prediction. Zero is used as a prediction for the leftmost MBs of the GOB, and also if the MB to the left has no motion vector.

The transform coefficients of either the original (Intra) or the differential (Inter) pels are ordered according to a zigzag scanning pattern. These transform coefficients are selected and quantized at the encoder, and then coded using variable-length codewords (VLCs) and/or fixed-length codewords (FLC), depending on the values. Just as with JPEG, successive zeros between two nonzero coefficients are counted and called a RUN. The value of a transmitted nonzero quantized coefficient is called a LEVEL. The most likely occurring combinations of (RUN, LEVEL) are encoded with a VLC, with the sign bit terminating the RUN-LEVEL VLC codeword.

The standard requires a compatible IDCT (inverse DCT) to be close to the ideal 64-bit floating point IDCT. H.261 specifies a measuring process for checking a valid IDCT. The error in pel values between the ideal IDCT and the IDCT under test must be less than certain allowable limits given in the standard, e.g., peak error $\leq 1$, mean error $\leq 0.0015$, and mean square error $\leq 0.02$.

A few other items are also required by the standard. One of them is the image-block updating rate. To prevent mismatched IDCT error as well as channel error propagation, every MB should be intra-coded at least once in every 132 transmitted picture frames.

The contents of the transmitted video bit stream must also meet the requirements of the hypothetical reference decoder (HRD). For CIF pictures, every coded frame is limited to fewer than 256 Kbits; for QCIF, the limit is 64 Kbits, where $K = 1,024$. The HRD receiving buffer size is $B + 256$ Kbits, where $B = 4 \times R_{\text{max}} / 29.97$ and $R_{\text{max}}$ is the maximum connection (channel) rate. At every picture interval ($1/29.97$ s), the HRD buffer is examined. If at least one complete coded picture is in the buffer, then the earliest picture bits are removed from the buffer and decoded. The buffer occupancy, right after the above bits have been removed, must be less than $B$.

2.2.3  MPEG-1 Video Coding

Video coding as per MPEG-1 uses coding concepts similar to H.261 just described, namely spatial coding by taking the DCT of $8 \times 8$ pixel blocks, quantizing the DCT
coefficients based on perceptual weighting criteria, storing the DCT coefficients for each block in a zigzag scan, and doing a variable run length coding of the resulting DCT coefficient stream. Temporal coding is achieved by using the ideas of uni- and bi-directional motion compensated prediction, with three types of pictures resulting, namely:

- \( I \) or Intra pictures which were coded independently of all previous or future pictures.
- \( P \) or Predictive pictures which were coded based on previous \( I \) or previous \( P \) pictures.
- \( B \) or Bi-directionally predictive pictures which were coded based on either the next and/or the previous pictures.

If video is coded at about 1.1 Mbits/s and stereo audio is coded at 128 kbits/s per channel, then the total audio/video digital signal will fit onto the CD-ROM bit-rate of approximately 1.4 Mbits/s as well as the North American ISDN Primary Rate (23 B-channels) of 1.47 Mbits/s. The specified bit-rate of 1.5 Mbits/s is not a hard upper limit. In fact, MPEG-1 allows rates as high as 100 Mbits/s. However, during the course of MPEG-1 algorithm development, coded image quality was optimized at a rate of 1.1 Mbits/s using progressive (NonInterlaced) scanned pictures.

Two Source Input Formats (SIF) were used for optimization. One corresponding to NTSC was 352 pels, 240 lines, 29.97 frames/s. The other corresponding to PAL, was 352 pels, 288 lines, 25 frames/s. SIF uses 2:1 color subsampling, both horizontally and vertically, in the same 4:2:0 format as H.261.

### 2.2.3.1 Basics of MPEG-1 Video Compression

Both spatial and temporal redundancy reduction are needed for the high compression requirements of MPEG-1. Most techniques used by MPEG-1 have been described earlier.

Exploiting Spatial Redundancy

The compression approach of MPEG-1 video combines elements of JPEG, elements of H.261, and significant new elements that allow not only higher compression but also frequent entry points into the video stream.

Because video is a sequence of still images, it is possible to achieve some compression using techniques similar to JPEG. Such methods of compression are called intraframe coding techniques, where each picture of video is individually and independently compressed or encoded. Intraframe coding exploits the spatial redundancy that exists between adjacent pels of a picture. Pictures coded using only intraframe coding are called \( I \)-pictures.

As in JPEG and H.261, the MPEG-1 video-coding algorithm employs a block-based two-dimensional DCT. A picture is first divided into \( 8 \times 8 \) blocks of pels, and the two-dimensional DCT is then applied independently on each block. This operation
results in an $8 \times 8$ block of DCT coefficients in which most of the energy in the original (pel) block is typically concentrated in a few low-frequency coefficients. The coefficients are scanned and transmitted in the same zigzag order as JPEG and H.261.

A quantizer is applied to the DCT coefficients, which sets many of them to zero. This quantization is responsible for the lossy nature of the compression algorithms in JPEG, H.261 and MPEG-1 video. Compression is achieved by transmitting only the coefficients that survive the quantization operation and by entropy-coding their locations and amplitudes.

**Exploiting Temporal Redundancy**

Many of the interactive requirements can be satisfied by intraframe coding. However, as in H.261, the quality achieved by intraframe coding alone is not sufficient for typical video signals at bit-rates around 1.1 Mbits/s.

Temporal redundancy results from a high degree of correlation between adjacent pictures. The MPEG-1 algorithm exploits this redundancy by computing an interframe difference signal called the prediction error. In computing the prediction error, the technique of motion compensation is employed to correct for motion. A Macroblock (MB) approach is adopted for motion compensation.

In unidirectional or *Forward Prediction*, $16 \times 16$ luma block of each macroblock in the current picture to be coded is matched with a block of the same size in a previous picture called the Reference picture. As in H.261 blocks of the Reference picture that “best match” the $16 \times 16$ luma blocks of current picture, are called the *Prediction* blocks. The prediction error is then computed as the difference between the Target block and the Prediction block. The position of this best-matching Prediction block is indicated by a motion vector that describes the displacement between it and the Target block. Unlike H.261 where each motion vector is specified at “integer pel” accuracy, in MPEG-1 each motion vector is specified at “half-pel” accuracy, thus allowing improved prediction. The motion vector information is also encoded and transmitted along with the prediction error. Pictures coded using Forward Prediction are called *P-pictures*.

The prediction error itself is transmitted using the DCT-based intraframe encoding technique summarized above. In MPEG-1 video (as in H.261), motion compensation is performed on MBs ($16 \times 16$ luma and associated chroma), representing a reasonable trade-off between the compression provided by motion compensation and the cost associated with transmitting the motion vectors.

**Bidirectional Temporal Prediction**

Bidirectional temporal prediction, also called Motion-Compensated Interpolation, is a key feature of MPEG-1 video. Pictures coded with Bidirectional prediction use...

---

5 Prediction $16 \times 16$ blocks do not, in general, align with coded $16 \times 16$ luma (of MB) boundaries in the Reference frame.
two Reference pictures, one in the past and one in the future. A Target 16×16 luma block in bidirectionally coded pictures can be predicted by a 16×16 block from the past Reference picture (Forward Prediction), or one from the future Reference picture (Backward Prediction), or by an average of two 16×16 luma blocks, one from each Reference picture (Interpolation). In every case, a Prediction 16×16 block from a Reference picture is associated with a motion vector, so that up to two motion vectors per macroblock may be used with Bidirectional prediction. As in the case of unidirectional prediction, motion vectors are represented at “half-pel” accuracy. Motion-Compensated Interpolation for a 16×16 block in a Bidirectionally predicted “current” frame is illustrated in Fig. 2.3.

Pictures coded using Bidirectional Prediction are called B-pictures. Pictures that are Bidirectionally predicted are never themselves used as Reference pictures, i.e., Reference pictures for B-pictures must be either P-pictures or I-pictures. Similarly, Reference pictures for P-pictures must also be either P-pictures or I-pictures.

Bidirectional prediction provides a number of advantages. The primary one is that the compression obtained is typically higher than can be obtained from Forward (unidirectional) prediction alone. To obtain the same picture quality, Bidirectionally predicted pictures can be encoded with fewer bits than pictures using only Forward prediction.

However, Bidirectional prediction does introduce extra delay in the encoding process, because pictures must be encoded out of sequence. Further, it entails extra encoding complexity because block matching (the most computationally intensive encoding procedure) has to be performed twice for each Target block, once with the past Reference picture and once with the future Reference picture.

2.2.3.2 MPEG-1 Bitstream Data Hierarchy

The MPEG-1 video standard specifies the syntax and semantics of the compressed bit stream produced by the video encoder. The standard also specifies how this bit stream is to be parsed and decoded to produce a decompressed video signal.
The details of the motion estimation matching procedure are not part of the standard. However, as with H.261 there is a strong limitation on the variation in bits/picture in the case of constant bit-rate operation. This is enforced through a Video Buffer Verifier (VBV), which corresponds to the Hypothetical Reference Decoder of H.261. Any MPEG-1 bit stream is prohibited from overflowing or underflowing the buffer of this VBV. Thus, unlike H.261, there is no picture skipping allowed in MPEG-1.

The bit-stream syntax is flexible in order to support the variety of applications envisaged for the MPEG-1 video standard. To this end, the overall syntax is constructed in a hierarchy\textsuperscript{6} of several Headers, each performing a different logical function.

Video Sequence Header

The outermost Header is called the Video Sequence Header, which contains basic parameters such as the size of the video pictures, Pel Aspect Ratio (PAR), picture rate, bit-rate, assumed VBV buffer size and certain other global parameters. This Header also allows for the optional transmission of JPEG style Quantizer Matrices, one for Intra coded pictures and one for Non-Intra coded pictures. Unlike JPEG, if one or both quantizer matrices are not sent, default values are defined. Private user data can also be sent in the Sequence Header as long as it does not contain a Start Code Header, which MPEG-1 defines as a string of 23 or more zeros.

Group of Pictures (GOP) Header

Below the Video Sequence Header is the Group of Pictures (GOP) Header, which provides support for random access, fast search, and editing. A sequence of transmitted video pictures is divided into a series of GOPs, where each GOP contains an intra-coded picture (I-picture) followed by an arrangement of Forward predictive-coded pictures (P-pictures) and Bidirectionally predicted pictures (B-pictures).

Figure 2.4 shows a GOP example with six pictures, 1–6. This GOP contains I-picture 1, P-pictures 4 and 6, and B-pictures 2, 3 and 5. The encoding/transmission order of the pictures in this GOP is shown at the bottom of Fig. 2.4. B-pictures 2 and 3 are encoded after P-picture 4, using P-picture 4 and I-picture 1 as reference. Note that B-picture 7 in Fig. 2.4 is part of the next GOP because it is encoded after I-picture 8.

Random access and fast search are enabled by the availability of the I-pictures, which can be decoded independently and serve as starting points for further decoding. The MPEG-1 video standard allows GOPs to be of arbitrary structure and length.

\textsuperscript{6}As in H.261, MPEG-1 uses the term Layers for this hierarchy. However, Layer has another meaning in MPEG-2. Thus, to avoid confusion we will not use Layers in this section.
Picture Header

Below the GOP is the Picture Header, which contains the type of picture that is present, e.g., I, P or B, as well as a Temporal Reference indicating the position of the picture in display order within the GOP. It also contains a parameter called vBV_delay that indicates how long to wait after a random access before starting to decode. Without this information, a decoder buffer could underflow or overflow following a random access.

Slice Header

A Slice is a string of consecutive Macroblocks of arbitrary length running from left to right and top to bottom across the picture. The Slice Header is intended to be used for re-synchronization in the event of transmission bit errors. Prediction registers used in the differential encoding of motion vectors and DC Intra coefficients are reset at the start of a Slice. It is again the responsibility of the encoder to choose the length of each Slice depending on the expected bit error conditions. The first and last MBs of a Slice cannot be skipped MBs, and gaps are not allowed between Slices. The Slice Header contains the vertical position of the Slice within the picture, as well as a quantizer_scale parameter (corresponding to GQUANT in H.261).

Macroblock Header

The Macroblock (MB) is the 16×16 motion compensation unit. In the Macroblock Header, the horizontal position (in MBs) of the first MB of each Slice is coded with the MB Address VLC. The positions of additional transmitted MBs are coded
differentially with respect to the most recently transmitted MB, also using the MB Address VLC. Skipped MBs are not allowed in I-pictures. In P-pictures, skipped MBs are assumed NonIntra with zero coefficients and zero motion vectors. In B-pictures, skipped MBs are assumed NonIntra with zero coefficients and motion vectors the same as the previous MB. Also included in the Macroblock Header are MB Type (Intra, NonIntra, etc.), quantizer_scale (corresponding to MQUANT in H.261), motion vectors and coded block pattern. As with other Headers, these parameters may or may not be present, depending on MB Type.

Block

A Block consists of the data for the quantized DCT coefficients of an 8×8 Block in the Macroblock. It is VLC coded as described in the next sections. For noncoded Blocks, the DCT coefficients are assumed to be zero.

2.2.3.3 MPEG-1 Video Encoding

Figure 2.5 shows a typical MPEG-1 video encoder. It is assumed that frame reordering takes place before coding, i.e., I- or P-pictures used for B-picture prediction must be coded and transmitted before any of the corresponding B-pictures.
Input video is fed to a Motion Compensation Estimator/Predictor that feeds a prediction to the minus input of the Subtractor. For each MB, the Inter/Intra Classifier then compares the input pels with the prediction error output of the Subtractor. Typically, if the mean square prediction error exceeds the mean square pel value, an Intra MB is decided. More complicated comparisons involving DCT of both the pels and the prediction error yield somewhat better performance, but are not usually deemed worth the cost.

For Intra MBs the prediction is set to zero. Otherwise, it comes from the Predictor, as described above. The prediction error is then passed through the DCT and Quantizer before being coded, multiplexed and sent to the Buffer. Quantized Levels are converted to reconstructed DCT coefficients by the Inverse Quantizer and then inverse transformed by the IDCT to produce a coded prediction error. The Adder adds the prediction to the prediction error and clips the result to the range 0–255 to produce coded pel values.

For B-pictures the Motion Compensation Estimator/Predictor uses both the Previous Picture and the Future Picture. These are kept in picture stores and remain unchanged during B-picture coding. Thus, in fact, the Inverse Quantizer, IDCT and Adder may be disabled during B-picture coding.

For I and P-pictures the coded pels output by the Adder are written to the Future Picture Store, while at the same time the old pels are copied from the Future Picture Store to the Previous Picture Store. In practice this is usually accomplished by a simple change of memory addresses.

The Coding Statistics Processor in conjunction with the Quantizer Adapter control the output bit-rate in order to conform to the Video Buffer Verifier (VBV) and to optimize the picture quality as much as possible. A simple control that works reasonably well is to define a target buffer fullness for each picture in the GOP. For each picture the quantizer_scale value is then adjusted periodically to try to make the actual buffer fullness meet the assigned value. More complicated controls would, in addition, exploit spatio-temporal masking in choosing the quantizer_scale parameter for each MB.

### 2.2.3.4 MPEG-1 Video Decoding

Figure 2.6 shows a typical MPEG-1 video decoder. It is basically identical to the pel reconstruction portion of the encoder. It is assumed that frame reordering takes place after decoding and video output. However, extra memory for reordering can often be avoided if during a write of the Previous Picture Store, the pels are routed also to the display.

The decoder cannot tell the size of the GOP from the bitstream parameters. Indeed it does not know until the Picture Header whether the picture is I-, P- or B-type. This could present problems in synchronizing audio and video were it not for the Systems part of MPEG-1, which provides Time Stamps for audio, video and ancillary data. By presenting decoded information at the proper time as indicated by the Time Stamps, synchronization is assured.
Decoders often must accommodate occasional transmission bit errors. Typically, when errors are detected either through external means or internally through the arrival of illegal data, the decoder replaces the data with skipped MBs until the next Slice is detected. The visibility of errors is then rather low unless they occur in an I-picture, in which case they may be very visible throughout the GOB. A cure for this problem was developed for MPEG-2 and will be described later.

### 2.2.4 MPEG-1 Capabilities and Interoperability

In demonstrations of MPEG-1 video at a bit-rate of 1.1 Mbits/s, SIF resolution pictures have been used. This resolution is roughly equivalent to one field of an interlaced NTSC or PAL picture. The quality achieved by the MPEG-1 video encoder at this bit-rate has often been compared to that of VHS\(^7\) videotape playback.

Although the MPEG-1 video standard was originally intended for operation in the neighborhood of the above bit-rate, a much wider range of resolution and bit-rates is supported by the syntax. The MPEG-1 video standard thus provides a generic bit-stream syntax that can be used for a variety of applications. MPEG-1 video (Part 2 of ISO/IEC 11172) provides all the details of the syntax, complete with informative sections on encoder procedures that are outside the scope of the standard.

---

\(^7\) VHS, an abbreviation for Video Home System, is a registered trademark of the Victor Company of Japan.
To promote interoperability between MPEG-1 applications (bitstreams) and decoders, MPEG introduced the concept of constrained parameters; the following parameters were specified.

Horizontal size <= 768 pels, vertical size of <=576 lines, Picture rate <=30
Number of MBs/pic <= 396, number of MBs/s <=396×25, Bitrate <=4,640

2.3 The MPEG-2 Video Coding Standard

The MPEG-2 standard was designed to provide the capability for compressing, coding, and transmitting high quality, multi-channel, multimedia signals over broadband networks, for example using ATM (Asynchronous Transmission Mode) protocols. The MPEG-2 standard specifies the requirements for video coding, audio coding, systems coding for combining coded audio and video with user-defined private data streams, conformance testing to verify that bitstreams and decoders meet the requirements, and software simulation for encoding and decoding of both the program and the transport streams. Because MPEG-2 was designed as a transmission standard, it supports a variety of Packet formats (including long and variable length Packets from 1 kB up to 64 kB), and provides error correction capability that is suitable for transmission over cable TV and satellite links.

MPEG-2 video was aimed at video bit-rates above 2 Mbits/s. Specifically, it was originally designed for high quality encoding of interlaced video from standard TV with bitrates on the order of 4–9 Mb/s. As it evolved, however, MPEG-2 video was expanded to include high resolution video, such as High Definition TV (HDTV). MPEG-2 video was also extended to include hierarchical or scalable coding. The official name of MPEG-2 is ISO/IEC 13818.

The original objective of MPEG-2 was to code interlaced BT601 video at a bit-rate that would serve a large number of consumer applications, and in fact one of the main differences between MPEG-1 and MPEG-2 is that MPEG-2 handles interlace efficiently. Since the picture resolution of BT601 is about four times that of the SIF of MPEG-1, the bit-rate chosen for MPEG-2 optimization were 4, and 9 Mbits/s. However, MPEG-2 allows much higher bitrates.

A bit-rate of 4 Mbits/s was deemed too low to enable high quality transmission of every BT601 color sample. Thus, an MPEG-2 4:2:0 format was defined to allow for 2:1 vertical subsampling of the color, in addition to the normal 2:1 horizontal color subsampling of BT601. Pel positions are only slightly different than the CIF of H.261 and the SIF of MPEG-1.

For interlace, the temporal integrity of the chrominance samples must be maintained. Thus, MPEG-2 normally defines the first, third, etc. rows of 4:2:0 chrominance CbCr samples to be temporally the same as the first, third, etc. rows of luminance Y samples. The second, fourth, etc. rows of chrominance CbCr samples are temporally the same as the second, fourth, etc. rows of luminance Y samples. However, an override capability is available to indicate that the 4:2:0 chrominance samples are all temporally the same as the temporally first field of the frame.
At higher bit-rates the full 4:2:2 color format of BT601 may be used, in which the first luminance and chrominance samples of each line are co-sited. MPEG-2 also allows for a 4:4:4 color format.

2.3.1 Requirements of the MPEG-2 Video Standard

The primary requirement of the MPEG-2 video standard was that it should achieve high compression of interlaced video while maintaining high video quality. In addition to picture quality, different applications have additional requirements even beyond those provided by MPEG-1. For instance, multipoint network communications may require the ability to communicate simultaneously with SIF and BT601 decoders. Communication over packet networks may require prioritization so that the network can drop low priority packets in case of congestion. Broadcasters may wish to send the same program to BT601 decoders as well as to progressive scanned HDTV decoders. In order to satisfy all these requirements MPEG-2 has defined a large number of capabilities.

However, not all applications require all the features of MPEG-2. Thus, to promote interoperability amongst applications, MPEG-2 introduced several sets of algorithmic choices (Profiles) and choice of constrained parameters (Levels) to enhance interoperability.

2.3.2 MPEG-2 Video Coding

As with MPEG-1, both spatial and temporal redundancy reduction are needed for the high compression requirements of MPEG-2. For progressive scanned video there is very little difference between MPEG-1 and MPEG-2 compression capabilities. However, interlace presents complications in removing both types of redundancy, and many features have been added to deal specifically with it. MP@ML\(^8\) only allows 4:2:0 color sampling.

For interlace, MPEG-2 specifies a choice of two picture structures. Field Pictures consist of fields that are coded independently. With Frame Pictures, on the other hand, field pairs are combined into frames before coding. MPEG-2 requires interlace to be decoded as alternate top and bottom fields.\(^9\) However, either field can be temporally first within a frame.

MPEG-2 allows for progressive coded pictures, but interlaced display. In fact, coding of 24 frame/s film source with 3:2 pulldown display for 525/60 video is also supported.

\(^8\)The top field contains the top line of the frame. The bottom field contains the second (and bottom) line of the frame.
2.3.2.1 Basics of MPEG-2 Video Compression

We now provide a brief overview of basic principles employed in MPEG-2 video compression.

Exploiting Spatial Redundancy

As in MPEG-1, the MPEG-2 video-coding algorithm employs an $8 \times 8$ Block-based two-dimensional DCT. A quantizer is applied to each DCT coefficient, which sets many of them to zero. Compression is then achieved by transmitting only the non-zero coefficients and by entropy-coding their locations and amplitudes.

The main effect of interlace in frame pictures is that since alternate scan lines come from different fields, vertical correlation is reduced when there is motion in the scene. MPEG-2 provides two features for dealing with this.

First, with reduced vertical correlation, the zigzag scanning order for DCT coefficients used for progressive frames is no longer optimum. Thus, MPEG-2 has an Alternate Scan that may be specified by the encoder on a picture by picture basis. A threshold on the sum of absolute vertical line differences is often sufficient for deciding when to use the alternate scan. Alternatively, DCT coefficient amplitudes after transformation and quantization could be examined.

Second, a capability for field coding within a frame picture MB is provided. That is, just prior to performing the DCT, the encoder may reorder the luminance lines within a MB so that the first 8 lines come from the top field, and the last 8 lines come from the bottom field. This reordering is undone after the IDCT in the encoder and the decoder. Chrominance blocks are not reordered in MP@ML. The effect of this reordering is to increase the vertical correlation within the luminance blocks and thus decrease the DCT coefficient energy. Again, a threshold on the sum of absolute vertical line differences is often sufficient for deciding when to use field DCT coding in a MB.

Exploiting Temporal Redundancy

As in MPEG-1, the quality achieved by intraframe coding alone is not sufficient for typical MPEG-2 video signals at bit-rates around 4 Mbits/s. Thus, MPEG-2 uses all the temporal redundancy reduction techniques of MPEG-1, plus other methods to deal with interlace.

Frame Prediction for Frame Pictures

MPEG-1 exploits temporal redundancy by means of motion compensated MBs and the use of P-pictures and B-pictures. MPEG-2 refers to these methods as Frame Prediction for Frame Pictures, and, as in MPEG-1, assigns up to one motion vector to Forward Predicted Target MBs and up to two motion vectors to Bidirectionally Predicted Target MBs. Prediction MBs are taken from previously coded Reference frames, which must be either P-pictures or I-pictures. Frame Prediction works well
for slow to moderate motion, as well as panning over a detailed background. Frame Prediction cannot be used in Field Pictures.

Field Prediction for Field Pictures

A second mode of prediction provided by MPEG-2 for interlaced video is called Field Prediction for Field Pictures. This mode is conceptually similar to Frame Prediction, except that Target MB pel's all come from the same field. Prediction MB pel's also come from one field, which may or may not be of the same polarity (top or bottom) as the Target MB field. For P-pictures, the Prediction MBs may come from either of the two most recently coded I- or P-fields. For B-pictures, the Prediction MBs are taken from the two most recently coded I- or P-frames. Up to one motion vector is assigned to Forward Predicted Target MBs and up to two motion vectors to Bidirectionally Predicted Target MBs. This mode is not used for Frame Pictures and is useful mainly for its simplicity.

Field Prediction for Frame Pictures

A third mode of prediction for interlaced video is called Field Prediction for Frame Pictures [9]. With this mode, the Target MB is first split into top-field pel's and bottom-field pel's. Field prediction, as defined above, is then carried out independently on each of the two parts of the Target MB. Thus, up to two motion vectors are assigned to Forward Predicted Target MBs and up to four motion vectors to Bidirectionally Predicted Target MBs. This mode is not used for Field Pictures and is useful mainly for rapid motion.

Dual Prime for P-Pictures

A fourth mode of prediction, called Dual Prime for P-pictures, transmits one motion vector per MB. From this motion vector two preliminary predictions are computed, which are then averaged together to form the final prediction. The previous picture cannot be a B-picture for this mode. The first preliminary prediction is identical to Field Prediction, except that the Prediction pel's all come from the previously coded fields having the same polarity (top or bottom) as the Target pel's. Prediction pel's, which are obtained using the transmitted MB motion vector, come from one field for Field Pictures and from two fields for Frame Pictures.

2.3.2.2 MPEG-2 Bitstream Data Hierarchy

Most of MPEG-2 consists of additions to MPEG-1. The Header structure of MPEG-1 was maintained with many additions within each Header. For constant bit-rates, the Video Buffer Verifier (VBV) was also kept. However, unlike MPEG-1, picture skipping as in H.261 is allowed.
Video Sequence Header

The Video Sequence Header contains basic parameters such as the size of the coded video pictures, size of the displayed video pictures, Pel Aspect Ratio (PAR), picture rate, bit-rate, VBV buffer size, Intra and NonIntra Quantizer Matrices (defaults are the same as MPEG-1), Profile/Level Identification, Interlace/Progressive Display Indication, Private user data, and certain other global parameters.

Group of Pictures (GOP) Header

Below the Video Sequence Header is the Group of Pictures (GOP) Header, which provides support for random access, fast search, and editing. The GOP Header contains a time code (hours, minutes, seconds, frames) used by certain recording devices. It also contains editing flags to indicate whether the B-pictures following the first I-picture can be decoded following a random access.

Picture Header

Below the GOP is the Picture Header, which contains the type of picture that is present, e.g., I, P or B, as well as a Temporal Reference indicating the position of the picture in display order within the GOP. It also contains the parameter vbb_delay that indicates how long to wait after a random access before starting to decode. Without this information, a decoder buffer could underflow or overflow following a random access.

Within the Picture Header a picture display extension allows for the position of a display rectangle to be moved on a picture by picture basis. This feature would be useful, for example, when coded pictures having Image Aspect Ratio (IAR) 16:9 are to be also received by conventional TV having IAR 4:3. This capability is also known as Pan and Scan.

MPEG-2 Slice Header

The Slice Header is intended to be used for re-synchronization in the event of transmission bit errors. It is the responsibility of the encoder to choose the length of each Slice depending on the expected bit error conditions. Prediction registers used in differential encoding are reset at the start of a Slice. Each horizontal row of MBs must start with a new Slice, and the first and last MBs of a Slice cannot be skipped. In MP@ML, gaps are not allowed between Slices. The Slice Header contains the vertical position of the Slice within the picture, as well as a quantizer_scale parameter (corresponding to GQUANT in H.261). The Slice Header may also contain an indicator for Slices that contain only Intra MBs. These may be used in certain Fast Forward and Fast Reverse display applications.
Macroblock Header

In the Macroblock Header, the horizontal position (in MBs) of the first MB of each Slice is coded with the MB Address VLC. The positions of additional transmitted MBs are coded differentially with respect to the most recently transmitted MB also using the MB Address VLC. Skipped MBs are not allowed in I-pictures. In P-pictures, skipped MBs are assumed NonIntra with zero coefficients and zero motion vectors. In B-pictures, skipped MBs are assumed NonIntra with zero coefficients and motion vectors the same as the previous MB. Also included in the Macroblock Header are MB Type (Intra, NonIntra, etc.), Motion Vector Type, DCT Type, quantizer_scale (corresponding to MQUANT in H.261), motion vectors and coded block pattern. As with other Headers, many parameters may or may not be present, depending on MB Type.

MPEG-2 has many more MB Types than MPEG-1, due to the additional features provided as well as to the complexities of coding interlaced video. Some of these will be discussed below.

Block

A Block consists of the data for the quantized DCT coefficients of an $8 \times 8$ Block in the Macroblock. It is VLC coded as described. MP@ML has six blocks per MB. For noncoded Blocks, the DCT coefficients are assumed to be zero.

2.3.2.3 MPEG-2 Encoding/Decoding Architecture

Figure 2.7 shows a high-level block diagram of an MPEG-2 nonscalable video codec. The video encoder consists of an inter-frame/field DCT encoder, a frame/field motion estimator and compensator, and a variable length entropy encoder (VLE). The frame/field DCT encoder exploits spatial redundancies in the video,
and the frame/field motion compensator exploits temporal redundancies in the video signal. The coded video bitstream is sent to a systems multiplexer, Sys Mux, which outputs either a Transport or a Program Stream.

The MPEG-2 decoder of Fig. 2.7 consists of a variable length entropy decoder (VLD), inter-frame/field DCT decoder, and the frame/field motion compensator. Sys Demux performs the complementary function of Sys Mux and presents the video bitstream to the VLD for decoding of motion vectors and DCT coefficients. The frame/field motion compensator uses a motion vector decoded by the VLD to generate a motion compensated prediction that is added back to a decoded prediction error signal to generate decoded video out. This type of coding produces nonscalable video bitstreams, since normally the full spatial and temporal resolution coded is the one that is expected to be decoded.

2.3.2.4 MPEG-2 Scalable Encoding/Decoding Architecture

A high level block diagram of a generalized codec for MPEG-2 scalable video coding is shown in Fig. 2.8. Scalability is the property that allows decoders of various complexities to be able to decode video of resolution/quality commensurate with their complexity from the same bit-stream. The generalized structure of Fig. 2.8 provides capability for both spatial and temporal resolution scalability in the following manner. The input video goes through a pre-processor that produces two video signals, one of which (called the Base Layer) is input to a standard MPEG-1 or MPEG-2 nonscalable video encoder, and the other (called the Enhancement Layer) is input to an MPEG-2 enhancement video encoder. The two bitstreams, one from each encoder, are multiplexed in Sys Mux (along with coded audio and user data). In this manner it becomes possible for two types of decoders to be able to decode a video signal of quality commensurate with their complexity, from the same encoded bit stream.
A *Profile* specifies the coding features supported, while a *Level*, specifies the picture resolutions, bit-rates, etc. that can be handled. A number of Profile-Level combinations have been defined, the most important ones are *Main Profile at Main Level* (MP@ML), and *Main Profile at High Levels* (MP@HL-1,440, and MP@HL).

Table 2.1 shows parameter constraints for MP@ML, MP@HL-1,440, and MP@HL.

In addition to the Main Profile, several other MPEG-2 Profiles have been defined. The *Simple Profile* is basically the Main Profile without B-pictures. The *SNR Scalable Profile* adds SNR Scalability to the main Profile. The *Spatially Scalable Profile* adds Spatial Scalability to the SNR Scalable Profile, and the *High Profile* adds 4:2:2 color to the Spatially Scalable Profile. All scalable Profiles are limited to at most three layers.

The Main Level is defined basically for BT601 video. In addition, the *Simple Level* is defined for SIF video. Two higher levels for HDTV are the *High-1,440 Level*, with a maximum of 1,440 pels/line, and the *High Level*, with a maximum of 1,920 pels/line.

### 2.3.3 MPEG-2 Capabilities and Interoperability

<table>
<thead>
<tr>
<th>Profile/Level</th>
<th>Video Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Main</strong></td>
<td>720 pels/frame</td>
</tr>
<tr>
<td></td>
<td>576 lines/frame</td>
</tr>
<tr>
<td></td>
<td>30 frames/s</td>
</tr>
<tr>
<td></td>
<td>10.4 Msamples/s</td>
</tr>
<tr>
<td></td>
<td>15 Mbits/s</td>
</tr>
<tr>
<td><strong>High</strong></td>
<td>1,920 pels/frame</td>
</tr>
<tr>
<td></td>
<td>1,152 lines/frame</td>
</tr>
<tr>
<td></td>
<td>60 frames/s</td>
</tr>
<tr>
<td></td>
<td>62.7 Msamples/s</td>
</tr>
<tr>
<td></td>
<td>80 Mbits/s</td>
</tr>
<tr>
<td><strong>High-1,440</strong></td>
<td>1,440 pels/frame</td>
</tr>
<tr>
<td></td>
<td>1,152 lines/frame</td>
</tr>
<tr>
<td></td>
<td>60 frames/s</td>
</tr>
<tr>
<td></td>
<td>47.0 Msamples/s</td>
</tr>
<tr>
<td></td>
<td>60 Mbits/s</td>
</tr>
</tbody>
</table>

### 2.4 The MPEG-4 (Part 2) Video Standard

MPEG-4 builds on and combines elements from three fields: digital television, interactive graphics and the World Wide Web with focus of not only providing higher compression but also an increased level of interactivity with the audio-visual content, i.e., access to and manipulation of objects in a visual scene.
The MPEG-4 standard, or formally ISO 14496 was started in 1993, and core of the standard was completed by 1998. Since then there have been a number of extensions and additions to the standard.

Overall, MPEG-4 addresses the following main requirements:

- Content Based Access
- Universal Accessibility
- Higher Compression

While previous standards code pre-composed media (e.g., video and corresponding audio), MPEG-4 codes individual visual objects and audio objects in the scene and delivers, in addition, a coded description of the scene. At the decoding side, the scene description and individual media objects are decoded, synchronized and composed for presentation.

2.4.1 H.263 Coding Concepts as Applicable to MPEG-4 Video

The H.263 video coding standard is based on the same DCT and motion compensation techniques as used in H.261. Several small, incremental improvements in video coding were added to the H.263 baseline standard for use in POTS conferencing. These included the following:

- Half pixel motion compensation in order to reduce the roughness in measuring best matching blocks with coarse time quantization. This feature significantly improves the prediction capability of the motion compensation algorithm in cases where there is object motion that needs fine spatial resolution for accurate modeling.
- Improved variable-length coding.
- Reduced overhead of channel input.
- Optional modes including unrestricted motion vectors that are allowed to point outside the picture.
- Arithmetic coding in place of the variable length (Huffman) coding.
- Advanced motion prediction mode including overlapped block motion compensation.
- A mode that combines a bidirectionally predicted picture with a normal forward predicted picture.

In addition, H.263 supports a wider range of picture formats including 4CIF (704 × 576 pixels) and 16CIF (1,408 × 1,152 pixels) to provide a high resolution mode picture capability.

2.4.2 MPEG-4 Video Objects Coding

While the MPEG-4 Visual coding includes coding of natural video as well as synthetic visual (graphics, animation) objects, here we discuss coding of natural video objects only.
The concept of Video Objects (VO) and their temporal instances, Video Object Planes (VOPs) is central to MPEG-4 video. A temporal instance of a video object can be thought of as a snapshot of arbitrary shaped object that occurs within a picture, such that like a picture, it is intended to be an access unit, and, unlike a picture, it is expected to have a semantic meaning.

Figure 2.9 shows an example of segmentation of a Picture into spatial VOPs with each VOP representing a snapshot in time of a particular object.

2.4.2.1 Basics of MPEG-4 (Part 2) Compression

To consider how coding takes place as per MPEG-4 video, consider a sequence of vops. MPEG-4 extends the concept of intra (I-) pictures, predictive (P-) and bidirectionally predictive (B-) pictures of MPEG-1/2 video to vops, thus I-vop, P-vop and B-vop result. Figure 2.10 shows a coding structure that uses a B-vop between a pair of reference vops (I- or P-vops). In MPEG-4, a vop can be represented by its shape (boundary), texture (luma and chroma variations), and motion (parameters).

The extraction of shape of vops per picture of an existing video scene by semi-automatic or automatic segmentation, results in a binary shape mask. Alternatively, if special mechanisms such as blue-screen composition are employed for generation of video scenes, either binary, or grey scale (8-bit) shape can be extracted. Both the binary- or the grey scale- shape mask needs to be sufficiently compressed for efficient representation. The shape information is encoded separate from encoding of the texture field, and motion parameters.
2.4.2.2 MPEG-4 (Part 2) Video Compression Core Tools, and Architecture

In Fig. 2.11, we show the high level structure of an MPEG-4 Video Encoder which codes a number of video objects of a scene. Its main components are: Motion Coder, Texture and Shape Coder. The Motion Coder performs motion estimation and compensation of macroblock and blocks, similar to H.263 and MPEG-1/2 but modified to work with arbitrary shapes. The Texture Coder uses block DCT coding that is much more optimized than that of H.263 and MPEG-1/2, and further it is adapted to work with arbitrary shapes. The Shape Coder, represents an entirely new component not present in previous standards. The coded data of multiple vops (representing multiple simultaneous video objects) is buffered and sent to the System Multiplexer.

The Shape Coder allows efficient compression of both the binary-, and the grey-scale shape. Coding of binary shape is performed on a 16×16 block basis as follows. To encode a shape, a tight bounding rectangle that extends to multiples of 16×16 blocks is first created around the shape, with transparency of extended samples set to zero. On a macroblock basis, binary shape is then encoded using context information, motion compensation, and arithmetic coding. Similarly, grey scale shape is also encoded on a macroblock basis. Grey scale shape has similar properties as luma, and is thus encoded in a similar manner using texture and motion compensation tools used for coding of luma.

The Motion Coder consists of a Motion Estimator, Motion Compensator, Previous/Next Vops Store and Motion Vector (MV) Predictor and Coder. In case of P-vops, Motion Estimator computes motion vectors using the current vop and temporally previous reconstructed vop available from the Previous Reconstructed Vops Store. In case of B-vops, Motion Estimator computes motion vectors using the current vop and temporally previous reconstructed vop from the Previous Reconstructed Vop Store, as well as, the current vop and temporally next vop from
the Next Reconstructed Vop Store. The Motion Compensator uses these motion vectors to compute motion compensated prediction signal using the temporally previous reconstructed version of the same vop (reference vop). The MV Predictor and Coder generates prediction for the MV to be coded. Special padding is needed for motion compensation of arbitrary shaped Vops.

MPEG-4 B-pictures include an efficient mode for generating predictions, called the “temporal direct mode” or simply the “direct mode”. This mode allows motion vectors not only for 16×16 blocks but also for 8×8 blocks. The mode uses direct bi-directional motion compensation by computing scaled forward and backward motion vectors. The direct mode utilizes the motion vectors of the co-located macroblock in the most recently decoded I- or P-vop; the co-located macroblock is defined as the macroblock that has the same horizontal and vertical index with the current macroblock in the B-vop. If the co-located macroblock is transparent and thus the motion vectors are not available, these motion vectors are assumed to be zero. Besides the direct mode, MPEG-4 B-vops also support other modes of MPEG-1/2 B-pictures such as forward, backward, and bidirectional (interpolation) modes.

The Texture Coder codes luma and chroma blocks of macroblocks of a vop. Two types of macroblocks exist, those that lie inside the vop and those that lie on the boundary of the vop. The blocks that lie inside the vop are coded using DCT coding similar to that used in H.263 but optimized in MPEG-4. The blocks that lie on the vop boundary are first padded and then coded similar to the block that lie inside the vop. The remaining blocks that are inside the bounding box but outside of the coded vop shape are considered transparent and thus are not coded.

The texture coder uses block DCT coding and codes blocks of size 8 × 8 similar to H.263 and MPEG-1/2, with the difference that since vop shapes can be arbitrary, the blocks on the vop boundary require padding prior to texture coding. The general operations in the texture encoder are: DCT on original or prediction error blocks of size 8 × 8, quantization of 8 × 8 block DCT coefficients, scanning of quantized coefficients and variable length coding of quantized coefficients. For inter (prediction error block) coding, the texture coding details are similar to that of H.263 and MPEG-1/2. However, for intra coding a number of techniques to improve coding efficiency such as variable de-quantization, prediction of the DC and AC coefficients, and adaptive scanning of coefficients, are included.

2.4.2.3 MPEG-4 (Part 2) Video Compression Extension Tools

A number of advanced coding efficiency tools were added to the core standard and can be listed as follows.

*Quarter pel* prediction allows higher precision in interpolation during motion compensation resulting in higher overall coding efficiency. While it involves higher coding cost of motion information it also allows for more accurate representation of motion, significantly reducing motion compensated prediction errors.

*Global motion compensation* allows increase in coding efficiency due to compact parametric representation of motion that can be used to efficiently compensate for global
movement of object/s or that of camera. It supports the five transformation models for the warping - stationary, translational, isotropic, affine, and perspective. The pixels in a global motion compensated macroblock are predicted using global motion prediction. The predicted macroblock is obtained by applying warping to the reference object. Each macroblock can be predicted either from previous vop by global motion compensation using warping parameters, or by local motion compensation using macroblock/block motion vectors. The resulting prediction error block is encoded by DCT coding.

**Sprite** in MPEG-4 video, is a video object that represents a region of coherent motion. Thus a sprite is a single large composite image resulting from blending of uncovered pixels belonging to various temporal instances of the video object. Alternatively, given a sprite, video objects at various temporal instances can be synthesized from it. A sprite is thus said to have the potential of capturing spatiotemporal information in a compact manner. MPEG-4 supports coding of sprites as a special type of vop (S-vop).

**Shape-Adaptive DCT** (SA-DCT) was added to the MPEG-4 standard to provide a more efficient way of coding blocks that contain shape boundaries and thus need to be padded resulting in much higher number of coefficients as compared to the active pixels of such blocks that need to be coded. SA-DCT produces at the most the same number of DCT coefficients as the number of pixels of a block that need to be coded. It is applied as a normal separable DCT, albeit on rows and columns such that each row or columns is not of equal size. Coefficients of SA-DCT are then quantized and coded as other DCT coefficients.

### 2.4.2.4 MPEG-4 (Part 2) Video Error Resilience Tools

MPEG-4 includes a number of error resilience tools to increase robustness of coded data. First, MPEG-4 uses a packet approach to allow resynchronization of the bitstream to restart decoding after detection of errors. It involves introducing periodic resynchronization markers in the bitstream. The length of the packet is independent of the contents of a packet but depends only on a fixed number of bits, ie, resynchronization markers are distributed uniformly throughout the bitstream. Second it allows for insertion of header extension information in the bitstream that makes it possible to decode each video packet independent of the previous information, allowing for faster resynchronization. Third, it allows for increased error resilience via use of “data partitioning” that involves separation of motion and macroblock header information from texture information using a motion marker to identify the separation point. This allows for better error concealment as in case of residual errors in the texture information, the previously received motion information can be used to provide motion compensated error concealment. Fourth, it allows use of reversible variable length codes (RVLCs) for DCT coefficient coding that can allow faster recovery after errors have been detected as the bitstream can be decoded both in forward and reverse directions. Thus the part of the bitstream that can not be decoded in the forward direction can be decoded in the backward direction; this
allows for recovery of a larger portion of the bitstream and thus increases the correctly recovered data in case of errors. However the use of RVLC incurs some additional constant penalty due to overhead so it can decrease coding performance when no errors occur.

### 2.4.2.5 MPEG-4 (Part 2) Video Scalability Tools

Some applications may require video objects to be simultaneously available in several spatial or temporal resolutions. Thus, MPEG-4 supports Temporal, and Spatial scalability of video objects.

#### Temporal Scalability

Temporal scalability involves partitioning of the video objects in layers, where the base layer provides the basic temporal rate, and the enhancement layers, when combined with the base layer, provide higher temporal rates. MPEG-4 supports temporal scalability of both rectangular and arbitrary shaped video objects. Temporal scalability is both computationally simple and efficient (essentially similar to B-vops coding that is already supported in nonscalable video coding). Figure 2.12 shows an example of two layer temporally scalable coding of arbitrary shaped vops. In this example, I- and P-vops in the base layer are independently coded using prediction from previous decoded vops, while the enhancement layer consists of B-vops that are coded using prediction from decoded base layer vops.

#### Spatial Scalability

Spatial scalability involves generating two spatial resolution video layers from a single source such that the lower layer can provide the lower spatial resolution, and the higher spatial resolution can be obtained by combining the enhancement layer with the interpolated lower resolution base layer. MPEG-4 supports spatial scalability of both rectangular and arbitrary shaped objects. Since arbitrarily shaped objects can have varying sizes and locations, the video objects are resampled. An absolute
reference coordinate frame is used to form the reference video objects. The resampling involves temporal interpolation, and spatial relocalization. This ensures the correct formation of the spatial prediction used to compute the spatial scalability layers. Before the object are resampled, the objects are padded to form objects that can be used in the spatial prediction process.

Figure 2.13 shows an example of two layer spatially scalable coding of arbitrary shaped vops. In this example, the I-, P- and B-vops in the base layer are independently coded using prediction from previous decoded vops in the same layer while the enhancement layer P- and B-vops are coded using prediction from decoded base layer vops as well as prediction from previous decoded vops of enhancement layer.

**2.4.3 MPEG-4 Video Capabilities and Interoperability**

As mentioned earlier, MPEG-4 is a large standard containing many tools as it was aimed at addressing a new class of applications requiring access to coded audio-visual objects. In this chapter we have covered key aspects of MPEG-4 part 2 video including video compression core tools, video compression extension tools, error resilience tools, and scalability tools. Due to large number of tools, there are a large number of (Object Type) Profiles in MPEG-4 each with many levels. However only the Simple Profile, and the Advanced Simple Profile have been widely implemented.

Besides MPEG-4 part 2 video, later MPEG-4 added a newer more efficient video coding technology referred to as MPEG-4 part 10 video (also known as AVC/H.264 standard) and is discussed in the next chapter of this book.

**2.5 Summary**

In this chapter we have provided an overview of the MPEG standards including requirements, technology, and applications/profiles of the MPEG-1 video, MPEG-2 video and MPEG-4 Part 2 video standards. Table 2.2 presents a side-side comparison of the technology features of the three early MPEG standards.
<table>
<thead>
<tr>
<th>Features/technology</th>
<th>MPEG-1 video</th>
<th>MPEG-2 video</th>
<th>MPEG-4 (part 2) video</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Video structure</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sequence type</td>
<td>Progressive</td>
<td>Interlace, progressive</td>
<td>Interlace, progressive</td>
</tr>
<tr>
<td>Picture structure</td>
<td>Frame</td>
<td>Top/bot. field, frame</td>
<td>Top/bot. field, frame</td>
</tr>
<tr>
<td>2. Core compression</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Picture types</td>
<td>I-, P-, B-pictures</td>
<td>I-, P-, B-pictures</td>
<td>I-, P-, B-rect./arb. vops</td>
</tr>
<tr>
<td>Motion comp. block size</td>
<td>16 × 16</td>
<td>16 × 16 (16 × 8 field)</td>
<td>16 × 16, 8 × 8 (overlapped)</td>
</tr>
<tr>
<td>B-motion comp modes</td>
<td>Forw/backw/intp</td>
<td>Forw/backw/intp</td>
<td>Forw/backw/intp/direct</td>
</tr>
<tr>
<td>Transform, and block size</td>
<td>DCT, 8 × 8</td>
<td>DCT, 8 × 8</td>
<td>DCT, 8 × 8</td>
</tr>
<tr>
<td>Quantizer type</td>
<td>Linear</td>
<td>Linear and nonlinear</td>
<td>Linear and nonlinear</td>
</tr>
<tr>
<td>Quantizer matrices</td>
<td>Full</td>
<td>Full</td>
<td>Full and partial</td>
</tr>
<tr>
<td>Coefficient prediction</td>
<td>DC only</td>
<td>DC only</td>
<td>Adapt. DC and AC coeff.</td>
</tr>
<tr>
<td>Scan of coefficients</td>
<td>Zigzag</td>
<td>Zigzag, alt. vert</td>
<td>Zigzag, alt. vert, alt. horiz</td>
</tr>
<tr>
<td>Entropy coding</td>
<td>Fixed vlc</td>
<td>Intra, and inter vlc</td>
<td>Adaptive vlc</td>
</tr>
<tr>
<td>3. Higher compression</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Impr. motion comp. prec.</td>
<td>No</td>
<td>No</td>
<td>Quarter pel (v2)</td>
</tr>
<tr>
<td>Impr. motion comp. mode</td>
<td>No</td>
<td>No</td>
<td>Global motion comp. (v2)</td>
</tr>
<tr>
<td>Background representation</td>
<td>No</td>
<td>No</td>
<td>Sprite mode (v2)</td>
</tr>
<tr>
<td>Transform of nonrect. block</td>
<td>No</td>
<td>No</td>
<td>Shape adaptive DCT (v2)</td>
</tr>
<tr>
<td>4. Interlace compression</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frame and field pictures</td>
<td>Frame picture</td>
<td>Frame, field picture</td>
<td>Frame, field picture</td>
</tr>
<tr>
<td>Motion comp. frame/field</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Coding frame/field</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>5. Object based video</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arbitrary object shape</td>
<td>No</td>
<td>No</td>
<td>Vop</td>
</tr>
<tr>
<td>Shape coding</td>
<td>No</td>
<td>No</td>
<td>Binary, grey scale alpha</td>
</tr>
<tr>
<td>6. Error resilient video</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resynch markers</td>
<td>Slice start code</td>
<td>Slice start code</td>
<td>Flexible markers</td>
</tr>
<tr>
<td>Data partitioning</td>
<td>No</td>
<td>Coeff only</td>
<td>Coeffs, mv</td>
</tr>
<tr>
<td>Resilient entropy coding</td>
<td>–</td>
<td>–</td>
<td>Reversible vlc</td>
</tr>
<tr>
<td>7. Scalable video</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNR scalability</td>
<td>No</td>
<td>Picture</td>
<td>(Via spatial)</td>
</tr>
<tr>
<td>Spatial scalability</td>
<td>No</td>
<td>Picture</td>
<td>Picture, vop</td>
</tr>
<tr>
<td>Temporal scalability</td>
<td>No</td>
<td>Picture</td>
<td>Picture, vop</td>
</tr>
<tr>
<td>8. Stereo/3D/multiview</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stereo/multiview support</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
References

3. J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall, MPEG Video Compression
4. B. G. Haskell, A. Puri, and A. N. Netravali, Digital Video: An Introduction to MPEG-2,
5. A. Puri, and T. Chen, Multimedia Systems, Standards and Networks, Marcel Dekker, New
6. ITU-T Recommendation H.261, Video Codec for Audiovisual Services at p*64 kbits/sec.
8. “Coding of Moving Pictures and Associated Audio for Digital Storage Media up to about
9. “Generic Coding of Moving Pictures and Associated Audio Information,” ISO/IEC 13818-2:
   Video (November 1994).
11. A. Puri, R. Aravind, and B. G. Haskell, “Adaptive Frame/Field Motion Compensated Video
The MPEG Representation of Digital Media
Chiariglione, L. (Ed.)
2012, VI, 268 p., Hardcover