CHAPTER

Video Coding Standards 12

CHAPTER OUTLINE

Communicating Pictures. http://dx.doi.org/10.1016/B978-0-12-405906-1.00012-X

The process of standardizing video formats and compression methods has been a major influence on the universal adoption of video technology. Standards are essential for interoperability, enabling material from different sources to be processed and transmitted over a wide range of networks, or stored on a wide range of devices. This interoperability provides the widest possible range of services for users. It also reduces risk for manufacturers, stimulates investment in research and development, and has created an enormous market for video equipment, with the advantages of volume manufacturing.

Video coding standards have, from the introduction of H.120 in 1984 through to the most recent H.265/HEVC codec, delivered a halving of bit rate for the equivalent video quality every 10 years. This chapter overviews the features that have enabled this progress. It is not intended to be a definitive guide to the structure and implementation of modern coding standards, but rather a description that enables the reader to understand how the architectures, approaches, and algorithms described in previous chapters are employed in the codecs that are in common use today.

12.1 The need for and role of standards

A chronology of video coding standards is represented in Figure 12.1 (repeated for convenience from Chapter 1). Their primary features are described in the following sections. Let us first, however, revisit why we need standards and consider the process that has led to a succession of successful coding standards over the last 30 years or so.

12.1.1 The focus of video standardization

In order for a video coding standard (or any other standard for that matter) to be successful it must clearly satisfy a demand. In particular: it must be superior in

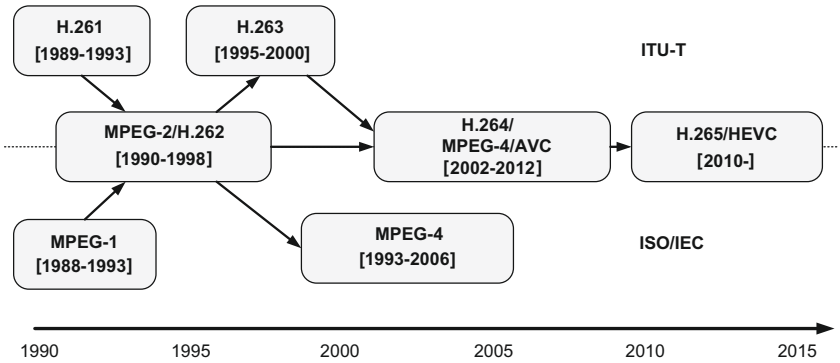

FIGURE 12.1

A chronology of video coding standards from 1990 to the present date.

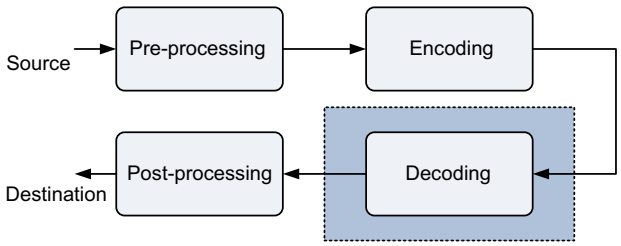

FIGURE 12.2

The scope of standardization.

performance to any previous standards, it must not stifle competition between producers of products (i.e. it must allow innovation) and, of course, it must provide interoperability through independence from specific communication networks or storage devices. In most cases it is advantageous if a standard is backward compatible—i.e. it should be able to decode bitstreams from prior standards. The standardization process must also be mindful that future standards will need to be backward compatible with it.

Video coding standards conventionally define the bitstream format and syntax and the decoding process, not (for the most part) the encoding process. This is illustrated in Figure 12.2 where the dashed box indicates the normative aspects of the standard. A standard-compliant encoder is thus one that produces a compliant bitstream and a standard-compliant decoder is one that can decode a compliant bitstream. In this context it is important to highlight the fact that the standard compliance of an encoder provides no guarantee of quality; the challenge remains for manufacturers to differentiate their products through innovative low complexity and efficient coding solutions.

12.1.2 The standardization process

Early video coding standards were produced independently either by the International Telecommunication Union (ITU) (previously CCITT) or by the International

Standards Organization (ISO), specifically the joint ISO/IEC (International Electrotechnical Commission) Technical Committee 1 (JTC 1) that addresses all computer-related activities including video compression. Most recent standards have however benefited from a joint approach between these two organizations.

The standardization process [1] follows a well defined path, with the following stages:

Requirements Definition: Here the scope of the standard is finalized and the requirements and goals of the standardization process defined.

Divergence: During this phase, competition is introduced, initially through the identification of Key Technical Areas (KTAs). A formal call for proposals is then normally issued, enabling experts from industry and academia to present and compare their methods and results, usually against a predefined set of test data.

Convergence: The aim of this phase is to create a solution for the standard by selecting the best algorithms contributed during the divergence phase. This is normally achieved through the use of an evolving test model such as the JM (Joint Model) in H.264/AVC or the HEVC Test Model (HM). This model evolves through a number of iterations through its use as a reference for subsequent proposals. These, if successful, are incorporated in a new improved version of the model. This process continues until the required performance specifications are met.

Verification: During verification, the resulting standard and its bitstream are validated for conformance. Conformance testing processes are then defined for compliance testing of products.

12.1.3 Intellectual property and licensing

When there is mass investment in producing products according to an international standard, the manufacturers and users rightly expect some degree of protection against patent infringement litigation. For most standardized codecs, vendors and users are required to pay royalties to the owners of associated intellectual property. This is, in the main, handled by a US organization (not affiliated with MPEG) called the MPEG Licensing Authority. MPEG LA administers patent licenses in connection with the patent pools for MPEG-2, MPEG-4, VC-1, and H.264/AVC. It is also pooling patents for licensing of HEVC.

Between 2005 and 2007, a dispute between Qualcomm and Broadcom, on the infringement of H.264 patents, came before the US Court. The associated patents were judged to be unenforceable as they had not been disclosed to MPEG JVT prior to the H.264 standardization in 2003. This type of ruling goes some way to provide confidence in the robustness of the licensing procedures adopted for video standardization.

12.2 H.120

12.2.1 Brief history

Study Group SG.XV of the CCITT commenced work on H.120 in 1980 and produced the first international digital video coding standard in 1984, followed by a second

version in 1988 [2]. H.120 addressed videoconferencing applications at 2.048 Mb/s and 1.544 Mb/s for 625/50 and 525/60 TV systems respectively. This standard was never a commercial success, partially because it was based on different coding strategies for different international regions, but mainly because its picture quality (especially temporal quality) was inadequate.

12.2.2 Primary features

H.120 implemented a *conditional replenishment* strategy, whereby each frame was divided into changed and unchanged regions. The changed regions were coded with intra-field DPCM in parts 1 and 2 of the standard, although part 3 (for use in the USA) employed background prediction and motion-compensated inter-field prediction.

12.3 H.261

12.3.1 Brief history

H.261 [3] followed H.120 in 1989 and was the first video codec that achieved widespread product adoption. It was based on a $p \times 64$ kbps ($p = 1, \ldots, 30$) model, targeted at ISDN conferencing applications. H.261 was the first block-based hybrid compression algorithm to use a combination of transformation (the Discrete Cosine Transform (DCT)), temporal DPCM, and motion compensation. This architecture has stood the test of time as all major video coding standards since have been based on it.

12.3.2 Picture types and primary features

Macroblock, GOB, and frame format

H.261 introduced a basic hierarchy into the picture coding process that, with some modification, is still in use today. H.261 pictures, in CIF or QCIF format, are represented as a sequence of Groups of Blocks (GOBs), each comprising a number of macroblocks (MBs). A macroblock, as described in Chapter 4, comprises four spatial 8×8 DCT transformed luma blocks and 2 (sub-sampled) spatial 16×16 DCT-transformed chroma blocks. A diagram showing the picture hierarchy is given in Figure 12.3 for the case of a CIF frame. As can be seen, there are 33 MBs in a GOB, giving a GOB dimension of 176×48 pixels, and there are 12 GOBs in a CIF frame, giving a frame dimension of 352×288 pixels.

H.261 supports only I and P frames—B frames are not supported. Motion estimation and compensation are an integral part of the coding process with a fixed block size of 16×16 and a search window of $[-15 \cdots +15]$ pixels.

Coder control

H.261 offered some flexibility in allowing switching between inter- and intra-frame modes and controlling quantizer step sizes at the macroblock level. Advanced features, such as a spatial loop filter, were incorporated for removing high frequency noise

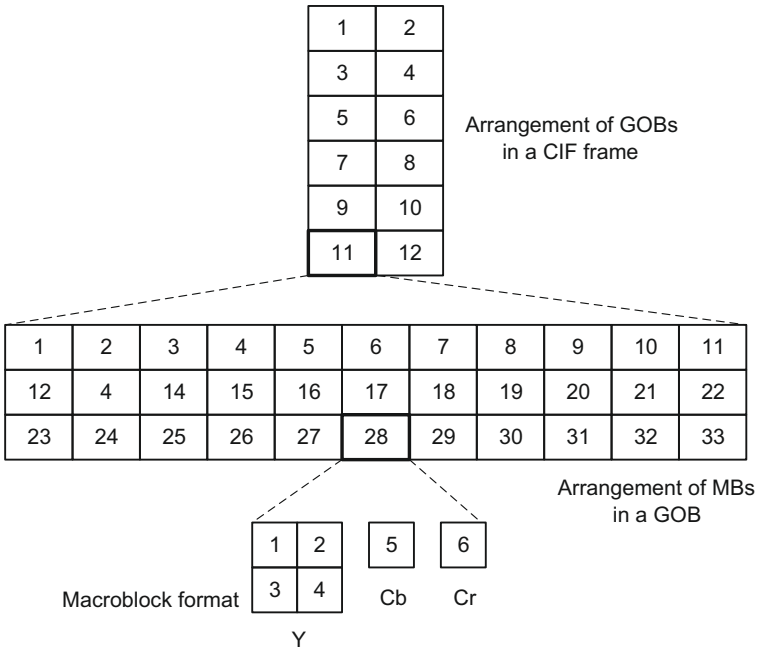

FIGURE 12.3

H.261 macroblock, GOB, and CIF frame format.

(a three-tap filter with coefficients of 0.25, 0.5, and 0.25). This allowed a reduction of prediction error by smoothing the pixels in the reference and output frames. For error resilience, an optional BCH error detection correction (511,493) scheme was also incorporated. The primary features of H.261 are summarized in Table 12.1.

Table 12.1 Primary features of H.261.

Codec feature	Approach
Formats supported	CIF, QCIF
Picture format	Four layers (picture, GOB, MB, and block)
Color sub-sampling	4:2:0 YCbCr
Frame types	I, P
Intra-coding transform	Block DCT
Inter-coding transform	Block DCT
Entropy coding	VLC and Huffman coding
Quantizer	Uniform (DC) and deadzone (AC)
Motion compensation	Optional $[-15 \cdots +15]$
Coding control	Selection of inter/intra and quantizer step size
Loop filter	Three-tap spatial filter
Error protection	Optional BCH (511, 493) coding

12.4 MPEG-2/DVB

12.4.1 Brief history

In 1988, the Moving Picture Experts Group (MPEG) was founded, delivering, in 1992, a video coding algorithm (MPEG-1) intended for digital storage media at 1.5 Mbs/s. This was followed in 1994 by MPEG-2 [4], which specifically targeted the emerging digital video broadcasting market. MPEG-2 was instrumental, through its inclusion in all set-top boxes for more than a decade, in truly underpinning the digital broadcasting revolution.

Because of its focus on broadcasting, where there is an accepted imbalance between encoder and decoder complexity, this led to significant investment in high cost, high complexity encoding technology to produce high performance studio and head-end based encoders. An example of how the performance of MPEG-2 professional encoders improved over a decade of development is shown in Figure 12.4, together with the innovations that led to a series of step changes in that performance.

12.4.2 Picture types and primary features

MPEG-2 (ISO/IEC-13818:2000) [4], also known as H.262, is a generic audiovisual coding standard supporting a range of applications at bit rates from about 2 to 30 Mbps. The standard comprises four main parts: 13818-1: systems, -2: video, -3: audio, and -4: conformance. It employs a hybrid motion-compensated block-based DCT architecture (similar to H.261), using 8 × 8 blocks and 16 × 16 macroblocks

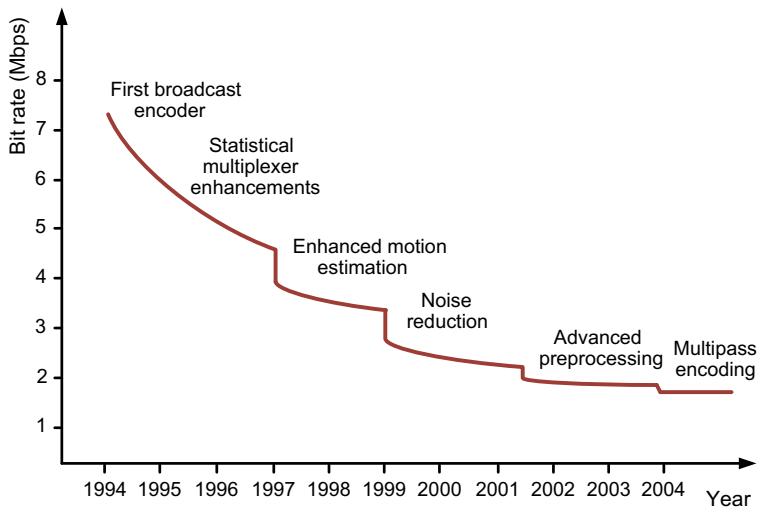

FIGURE 12.4

MPEG-2 encoder performance improvements between 1994 and 2004. (Adapted from an original presented by Tandberg.)

with translational block-based motion estimation to half-pixel accuracy. Because of its focus on digital TV broadcasting, MPEG-2 necessarily supports both progressive and interlaced picture formats. The coding standard was defined primarily for error-free environments and channel coding was added according to the application scenario (e.g. terrestrial, cable, or satellite broadcasting). MPEG-2 was defined to support a broad range of applications from studio processing (capture and editing) to distribution and broadcast delivery. It is intimately coupled with the DVB-T, DVB-S, and DVB-C broadcast content delivery standards.

MPEG-2 supports three picture types: Intra (I)—coded without reference to other frames (least efficient), Predicted (P)—coded based on prediction from previous I or P frame, and Bidirectionally predicted (B)—predicted from P and or I frames but not used as a basis for further predictions (most efficient). It utilizes a GOP structure as illustrated in Figure 12.5 and was the first standard to introduce bidirectionally predicted B frames.

12.4.3 MPEG-2 profiles and levels

To match performance against decoder capability or capacity, MPEG-2 introduced an extensive range of *profiles* and *levels*. A profile is a defined subset of the entire bitstream syntax and profiles are further partitioned into levels. Each level specifies a range of allowable values for the parameters in the bitstream.

MPEG-2 supports six profiles—Simple, Main, SNR, Spatial, High 4:2:2, and Multiview. Provisions for scalability are included in the SNR, Spatial, and High profiles whereas the Simple Profile and Main Profile allow only single-layer coding. It also offers four possible levels—Low, Main, High1440, and High—in each profile. The parameters for the Main Level approximately correspond to normal TV resolution, the Low Level corresponds to CIF resolution, and the values for

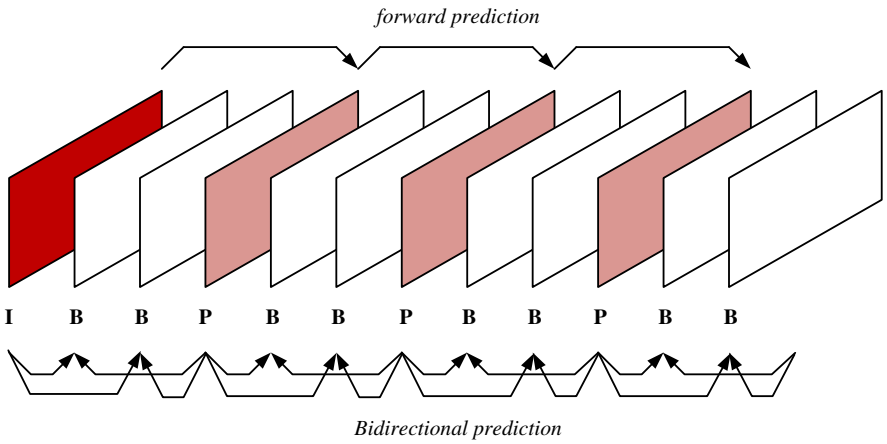

FIGURE 12.5

MPEG-2 GOP structure.

Table 12.2 MPEG-2 main profile@main level (MP@ML) bounds.

Parameter	Bound
Samples/line	720
Lines/frame	576
Frames/sec	30
Samples/sec	10,368,000
Bit rate	15 Mb/s
Buffer size	1,835,008 bits
Chroma format	4:2:0
Aspect ratio	4:3, 16:9 square pixels

High1440 and High correspond to HDTV resolution. Only two profiles were ever used in practice; the 4:2:2 Profile for studio work and post-production and the Main Profile for broadcast TV delivery. Table 12.2 provides the specification for the MPEG-2 Main Profile at Main Level (MP@ML) as used in most delivery applications.

12.5 H.263

12.5.1 Brief history

H.263 [5] was defined by ITU-T SG15, starting in 1993 with the goal of coding at bit rates below 64 kbps. The initial application focus was on PSTN and early mobile radio applications at bit rates between 10 and 24 kb/s. Despite mobile video being slower to take off than expected, H.263 nonetheless had a significant impact in conferencing and surveillance applications, as well as in early internet streaming. In particular, H.263 was used to encode Flash Video content for sites such as YouTube and MySpace. RealVideo was initially based on it, and it was specified in several ETSI 3GPP video services. The codec was first designed to be integrated in the H.324 framework for circuit-switched applications, but has been extensively used in H.323 (RTP-based video conferencing) and other IP streaming wrappers.

12.5.2 Picture types and primary features

Macroblock, GOB, and frame format

H.263 supports the following picture formats: sub-QCIF (88×72), QCIF (176×144), CIF (352×288), 4CIF (704×576), and 16CIF (1408×1152) all in YCbCr 4:2:0 format. As with H.261, each picture is divided into a number of groups of blocks (GOBs) defined as an integer number, k, of rows of macroblocks (MBs). $k = 1$ for SQCIF, QCIF, and CIF, $k = 2$ for 4CIF, and $k = 4$ for 16CIF. The structure of an H.263 QCIF picture is shown in Figure 12.6.

Primary features of H.263

Despite using a similar coding architecture to H.261 and MPEG-2, H.263 achieved significantly enhanced performance through the incorporation of a comprehensive

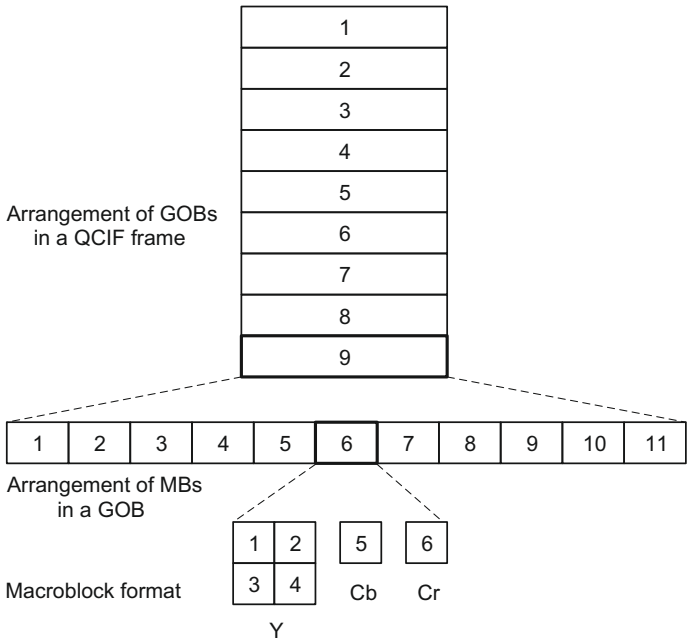

FIGURE 12.6

H.263 picture format (QCIF).

Table 12.3 Primary features of H.263.

Codec feature	Approach
Formats supported	SQCIF, QCIF, CIF, 4CIF, 16CIF
Picture format	Four layers (picture, GOB, MB, and block)
Color sub-sampling	4:2:0 YCbCr
Frame types	I, P, B, (PB)
Intra-coding transform	Block 8 $\times$ 8 DCT
Inter-coding transform	Block 8 $\times$ 8 DCT
Quantizer	Uniform (DC) and deadzone (AC) QP $= 1 \cdots 31$
Motion compensation	Half-pixel, unrestricted vectors, overlapped.
Entropy coding	3-D VLC, Huffman, and arithmetic coding
Loop filter	None
Coding control	Inter/intra, quantizer step size, and transmit/Skip at MB level
Error protection	Provided by transport layer or BCH-FEC

set of advanced features. These are summarized in Table 12.3. The standard is based on a hybrid motion-compensated DCT, using zig-zag scanning with quantization step sizes (QP $= 1 \cdots 31$) changeable at each MB. 3-D VLC tables are used in H.263, incorporating the events {RUN, LEVEL, LAST} where LAST signals the last

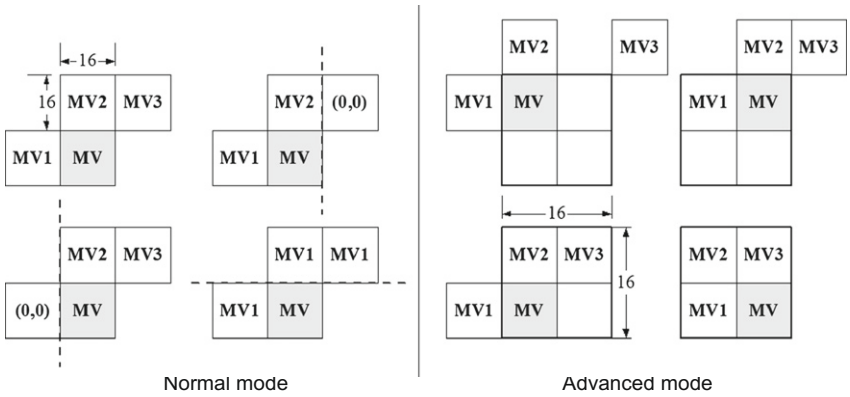

FIGURE 12.7

Motion vector coding in H.263.

non-zero coefficient in the block. This avoids the need for an explicit EOB symbol and provides a compact representation of the 8 × 8 DCT block. H.263 uses motion compensation with half-pixel precision and motion vectors are coded predictively as shown in Figure 12.7.

In terms of PSNR, H.263 can provide up to 3–4 dB improvement over H.261. It does this because of its half-pixel motion prediction but also by incorporating four new optional coding modes:

- **Unrestricted motion vectors:** Motion vectors are allowed to point outside of the reference picture area and the search range is extended to [−31.5, +31.5].
- **Syntax-based arithmetic coding:** The conventional Huffman entropy coder is replaced with an arithmetic coder, enabling fractional wordlengths per encoded symbol.
- **Advanced prediction mode:** This includes two submodes:
  - The use of four MVs per block, i.e. one MV for each 8 × 8 block rather than one per 16 × 16 block. This enables the encoder to better deal with multiple motions within a block.
  - Overlapped Motion Compensation (OMC): Here, each pixel in an 8 × 8 luma block is predicted as a weighted sum of three prediction values. These prediction values correspond to the vector of the current block and two of four predictions from the blocks to the top, bottom, left, and right of the current block. The two blocks selected are those closest to the pixel being computed—e.g. a pixel in the top left quadrant of the block uses the predictions from above and left. This is illustrated in Figure 12.7 (right).

- **PB-frames mode:** In a PB-frame, two pictures are coded as a single unit—a P-frame predicted from the previous P-frame and a B-frame, predicted from both adjacent P-frames using bidirectional prediction. A PB-frame macroblock comprises 12 blocks—six for the predicted P-block and six for the B-block. The benefit of a PB-frame is that motion vectors are not transmitted for the B-blocks, but are instead derived using a scaled version of that for the P-block based on the local temporal activity relative to the corresponding P-block in the previous P-frame. This process enables an effective increase in frame rate without a significant increase in bit rate.

H.263 extensions (H.263+ and H.263++)

H.263 Version 2, also known as H.263+, was standardized in 1998 and extends H.263 with many new modes and features that further improve compression efficiency. H.263++ (Version 3) provided still further enhancements in 2000. The 12 new modes in H.263+ improve coding gain, improve error resilience, enable scalable bit streams, introduce flexibility in picture size and clock frequency, and provide supplemental display capabilities. Many of these new modes were pulled through into the later H.264/AVC standard.

Firstly, several new modes to support error resilience were introduced and these are summarized below. We have covered many of these already in Chapter 11 and they are also reviewed by Wenger et al. [6].

- **Slice Structured Mode (Annex K):** In this mode, the GOB structure is replaced by a set of slices. All macroblocks in one slice can be decoded independently since prediction dependencies are not permitted across slice boundaries.
- **Independent Segment Decoding Mode (Annex R):** A segment boundary acts in the same way as a picture boundary. A segment can be a slice, a GOB, or a collection of GOBs and the shape of a segment must be identical from frame to frame. Independent segments support error resilience as error propagation is eliminated between defined parts of the picture. They also enable special effects in a similar manner to the object planes in MPEG-4.
- **Reference Picture Selection Mode (Annex N):** As described in Chapter 11, this allows flexibility in the choice of reference picture. It is also possible to apply the reference picture selection mode to individual segments rather than to full pictures. This mode supports error resilience if a feedback channel exists, but is also a precursor to the multiple reference frame methods used in H.264 and HEVC.
- **Temporal, SNR, and Spatial Scalability Mode (Annex O):** This mode introduced layering into the H.263 standard to support flexibility in delivery to terminals of different capabilities and in congestion management. As discussed in Chapter 11, a scalable bitstream comprises a base layer and enhancement layers, where the base layer provides an acceptable level of quality which can be further

enhanced by the other layers (in terms of improved signal to noise ratio, improved temporal resolution or improved spatial resolution) if they are available.

Other H.263 modes that support enhanced coding gain are:

- Advanced Intra-coding Mode (Annex I).
- Modified Unrestricted Motion Vectors Mode (Annex D).
- Improved PB-Frames Mode (Annex M).
- Deblocking Filter Mode (Annex J).
- Reference Picture Resampling Mode (Annex P).
- Reduced Resolution Update Mode (Annex Q).
- Alternative Inter VLC Mode (Annex S).
- Modified Quantization Mode (Annex T).

Finally one mode was introduced to provide additional bitstream information:

- **Supplemental Enhancement Information Mode (Annex L):** This enables the incorporation of additional information that may or may not be decodable by a specific decoder. It can contain information such as chroma keying, time segments, picture freezing, and picture resizing.

H.263++ provided further enhancements to H.263 in terms of introducing a range of nine profiles and seven levels similar to those in MPEG-2. The profiles enabled formalization of specific combinations of modes to support different application scenarios (e.g. Baseline and Wireless). The levels enabled matching of coder performance to environmental constraints (e.g. maximum bit rate, spatial resolution, etc.). It also introduced some new modes and revised others—a Data Partitioned Slice Mode (Annex V), Additional Supplemental Enhancement Information (Annex W), and an Enhanced Reference Picture Selection Mode (Annex U).

12.6 MPEG-4

12.6.1 Brief history

MPEG-4 [7] was a very ambitious project that sought to introduce new approaches using object-based as well as waveform-based methods. The MPEG-4 standard was ratified in 2000, as ISO/IEC 14496, potentially offering significant encoding improvements over and above MPEG-2. It was however, in general, found to be too complex. It is interesting to compare the standards documents for H.261 and MPEG-4. The former was 25 pages long while MPEG-4 comprised 500 pages for the visual part alone. Only the Advanced Simple Profile (ASP) was used in practice, and this gained some traction forming the basis for the emerging digital camera technology of the time. The early MPEG-4 part 2 offered little practical advantage over MPEG-

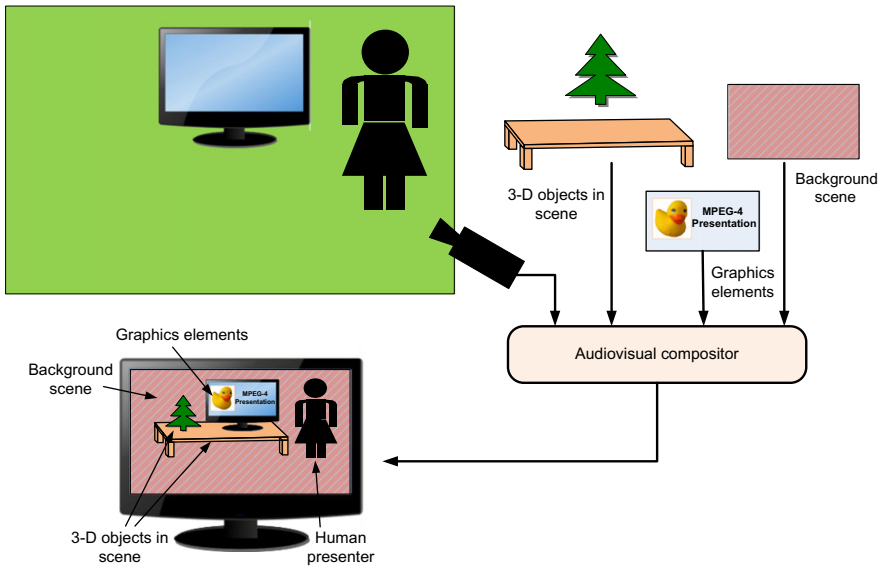

FIGURE 12.8

An example of an MPEG-4 audiovisual scene.

2 or H.263 and was rapidly overtaken by MPEG-4 part 10/AVC, now known as H.264/AVC.

12.6.2 Picture types and primary features

MPEG-4 was aimed at encoding video at lower bit rates and higher video qualities than MPEG-2, effectively creating a multimedia transmission and storage framework. MPEG-4 (part 2) was designed to be suitable for a wide range of video encoding scenarios ranging from the studio and movies to video applications on mobile phones. In this context it focused on object-based as well as waveform-based coding.

MPEG-4 provided a set of coding tools for audiovisual scenes, supporting the coding of arbitrarily shaped objects as shown in Figure 12.8. The scene in this figure represents a 2-D background, a video playing on the screen, a presenter, and 3-D objects such as the table and tree. MPEG-4 was aimed at the independent processing of such audiovisual objects as well as their compositing and allowed modification of both natural and synthetic (computer generated) content.

As with MPEG-2 and H.263++, MPEG-4 initially defined a number of conformance points using a Simple Profile, a Core Profile, and a Main Profile. The Simple and Core Profiles address scene sizes of QCIF and CIF at 64, 128, 384 kbps, and 2 Mbps. The Main Profile is targeted at CIF, ITU-R 601, and HD sizes, with bit rates at 2, 15, and 38.4 Mbps. The standard also incorporated support for sprites, scalability, and error resilience.

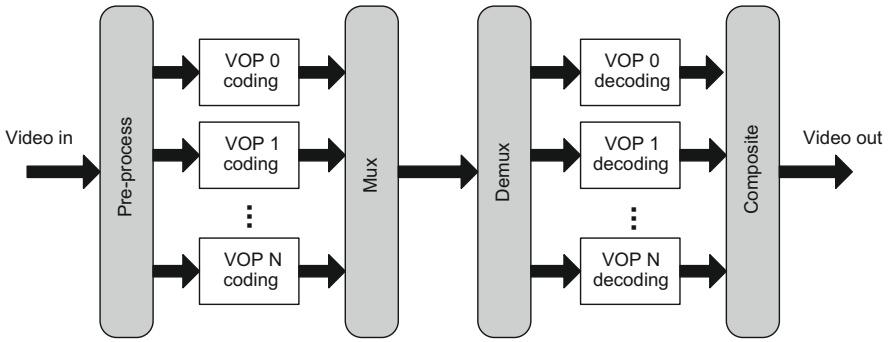

FIGURE 12.9

Generic representation of MPEG-4 Video Object Plane (VOP) coding.

MPEG-4 also provided, for the first time, support for integration with the MPEG-7 content description scheme to facilitate the use of metadata. An excellent overview of MPEG-4 natural video coding is provided by Ebrahimi and Horne [8].

Coding framework

Although we will not devote a lot of time to describing the details of MPEG-4, it is worth briefly putting its aims into context.

MPEG-4 supported the composition or decomposition of video content into visual objects, allowing these to be processed independently as a number of Video Object Planes (VOPs) (Figure 12.9). In order to describe these object planes it was necessary to code their textures, shapes, and motions. Shape coding is done using a binary mask, or a gray-scale alpha channel, allowing transparency. Both motion compensation and DCT-based texture coding have to take account of object boundaries. Error resilience is provided by resynchronization markers, data partitioning, header extension codes, and reversible variable length codes. Scalability is provided for both spatial and temporal resolution enhancement. MPEG-4 also provided scalability on an object basis, with the restriction that the object shape has to be rectangular.

MPEG-4 part 2 Advanced Simple Profile (ASP)

The Advanced Simple Profile was the most commonly used MPEG-4 profile, offering support for interlaced video, B-pictures, quarter-pixel motion compensation, and global motion compensation. The quarter-pixel accuracy was subsequently adopted in H.264/AVC whereas the global motion compensation feature was not generally supported in most implementations. As the reader will notice, ASP does not offer a lot that is not present in H.263; in fact the uptake of H.263 was far more widespread than that of ASP.

Soon after the introduction of ASP, MPEG-4 visual work refocused on part 10 and this became a joint activity with ITU-T under the H.264/AVC banner. This major step forward is the topic of the next section.

12.7 H.264/AVC

12.7.1 Brief history

ITU-T/SG16/Q6 (Video Coding Experts Group) commenced work in 1998 on a project called H.26L. The JVT (Joint Video Team) between VCEG and MPEG (Moving Picture Experts Group) was formed in 2001 to establish a joint standard project known as H.264/MPEG-4-AVC. The final draft for formal approval was produced in March 2003 [9], with the scalable video coding (SVC) extension finalized in 2007.

MPEG-4 (part 10), or H.264/AVC, is by far the most ubiquitous video coding standard to date. It will remain so, probably until 2015 when volume production and infrastructure changes enable a major shift to H.265/HEVC. In the same way that MPEG-2 underpinned the revolution in digital broadcasting, H.264/AVC has played a key role in enabling internet video, mobile services, OTT services, IPTV, and HDTV. H.264/AVC is a mandatory format for Blu-ray players and is used by most internet streaming sites including Vimeo, YouTube, and iTunes. It is used in Adobe Flash Player and Microsoft Silverlight and it has also been adopted for HDTV cable, satellite, and terrestrial broadcasting.

ITU-T, in partnership with ISO/IEC, delivered H.264/AVC in 2004. We have already examined many of the features that contribute to the success and performance of H.264 in previous chapters and we summarize these below.

12.7.2 Primary features

The aim of the H.264/AVC project was to provide:

- **Improved coding efficiency:** A targeted average bit rate reduction of 50% given fixed fidelity, compared to any other video standard.
- **Error robustness:** Tools to deal with packet loss and congestion management.
- **Network friendliness:** Major targets were mobile networks and the internet—separation of coding layer through network abstraction.
- **Adaptation to delay constraints:** To provide low delay modes.
- **Simple syntax specification:** Avoiding an excessive quantity of optional features or profile configurations.

These are all sensible objectives, but the last one is interesting in as much as it acknowledges the over-complex nature of its predecessor.

Let us now consider some of these attributes in more detail. A summary is provided in Table 12.4; although not all the details of a complex codec such as H.264 can be captured in a table of this type, it can be seen that many of the features are similar to those in the extensions to H.263. For further details, the reader is referred to the standards documents [9], or overviews such as that by Wiegand et al. [10]. A very accessible introduction to the standard is provided by Richardson [11].

Table 12.4 Primary features of H.264/AVC.

Codec feature	Approach
Formats supported	QCIF to UHDTV (4k)
Picture format	Four layers (picture, GOB, MB and block)
Color sub-sampling	4:0:0, 4:2:0, 4:2:2, 4:4:4 YCbCr
Partitions	Flexible slice structuring
Frame types	I, P, B, SP, SI
Intra-coding transform	Block 8×8 or 4×4 integer transform
Inter-coding transform	Block 8×8 or 4×4 integer transform
Intra-prediction	Multi-direction, multi-pattern
Quantizer	Scalar with logarithmic control, QP $= 0 \cdots 51$
Motion compensation	Quarter pixel, unrestricted vectors, multiple reference frames
ME block size	16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4
Entropy coding	CAVLC or CABAC
Coding control	Many sophisticated coding control modes—RDO
Loop filter	In-loop non-linear deblocking filter
Error resilience	Slices, FMO, SI, SP frames, multiple reference frames

12.7.3 Network abstraction and bitstream syntax

Figure 12.10 shows the layered structure of H.264/AVC coding. This ensures that the coding layers are independent of the network, relying on protocols as shown to provide mapping to the appropriate video transport mechanism. VCL data can thus easily be mapped to transport layers such as RTP/IP for real-time wired or wireless

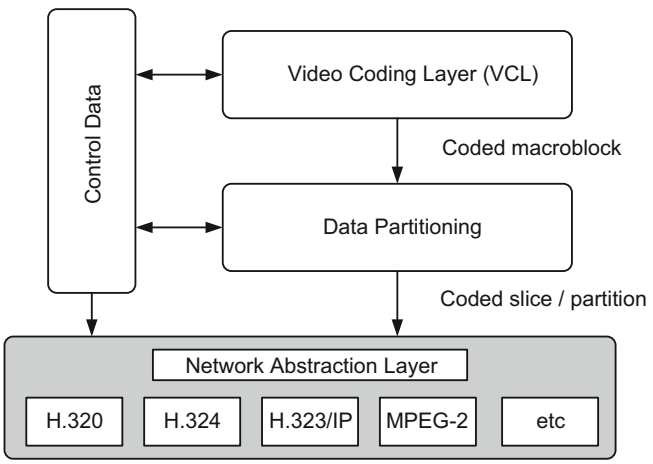

FIGURE 12.10

H.264/AVC layer structure.

internet, MP4 for storage and MMS, H.32X for wireline and wireless conversational services or MPEG-2 systems for broadcasting.

In H.264/AVC, the bitstream is organized hierarchically, comprising the Network Abstraction Layer (NAL) and the Video Coding Layer (VCL). When coded, H.264 information is represented as a series of NAL Units (NALUs). The NALU header indicates the type of NALU which may be a Sequence Parameter Set (SPS), a Picture Parameter Set (PPS), VCL coded slice data, or Supplemental Enhancement Information (SEI). The SPS contains information about the whole sequence such as Profile, Level, frame size, and other characteristics important to the decoder. A PPS contains more localized information, relevant to a subset of frames, such as the number of slice groups, entropy coding mode, whether weighted prediction is used, and other initialization parameters. SEI relates to information that is not essential for decoding but may assist in, for example, buffer management or other non-normative tasks. A diagram showing the hierarchical H.264 syntax is shown in Figure 12.11.

Each sequence starts with an instantaneous decoder refresh (IDR) access unit. This is an intra-coded picture which indicates to the decoder that no subsequent picture will require reference to pictures prior to it. The video slice data corresponds to a sequence of coded macroblocks, each of which contains information relevant to decoding plus the actual encoded residual data. The following describes the MB level information, with reference to Figure 12.11:

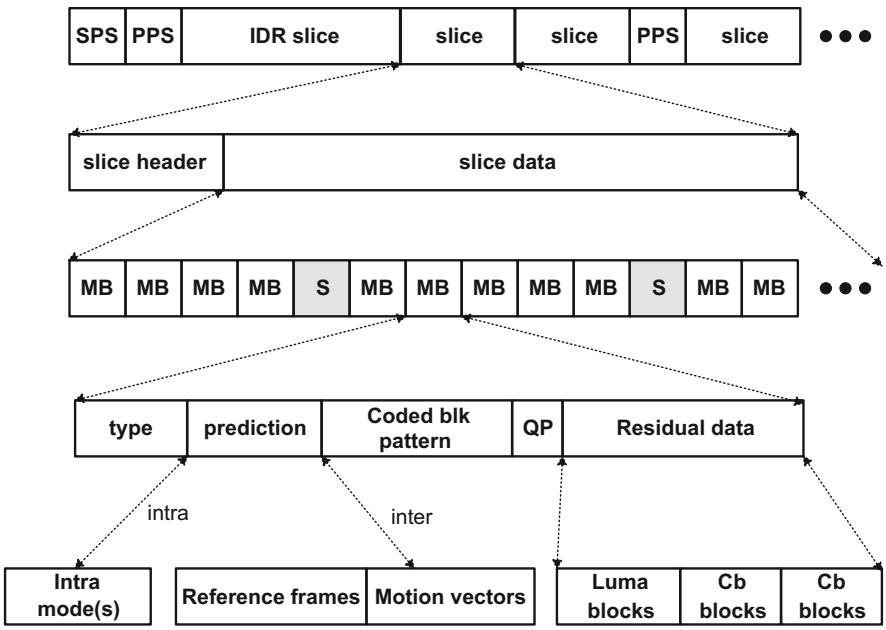

FIGURE 12.11

H.264 syntax.

- **Type:** I, P, B.
- **Prediction:** I-block mode ($16 \times 16, 8 \times 8$ or 4×4), P-block partition (16×16, 16×8, 8×16 or 8×8 (contains sub-block partitions)), B-block partition (as for P-block), choice of reference frames, and motion vector data. Note: P- and B-blocks may indicate a SKIP condition to signal that no residual data is sent.
- **Coded Block Pattern:** Indicates for which blocks coefficients are present.
- **Quantization Parameter:** QP value for the MB.
- **Residual data:** Coded residual data where indicated by the CBP parameter.

12.7.4 Pictures and partitions

Picture types

H.264/AVC supports the processing of both progressive and interlaced formats in a consistent manner, as a single unit. The fact that the frame is progressive or interlaced is indicated in the PPS and adaptive field-frame encoding can be used (either to encode a frame as a single unit or as two fields) but this has no impact on the way the picture is decoded via the decoder buffer.

H.264/AVC supports I-frames, P-frames, and B-frames. B-frames are referred to as generalized B-frames in H.264 in that they can be predicted from one or from two reference frames. H.264/AVC supports a very wide range of picture formats from QCIF to 4K UHDTV.

Slices and slice groups

As discussed in Chapter 11, slices can be arranged into slice groups (SG) where each SG can contain one or more slices. The number of slices is signaled by a PPS parameter. In cases where there is more than one slice group, the mapping arrangement of slices must be signaled using a value in the range of 0–6. As examples, SG type 0 is interleaved, SG type 1 refers to the dispersed allocation, and SG type 2 represents the foreground and leftover case. Depending on the slice group map type, additional syntax elements are required to enable the decoder to remap the received macroblocks to the slice groups used at the encoder. We introduced the various types of slice used in H.264/AVC in Section 12.7.3, and selected slice group patterns are illustrated in Figure 12.12.

Slice groups in H.264 support independent decoding, for example as required in multiple description coding and error concealment (both described in Chapter 11). Switching slices are also included in H.264. SI and SP slices direct a decoder to jump to a different point in the stream or to a different stream. These features are useful for video streaming where operations such as fast-forward are used.

Blocks

The macroblock in H.264 is based on a 16×16 sample arrangement, comprising luminance and chrominance residual data. The difference with H.264/AVC is that this structure can be subdivided in various ways according to the available prediction modes. This is discussed in more detail below.

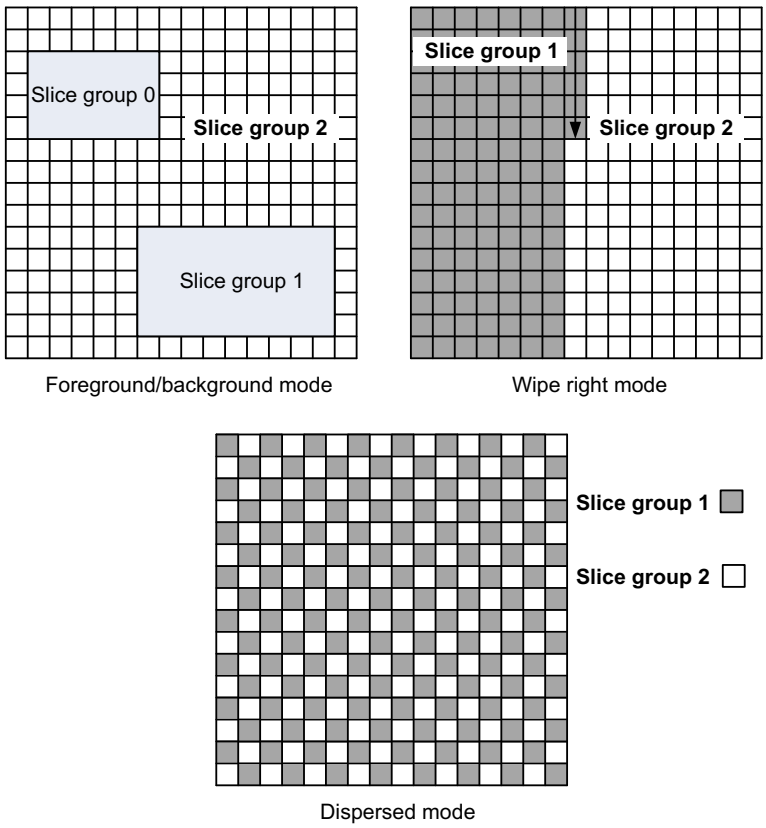

FIGURE 12.12

Example slice structures.

12.7.5 The video coding layer

Intra-coding

- **I_PCM mode:** This is the most basic H.264 mode and switches off all of the normal prediction and transform modes associated with the standard. This does not offer any compression but can be useful in high quality regimes where the PCM approach may offer better rate–distortion performance than a prediction–transform approach.
- **Intra-prediction:** H.264/AVC has benefited from the introduction of a range of intra-prediction modes. These offer significant benefits when there is a high level of local detail, particularly with oriented features such as edges. H.264 supports intra-prediction for 4×4, 8×8 (High profiles only), and 16×16 blocks. Coding of a given block is performed with reference to a subset of samples from previously coded blocks that lie to the left and above the current block. These modes were discussed in detail in Chapter 9.

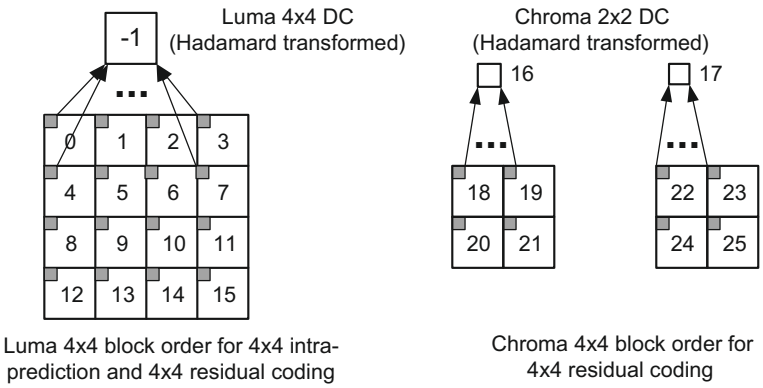

FIGURE 12.13

Hadamard transformation of prediction residuals and coded block ordering.

- **Transforms:** In the case of intra_4 × 4 modes, the residual signal after prediction is transformed using the core 4 × 4 integer transform as described in Chapter 9. For the case of intra_16 × 16 modes, each of the 16 4 × 4 residual blocks is transformed as shown in Figure 12.13 and each of the 16 DC coefficients are further transformed using a 4 × 4 Hadamard transform. A similar approach is taken for the DC coefficients of the transformed chroma residuals, but this time a 2 × 2 Hadamard transform is applied.

Inter-coding

- **Variable block sizes:** We saw in Chapter 9 how the use of multiple block sizes for predictive coding of frames can provide significant benefits in dealing with complex textures and motions, offering up to 15% bit rate saving over the use of fixed block sizes. The range of block sizes available in H.264/AVC is shown again for convenience in Figure 12.14 and an example of a typical block decomposition is given in Figure 12.15. In order to make the most benefit of this flexibility, however, significant complexity must be added to the encoding process in terms of rate–distortion optimization (see Chapter 10).
- **Transforms:** H.264/AVC adopts the 4 × 4 DCT-like integer transform (described in Chapters 5 and 9) for residual coding in its baseline and main profiles. This is complemented by the use of a 2 × 2 Hadamard transform on the DC values for the case of chroma signals. For the High profiles, the standard also includes an 8 × 8 transform as this is often better suited to content with higher spatial resolutions.
- **Multiple reference frames:** Up to 16 frames or 32 fields (depending on profile) can be buffered in the H.264 reference buffer. The reference buffer does not have to contain the 16 most recent frames; frames can be updated according to their utility, provided that the encoder and decoder remain synchronized. For further details on MRF-ME, the reader is referred back to Chapter 9.

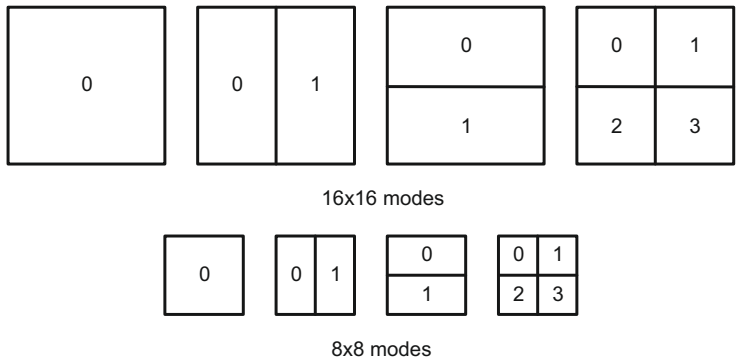

FIGURE 12.14

Variable block sizes supported by H.264/AVC. Top: 16×16 modes 1–4. Bottom: 8×8 modes 1–4.

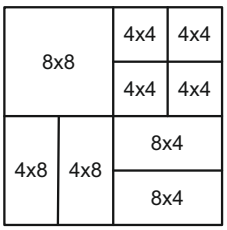

FIGURE 12.15

Example H.264/AVC macroblock partition for inter-coding.

- **Prediction structures:** The H.264 coding regime enables a much more flexible approach to temporal prediction. For example, in the case of low delay, low memory constraints, an IPPP $\cdots$ I structure is most suitable, as offered by the baseline profile. If some delay is tolerable, along with the availability of B-frames, and multiple reference frames, then improved compression performance can be achieved. An example of this is shown in Figure 12.16, where a hierarchical prediction structure is imposed. Using this approach it has been demonstrated [12] that, if the quantization parameter is increased in a controlled fashion, with layer, then more efficient compression performance can be achieved.
- **Weighted prediction:** This is a useful tool, introduced in H.264, that enables the weighting/offsetting of prediction sample values in B-frames. This provides coding gains in the case of accelerating motion or during scene cuts.
- **Quarter-pixel prediction:** H.264 benefits from the availability of subpixel motion compensation to quarter-pixel accuracy. This has been estimated to provide up to 20% saving over the case of integer-pixel accuracy. A comprehensive description of the H.264/AVC sub-pixel motion estimation and compensation process was provided in Chapter 9. H.264/AVC uses a five-tap filter for half-pixel interpolation and a simple two-tap filter for local refinement to quarter-pixel accuracy.

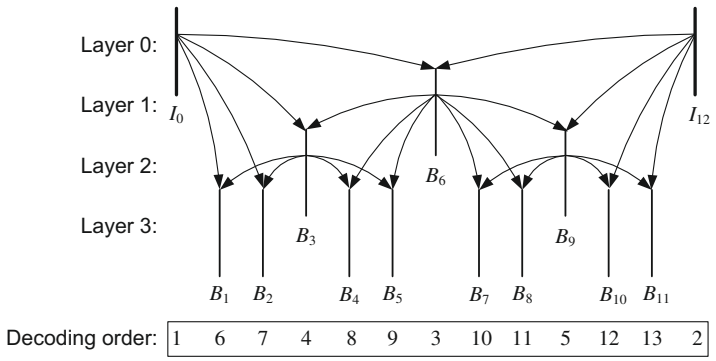

FIGURE 12.16

Hierarchical B-pictures in H.264.

Deblocking operations

The H.264 deblocking filter has been assessed to reduce the coded bit rate for the same subjective quality by up to 10%. It improves subjective visual and objective quality of the decoded picture. This highly content-adaptive non-linear filter removes blocking artifacts and does not unnecessarily blur the visual content. Its structure and operation were described in Chapter 9.

Variable length coding

- **Exp-Golomb coding:** This is a highly structured and simple variable length coding method, used almost universally in H.264 for all symbols apart from transform coefficients.
- **CAVLC:** Context-Adaptive VLC is a relatively low complexity but effective method, used as the Baseline and Extended Profiles for entropy coding of transform coefficients. Local contexts are used at the encoder to select between different VLC tables according to the local statistics, in this case the number of non-zero coefficients in neighboring blocks.
- **CABAC:** CABAC is a more complex, arithmetic coding-based method that is only supported in the H.264 Main and High Profiles. It uses local contexts to adjust its conditioning probabilities. CABAC has been found to reduce bit rate by between 10 and 20% compared to CAVLC, dependent on content type and quantization level.

Further details on all of the above entropy coding methods can be found in Chapter 7.

Coder control

Coder control is a non-normative part of H.264/AVC. It is however a key element in achieving the optimum performance from the standard, since without good RDO, many of the advanced features (e.g. block size selections) included will not deliver

major savings. The goal of coder control is to select what parts of the video signal should be encoded using which methods and with what parameter settings. Often a Lagrangian optimization approach is adopted where the distortion measure, D, is based on sum of squared difference and the rate, R, includes all bits associated with the decision including header, motion, side information, and transform data. Further details on the RDO methods used in H.264/AVC can be found in Chapter 10.

12.7.6 Profiles and levels

H.264/AVC started with a relatively small number of profiles and levels. The number has however grown considerably, since its introduction, to reflect its popularity in a widening range of applications and the demand for higher quality profiles to deal with emerging formats. The standard now supports 21 profiles ranging from the simplest Constrained Baseline Profile (CBP) and Baseline Profile (BP) to the High 4:2:2 Profile (Hi422P—used for studio work) and the High 4:4:4 Predictive Profile (Hi444PP). The Main Profile (MP) is typically used for standard definition (4:2:0) DVB broadcasting. The Extended Profile (XP) is used in streaming and the High Profile (HiP) is used in Blu-ray players and for DVB HDTV delivery.

H.264 also supports a range of decoder constraints on memory and computational power in terms of its 17 levels, specifying parameters such as decoded bit rate, decoded picture buffer size, picture size, and processing speed in terms of numbers of macroblocks per second. A selection of the H.264 profiles are shown in Figure 12.17 and a good summary of all of profiles and levels is provided in Ref. [13].

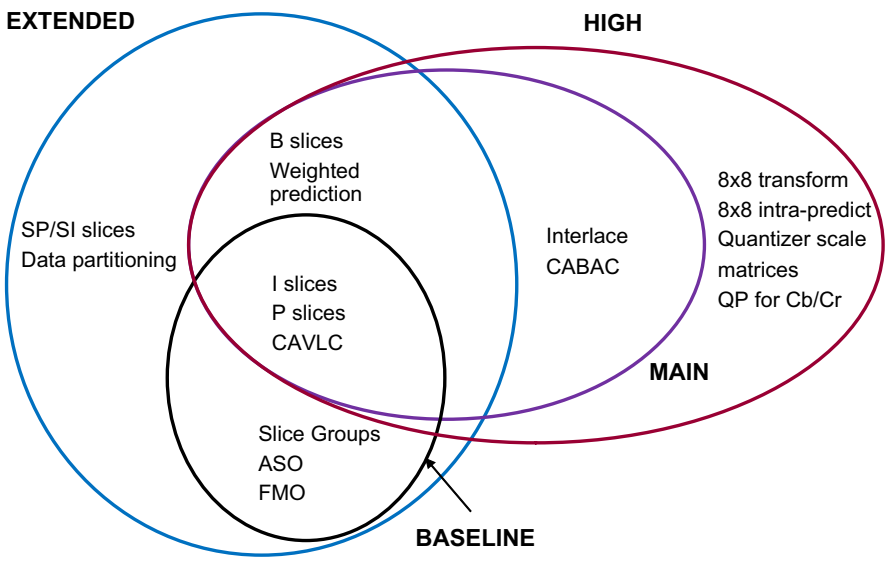

FIGURE 12.17

Selected H.264 profiles.

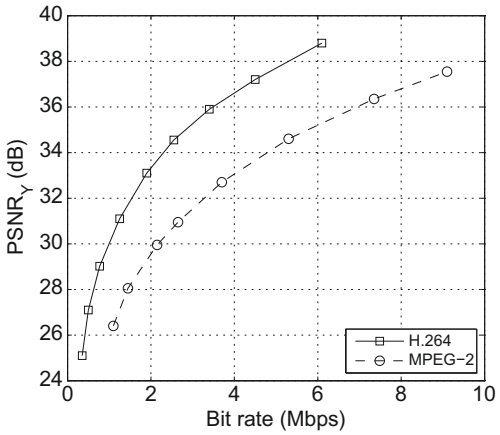

FIGURE 12.18

Typical H.264/AVC (MP) performance relative to MPEG-2 for standard definition entertainment content.

12.7.7 Performance

The performance of H.264/AVC relative to previous standards is summarized in Ref. [14]. An example RD profile is shown in Figure 12.18 for the case of an entertainment application. This shows approximately 50% saving in bit rate for the same PSNR value between H.264/AVC and MPEG-2. A more recent comparison including HEVC is provided by Ohm et al. [15] (see Table 12.7).

12.7.8 Scalable extensions

A scalable extension to H.264/AVC (SVC) was approved in 2007. As described in Chapter 11, bitstream scalability is useful for providing graceful degradation of performance in the context of dynamic channel conditions. SVC contains five extra profiles, offering a layered approach to scalability. This creates an embedded bitstream providing SNR, spatial and/or temporal scalability, where packets can be discarded in an orderly fashion to manage congestion or terminal limitations. As can be observed from Figure 12.16, the use of hierarchical B-frames is a very useful tool in providing temporal scalability. The reader is referred to Ref. [16] for more details on SVC.

12.7.9 Multiview extensions

Finally, to cater for the predicted boom in 3-D stereoscopic video content and broadcasting, a multiview extension (MVC) was approved in 2009 comprising three additional H.264 profiles. As its name suggests, MVC supports the coding of content acquired from multiple cameras. Its primary focus is stereoscopic (3-D) video, but it also supports free-viewpoint and multiview 3-D applications, television, and multiview

3-D television. The Stereo High profile is used in 3-D Blu-ray devices. For further information on MVC, the reader is referred to Refs. [17,18].

12.8 H.265/HEVC

12.8.1 Brief background

HEVC, formally known as ISO/IEC MPEG-H Part 2 and ITU-T H.265, is the most recent outcome from collaborative working between ISO/IEC MPEG and ITU-T VCEG. The standard was approved in April 2013 [19] and, with a patent licensing framework in place and the announcement of several HEVC-enabled products, it is poised to make a major impact on the video industry. HEVC offers the potential for up to 50% compression efficiency improvement over AVC and will go some way to providing a solution to the explosion in mobile video and the need to support broadcast and download services with ever-increasing quality and experience levels.

HEVC is targeted at the same application portfolio as H.264/AVC, but with a specific focus on bit rate reduction for increased video resolutions and on support for parallel processing as well as loss resilience and ease of integration with appropriate transport mechanisms. An excellent introduction to the range of HEVC features is provided by the papers in a special issue of the *IEEE Transactions on Circuits and Systems for Video Technology* published in December 2012, in particular the overview by Sullivan et al. [20] and the performance comparisons by Ohm et al. [15]. For full details, the reader should consult the standards documentation [19].

12.8.2 Primary features

As we will see later, HEVC reduces bit rate requirements roughly by half compared to existing codecs while retaining image quality. It does this using the proven hybrid approach that has served us well since H.261, but with the addition of several new and important features. While some of these add complexity to the codec, many actually offer simplicity compared to H.264/AVC and facilitate parallel processing, compatible with recent hardware developments. HEVC, like previous standards, can trade off computational complexity, compression rate, robustness to errors, and processing delay time, according to application requirements, and this range of operating points is defined by its profile and level structure. A major focus for HEVC is the next generation of HDTV displays and acquisition systems which feature progressively scanned frames, higher frame rates, and resolutions up to UHDTV.

In the following sections, we examine more closely the video coding and syntactic structure of HEVC. The reader will see that a large amount of HEVC technology has been pulled through from H.264/AVC, with some H.264 features omitted and other new and modified features added where appropriate to improve performance. The primary new features in HEVC relate to (i) its block structure, where super-macroblocks known as Coding Tree Units (CTUs) are used, where the size of a CTU can range up to 64×64 samples; (ii) the way in which these CTUs are partitioned into

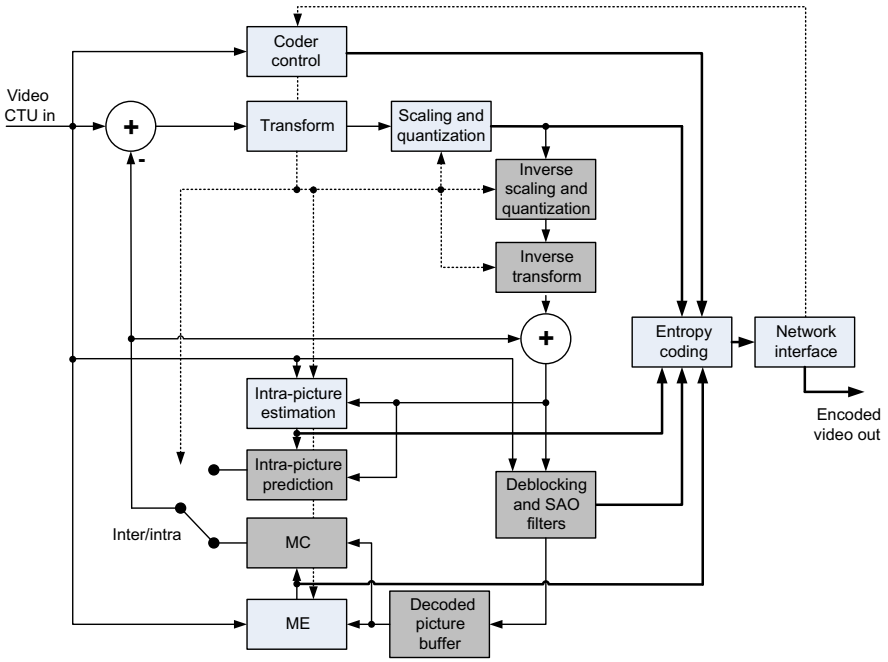

FIGURE 12.19

HEVC video encoder architecture.

variable sized coding, transform, and prediction blocks using a quadtree structure; (iii) significantly enhanced intra-prediction, supporting 35 modes; (iv) the use of Sample Adaptive Offset (SAO) in conjunction with the deblocking filter in the inter-prediction loop to reduce banding and ringing artifacts.

The architecture of the HEVC codec, illustrating many of these features, is shown in Figure 12.19. This will be described in more detail in the following sections.

12.8.3 Network abstraction and high level syntax

NAL units

The high level syntax of HEVC retains a number of the elements used in H.264/AVC to provide network abstraction and the basis for error resilience. NALUs are still employed as the basic interface to the transport mechanism and HEVC supports 63 NALU (VCL and non-VCL) types [19,20].

Slice structures

Slices enable independent decoding and assist with resynchronization and error robustness. The slice structure in HEVC has been largely pulled through from H.264/AVC and the reader is referred to Chapters 9 and 11 for a more detailed description.

Parameter sets

The same parameter sets as in H.264/AVC are employed, supplemented by a new Video Parameter Set (VPS). VPS can include metadata to describe the general characteristics of coded layers, such as could be useful in future scalable extensions. Information in the NAL Unit header also supports the identification of temporal layers.

Reference picture sets and reference picture lists

It is important to manage the content of the Decoded Picture Buffer (DPB) to ensure synchronization between encoder and decoder. The retained set of pictures is referred to as a Reference Picture Set (RPS). HEVC supports two reference picture lists, RPL0 and RPL1. In the case of unidirectional prediction, a reference picture can be selected from either list and in the case of bidirectional prediction, one picture must come from each list. Some improvements have been made in HEVC to make the identification of the RPS more robust than in H.264/AVC.

12.8.4 Pictures and partitions

HEVC supports a very wide range of picture sizes and frame rates as illustrated in the level examples in Table 12.5. The basic picture types supported in HEVC are identical to those in H.264/AVC, with the small exception of a modified version of the IDR frame known now as a Clean Random Access (CRA) picture.

An HEVC Coding Unit (CU) can contain up to three blocks (arrays of luma and chroma samples) and any information required to decode these blocks. Each CU is subdivided into Prediction Units (PUs) and these are coded using either intra- or inter-prediction. The appropriate prediction parameters (inter- or intra-) are signaled within each PU. A Residual Quadtree (RQT) is then used to divide the CU into transform units (TUs) within which the residual information is coded. These units all contain coding blocks (CBs), prediction blocks (PBs), and transform blocks (TBs) for the constituent luma and chroma components, as described above. This structure is described in more detail in Ref. [21]. The decomposition of the HEVC picture is described in more detail below.

Table 12.5 Example HEVC levels.

Level	Max. luma sample rate	Max. luma picture rate	Max bit rate (kbps)	Example resolution
1	552,960	36,864	128	176 × 144@15
3	16,588,800	552,960	6,000	720 × 576@30
5	267,386,880	8,912,896	160,000*	3,840 × 2160@60
6.2	4,278,190,080	35,651,584	800,000*	7680 × 4320@120

** High Tier specification, others are Main Tier.*

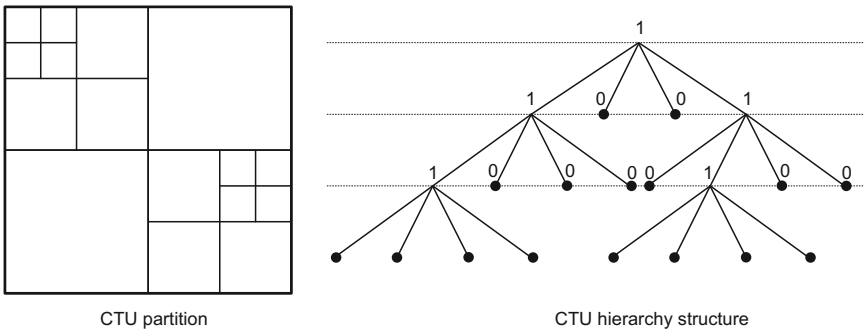

FIGURE 12.20

CTU partitioning in HEVC—an example.

Coding tree units (CTUs) and coding tree blocks (CTBs)

HEVC replaces the macroblock of earlier standards with an $L \times L$ CTU where $L = 16, 32, 64$. This provides more efficient coding for higher resolution formats. The CTU is made up of luma and chroma CTBs.

Quadtree CTU structure

CTBs can be partitioned into Coding Units (CUs) comprising luma and chroma Coding Blocks (CBs) with smaller block sizes using a quadtree decomposition. The size of the largest coding unit (LCU) is 64×64. The manner in which the CTU is decomposed is not specified in the standard (it is non-normative) but nonetheless forms an important part of the encoder's RDO processing. An example of CTU partitioning is given in Figure 12.20. Each CU is further partitioned into Transform Units (TUs) and Prediction Units (PUs). This approach provides the encoder with much more flexibility to optimize block partitions according to the video content.

Prediction units (PUs)

A much larger variety of block decompositions for intra- and inter- coding is supported in HEVC, from 4×4 to 64×64.

Transform units (TUs)

Prediction residuals, whether from intra- or inter-coding, are transformed using integer transforms similar to the DCT, with sizes from 4×4, 8×8, 16×16, and 32×32. For the case of intra-prediction residuals, a 4×4 integer sine transform is also available as this provides better performance on less correlated residuals.

Random access points (RAPs) and clean random access (CRA) pictures

The concept of CRA pictures is the same as that for IDR frames in H.264/AVC. CRAs generalize this by embodying independently coded pictures, where following frames

have no dependence on frames received before the CRA picture. CRA pictures can be placed at locations where random access is required, known as RAPs.

12.8.5 The video coding layer (VCL)

Similar to H.264, HEVC uses a hybrid coding structure combining motion compensation with block transform and entropy coding. Several innovations have been incorporated into HEVC that provide significant gains in performance and these are described below.

Intra-coding

In a similar manner to H.264/AVC, the boundary pixel values from adjacent decoded blocks can be used as a basis of spatial prediction. Whereas H.264/AVC supports only eight prediction modes, this has been significantly enhanced in HEVC with 35 modes including 33 directional modes (Figure 12.21) plus surface fitting and DC prediction. Further details on the computation of the predictions for these modes can be found in Ref. [19]. The amplitude of the planar surface in mode 0 is computed based on horizontal and vertical slopes derived from the reference pixel values in the surrounding blocks. The DC mode (1) is computed in the same way as for H.264/AVC—based on the average of the block boundary reference values.

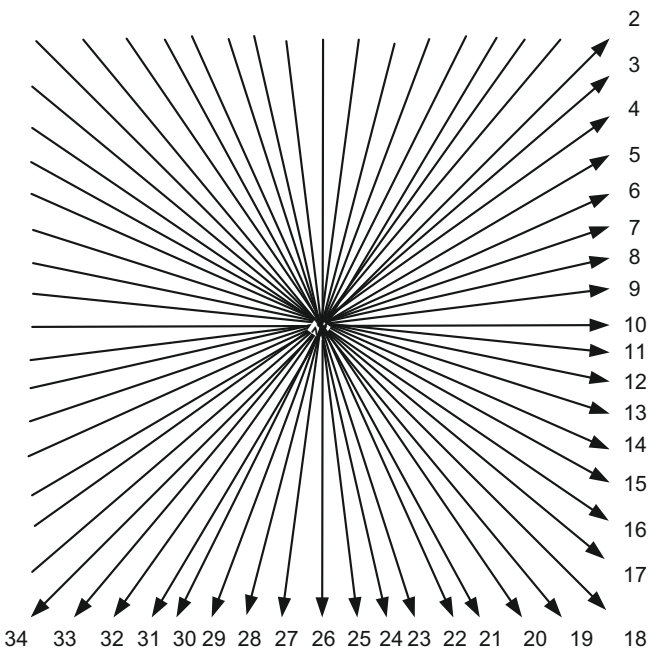

FIGURE 12.21

HEVC intra-prediction modes (mode 0 = planar and mode 1 = DC).

HEVC also extends the concept of reference and boundary sample smoothing. These operations help to reduce residual and reconstruction edges by filtering either the reference samples (from surrounding blocks) or the boundary samples (within the predicted block) prior to prediction.

When intra-modes are used the prediction block (PB) size is the same as that for the coding block (CB), except for the smallest CB size where a further partition into four PB quadrants of 4×4 is permitted. After prediction, residuals are coded using a forward integer transform, quantized, and entropy coded.

Inter-coding

The approach to inter-prediction and coding is similar to H.264/AVC. Multiple reference frames are stored in the Decoded Picture Buffer and are used in a similar way to H.264/AVC, supporting both uni predictive and bipredictive modes. As with H.264/AVC, weighted prediction can be employed. Some modifications included in HEVC are as follows:

- **Subpixel motion compensation:** Quarter-pixel precision is used for estimating motion during temporal prediction. Local luma interpolation to half-pixel resolution is provided by an eight-tap filter, rather than the six-tap filter in H.264/AVC. A seven-tap filter is then used for quarter-sample interpolation rather than the two-tap filter used in H.264/AVC. These are applied separably and (unlike H.264) without intermediate rounding. This improves precision and simplifies the prediction architecture. The filter coefficients for interpolation are given in equations (12.1) and (12.2). Further details on the use of these can be found in Refs. [19,20].

$$h_h[i] = \{-1, 4, -11, 40, 40, -11, 4, -1\}/64 \tag{12.1}$$

$$h_q[i] = \{-1, 4, -10, 58, 17, -5, 1\}/64 \tag{12.2}$$

- **Motion vector prediction:** An Advanced Motion Vector Prediction mode (AMVP) is used for initialization and derivation of likely motion vector candidates. Also, improved skip modes are included along with a capability of merging MVs between adjacent blocks.
- **Coding block partitioning into prediction blocks:** In the case of inter-prediction, luma and chroma CBs can be divided into one, two, or four PBs (again the latter only for the smallest CB size). These partitioning options are illustrated in Figure 12.22.

Transforms in HEVC

Residual data after prediction is coded with a separable integer transform whose structure depends on prediction mode and block size. The transform block sizes supported in HEVC are: 4×4, 8×8, 16×16, and 32×32. As for the case of H.264/AVC, the transform basis functions are derived from the DCT. The transform is defined as a 32×32 matrix of basis function and the smaller transform sizes are obtained by sub-sampling this larger matrix. For example, the 16×16 matrix is

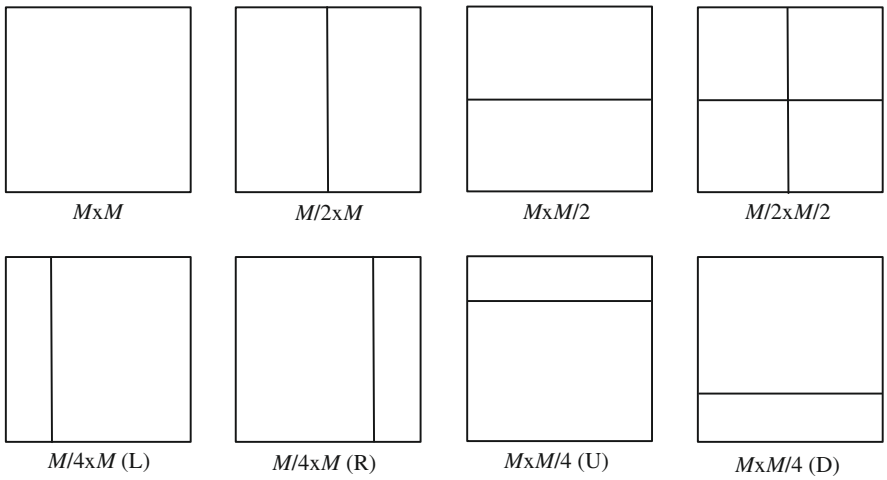

FIGURE 12.22

CB partitions for inter-prediction PBs.

derived from the 32 × 32 matrix by taking the first 16 values from rows 0, 2, 4, etc. A similar sub-sampling is used to produce 8 × 8 and 4 × 4 transform matrices. As an example, the 8 × 8 transform is given below.

$$\mathbf{A}_{8\times 8} = \begin{bmatrix} 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 \\ 89 & 75 & 50 & 18 & -18 & -50 & -75 & -89 \\ 83 & 36 & -36 & -83 & -83 & -36 & 36 & 83 \\ 75 & -18 & -89 & -50 & 50 & 89 & 18 & -75 \\ 64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 \\ 50 & -89 & 18 & 75 & -75 & -18 & 89 & -50 \\ 36 & -83 & 83 & -36 & -36 & 83 & -83 & 36 \\ 18 & -50 & 75 & -89 & 89 & -75 & 50 & -18 \end{bmatrix} \tag{12.3}$$

In the case of intra-coded 4 × 4 residuals, an alternative integer transform is used—derived from the DST. It has been observed that the statistics of intra residuals better fit the basis functions of the DST since the error values increase with distance from the boundary samples used to predict them. The 4 × 4 transform matrix is thus:

$$\mathbf{A}_{4\times 4\mathrm{I}} = \begin{bmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{bmatrix} \tag{12.4}$$

Quantization in HEVC

HEVC employs a similar quantization method for transform coefficients to that used in H.264. It uses uniform quantization with a range of scaling matrices available for each block size.

Coefficient scanning

The scanning of coefficients in an HEVC TB [22] is based on the use of multiples of 4 × 4 blocks for all TB sizes. The three coefficient scanning options available to represent each Coefficient Group (CG: a group of 16 consecutive coefficients) are shown in Figure 12.23. In the case of intra-blocks, a new method referred to as Mode-Dependent Coefficient Scanning (MDCS) is used, where the mode closest to the orientation of the intra-prediction mode is selected. For inter-coded blocks, only the diagonal mode is used. For larger inter-blocks, coefficients are scanned from the least significant to the most significant as shown in Figure 12.23 for the case of an 8×8 TB. In contrast to the zig-zag scanning used in H.264/AVC and previous standards, this simplified approach enables better matching of scanning to coefficient contexts. In terms of coding efficiency, improvements of around 3.5% compared to H.264/AVC have been reported [22].

A further modification in HEVC relative to AVC is related to the signaling of the last significant coefficient in a scan. As we have seen, signaling the last significant coefficient reduces the number of coded bins by eliminating the need to explicitly code runs of trailing zeros. In HEVC the position of the last significant coefficient is coded explicitly followed by the flags to indicate significant coefficients.

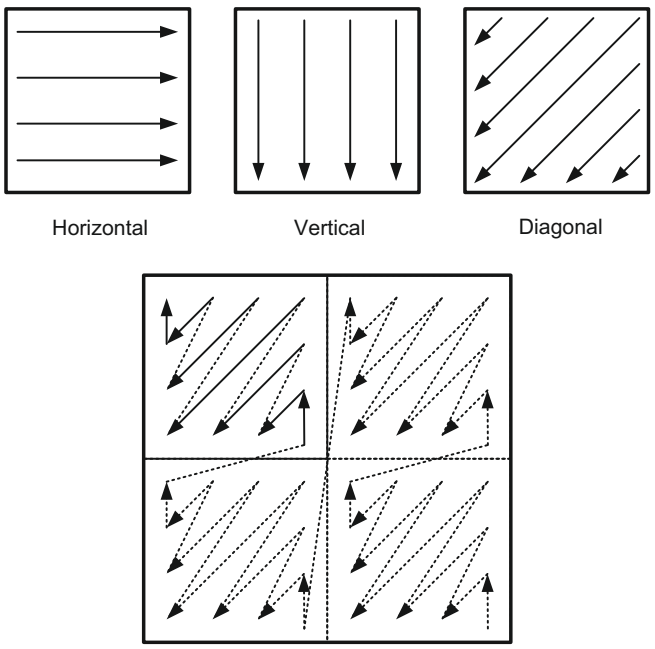

FIGURE 12.23

Coefficient scanning in HEVC.

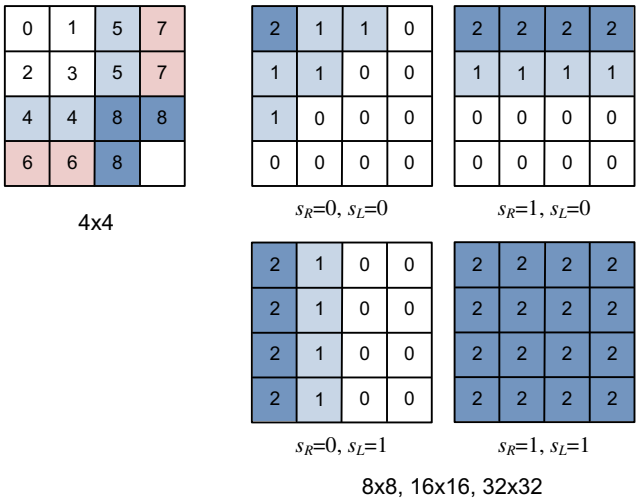

FIGURE 12.24

Significance flag contexts for 4 × 4 and 8 × 8 HEVC transform blocks.

Contexts and significance

As with H.264/AVC, HEVC signals the significance of a TB using a Coded Block Flag (CBF) which indicates whether the TB contains any non-zero coefficients. The context of a CBF depends on the level of the block in the quadtree decomposition. The significance maps within HEVC are coded in a way that exploits the sparsity of the significance map, using two passes. Firstly the significance of coefficient groups is encoded and then the significance of the coefficients within them. The coefficient significance contexts in HEVC for 4 × 4 and 8 × 8 blocks are shown in Figure 12.24. The significance flags are coded for each coefficient between the last and the DC based on a context model that depends on its location and the significance of its neighbors. In the case of 4 × 4 TBs, coefficients are grouped according to frequency and the distribution, where coefficients with similar distributions are grouped together.

For the case of 8 × 8, 16 × 16 and 32 × 32 TBs in HEVC, context modeling is again based on position but according to the templates shown in Figure 12.24 (right). One of the four templates is selected according to the values of s_R and s_L where s_R and s_L correspond to the coded subblock flag for the right-hand and lower sub-blocks respectively.

Variable length coding with CABAC

CABAC, as described in Chapter 7, is a powerful arithmetic coding method that was supported in the H.264/AVC Main and High Profiles. Because of its performance advantages, HEVC has adopted CABAC for all entropy coding purposes. As with H.264/AVC, CABAC in HEVC uses local contexts to adjust its conditioning probabilities. CABAC has been improved in HEVC through:

1. Better context modeling: based on exploitation of HEVC's tree-structured approach.
2. Reducing the number of context coded bins: although the number of contexts is fewer than in H.264/AVC, its performance is better.
3. Introducing an enhanced bypass mode: more extensive use of bypass is made to reduce the amount of data that needs to be processed using CABAC contexts, resulting in a throughput increase.

In-loop filters

HEVC includes two stages of in-loop filtering prior to updating its decoded picture buffer: a Deblocking Filter (DBF) and a Sample Adaptive Offset (SAO) process.

- **Deblocking Filter:** The deblocking operation in HEVC is very similar to that for H.264/AVC which was described in Chapter 9. It is slightly simplified to make it more amenable to parallel processing. DBF is applied to all TU and PU boundaries, except when they coincide with picture or slice boundaries. In the case of 4×4 blocks the DBF is simplified by only using it when block boundaries coincide with 8×8 boundaries. This simplifies processing generally with no noticeable degradation in performance. In the case of HEVC, only three deblocking strengths are available rather than the five with H.264/AVC.
- **Sample Adaptive Offset:** SAO is a new feature in HEVC that is invoked after DBF, in order to increase picture quality through a reduction in banding and ringing artifacts. It is a non-linear adjustment that adds offsets to samples based on a look-up table created at the encoder (and transmitted to the decoder) using histogram analysis of signal amplitudes. Two SAO modes are available: *edge offset mode* and *band offset mode*. The first operates on a CTB basis using one of four gradient operators to compare a sample with adjacent values in a given orientation. Five conditions can result according to the relative value of the sample compared to its neighbors. Consider, for example, the case of a horizontal orientation, if a sample p has a left neighbor n_0 and a right neighbor n_1, then the regions are classified according to Table 12.6. In the second case, *band offsetting*, the offset is applied directly based on the sample amplitude. The full amplitude range is divided into 32 bands and the values of samples in four consecutive bands are modified using transmitted offsets. This can help to reduce banding artifacts in smooth areas.

Table 12.6 SAO edge classification.

EdgeIdx	Condition	Case
0	Exceptions to cases (1...4)	Monotonic
1	$p < n_0$ and $p < n_1$	Local min.
2	$p < n_0$ and $p = n_1$ or $p < n_1$ and $p = n_0$	Edge
3	$p > n_0$ and $p = n_1$ or $p < n_1$ and $p = n_0$	Edge
4	$p > n_0$ and $p > n_1$	Local max.

12.8.6 Profiles and levels

Version 1 of HEVC specified three profiles, Main, Main 10, and Main Still Picture.

Main profile

The Main Profile supports a bit depth of 8 bits per sample and 4:2:0 (and 4:0:0) chroma sampling, employing the features described in the previous subsections.

Main 10 profile

The Main 10 profile supports bit depths up to 10 bits and up to 4:2:2 chroma sampling. In recent tests, it has been demonstrated that the Main 10 profile outperforms the Main Profile with a bit rate reduction of approximately 5% for identical PSNR values.

Main Still Picture profile

HEVC also includes, like H.264, a still image coding profile, which is a subset of the Main Profile. Performance comparisons based on MOS and PSNR scores have provided indications that HEVC still image coding outperforms JPEG 2000 in terms of bit rate reduction by approximately 20% based on PSNR and 30% for MOS scores. Likewise it has been shown to outperform JPEG by 40% and 60%.

Levels

In the same way as previous standards, HEVC also supports a range of levels that impose parameter limits that constrain decoder performance. There are currently 13 levels associated with HEVC. Some examples are shown in Table 12.5.

12.8.7 Extensions

At the time of writing, a number of HEVC range extensions and five additional profiles are being worked on. These include a Main 12, Main 4:2:2 10, Main 4:2:2 12, Main 4:4:4 10, and Main 4:4:4 12. The extended ranges in terms of color sampling and bit depths are obvious from their titles.

12.8.8 Performance gains for HEVC over recent standards

An excellent comparison of coding standards is provided by Ohm et al. [15], where a comprehensive comparison between HEVC and previous standards is reported. We reproduce their results here in Table 12.7, noting that even greater improvements were

Table 12.7 Comparison of video coding standards for entertainment applications. Average bit rate savings are shown for equal objective quality values measured using PSNR.

Video standard	Relative bit rate savings (%)		
	H.264/AVC	MPEG-4	MPEG-2
HEVC MP	35.4	63.7	70.8
H.264/AVC HP	X	44.5	55.4
MPEG-4 ASP	X	X	19.7
H.263 HLP	X	X	16.2

reported for interactive applications. As a rule of thumb, we have stated previously that video coding standards have, since the introduction of H.120 in 1984, delivered a halving of bit rate for the equivalent video quality every 10 years. This is evidenced by the results in Table 12.7.

12.9 Other de-facto standards and proprietary codecs

A number of other video codecs exist. While these are worthy of mention, we will not cover them in detail here.

12.9.1 VP9

Perhaps foremost is Google's VP9. VP9 is a successor to VP8 and is open-source and royalty-free. VP8 was created by On2 Technologies, who were purchased by Google in 2010. YouTube, also owned by Google, uses Adobe Flash Player and H.264/AVC, but there is a reported move toward VP9. In 2013 a version of the Google Chrome web browser was released that supports VP9 decoding. VP9 is also likely to find favor in some cloud-based applications.

12.9.2 VC-1

The Microsoft WMV 9 codec was a proprietary codec but was in 2006 adopted as international standard SMPTE 421M or VC-1. VC-1 is a recognized format for Blu-ray devices.

12.9.3 RealVideo

RealVideo comprises a series of proprietary video compression formats developed by RealNetworks for their players. The current version, rv40, is thought to be based on H.264/AVC.

12.9.4 Dirac

Dirac was developed by BBC Research and Development as a royalty-free alternative to the mainstream standards. Dirac employs wavelet compression rather than block transforms such as the DCT, and offers good performance for HDTV formats and beyond. An I-frame-only version, Dirac Pro, has been used by the BBC internally for studio and outside broadcast applications and was standardized by SMPTE as VC-2 in 2010. A portable version, known as Schrödinger, also exists.

12.10 Summary

We have examined in this chapter the primary features of all major video compression standards from H.120 through H.261, H.263, and MPEG-2, to the current H.264/AVC standard and the new H.265/HEVC codec. Although these have been largely built on the same hybrid motion-compensated transform architecture, continued innovations over the past three decades or so have delivered, on average, a halving of bit rate

for the equivalent video quality every 10 years. While this is impressive progress, the demand for high volumes of high quality video content keeps growing and new approaches will continue to be needed to fulfill future requirements. Some of the potential candidate solutions are considered in the next chapter.

References

[1] S. Okubu, Reference model methodology—A tool for the collaborative creation of video coding standards, Proceedings of the IEEE 83 (2) (1995) 139–150.
[2] CCITT/SG XV, Codecs for Videoconferencing Using Primary Group Transmission, Rec. H.120, CCITT (now ITU-T), 1989.
[3] International Telecommunication Union-Telecommunication (ITU-T), Recommendation H.261, Video Codec for Audiovisual Services at px64 kbit/s, Version 1, 1990; Version 2, 1993.
[4] ITU-T and ISO/IEC JTC 1, Generic Coding of Moving Pictures and Associated Audio Information—Part 2: Video, ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG-2 Video), Version 1, 1994.
[5] ITU-T, Video Coding for Low Bitrate Communication, ITU-T Rec. H.263, Version 1, 1995, Version 2, 1998, Version 3, 2000.
[6] S. Wenger, G. Knorr, J. Ott, F. Kossentini, Error resilience support in H.263+, IEEE Transactions on Circuits and Systems for Video Technology 8 (7) (1998) 867–877.
[7] ISO/IEC JTC 1, Coding of Audio-Visual Objects—Part 2: Visual, ISO/IEC 14496-2 (MPEG-4 Visual), Version 1, 1999, Version 2, 2000, Version 3, 2004.
[8] T. Ebrahimi, C. Horne, MPEG-4 natural video coding: an overview, Signal Processing Image Communication 15 (2000) 365–385.
[9] ITU-T and ISO/IEC JTC 1, Advanced Video Coding for Generic Audiovisual Services, ITU-T Rec. H.264 and ISO/IEC 14496–10 (AVC), Version 1, 2003, Version 2, 2004, Versions 3, 4, 2005, Versions 5, 6, 2006, Versions 7, 8, 2007, Versions 9, 10, 11, 2009, Versions 12, 13, 2010, Versions 14, 15, 2011, Version 16, 2012.
[10] T. Wiegand, G. Sullivan, G. Bjøntegaard, A. Ajay Luthra, Overview of the H.264/AVC video coding standard, IEEE Transactions on Circuits and Systems for Video Technology 13 (7) (2003) 560–576.
[11] I. Richardson, The H.264 Advanced Video Compression Standard, second ed., Wiley, 2010.
[12] H. Schwartz, D. Marpe, T. Wiegand, Analysis of hierarchical B-pictures and MCTF, in: International Conference on Multimedia and Expo, 2006, pp. 1929–1932.
[13] <http://en.wikipedia.org/wiki/H.264/MPEG-4_AVC>, 2013 (accessed August 2013).
[14] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, T. Wedi, Video coding with H.264/AVC: tools, performance, and complexity, IEEE Circuits and Systems Magazine 4 (1) (2004) 7–28.
[15] J.-R. Ohm, G. Sullivan, H. Schwartz, T. Tan, T. Wiegand, Comparison of the coding efficiency of video coding standards—including High Efficiency Video Coding (HEVC), IEEE Transactions on Circuits and Systems for Video Technology 22 (12) (2012) 1669–1684.
[16] H. Schwarz, D. Marpe, T. Wiegand, Overview of the scalable video coding extension of the H.264/AVC standard, IEEE Transactions on Circuits and Systems for Video Technology 17 (9) (2007) 1103–1120.

[17] P. Merkle, A. Smolic, K. Müller, T. Wiegand, Efficient prediction structures for multiview video coding, IEEE Transactions on Circuits and Systems for Video Technology 17 (11) (2007) 1461–1473.
[18] A. Vetro, T. Wiegand, G.J. Sullivan, Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard, Proceedings of the IEEE 99 (4) (2011) 626–642.
[19] Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 ISO/IEC 23008-2 and ITU-T Recommendation H.265, High Efficiency Video Coding (HEVC), January 2013.
[20] G. Sullivan, J.-R. Ohm, W. Han, T. Wiegand, Overview of the High Efficiency Video Coding (HEVC) standard, IEEE Transactions on Circuits Systems, Video Technology 22 (12) (2012) 1648–1667.
[21] I. Kim, J. Min, T. Lee, W. Han, J. Park, Block partitioning in the HEVC standard, IEEE Transactions on Circuits and Systems for Video Technology 22 (12) (2012) 1697–1706.
[22] J. Sole, R. Joshi, N. Nguyen, T. Ji, M. Karczewicz, G. Clare, F. Henry, A. Duenas, Transform coefficient coding in HEVC, IEEE Transactions on Circuits and Systems for Video Technology 22 (12) (2012) 1765–1777.