the price of an increased coding delay. This advantage in coding efficiency disappears with increasing fidelity of the coded parameters.

FIT

The encoder may allow the decoder to extrapolate the values of some FAPs from the transmitted FAPs [5.83]. Alternatively, the decoder can specify the interpolation rules using FIT. A FIT allows a smaller set of FAPs to be sent for facial animation. This smaller set can then be used to determine the values of other FAPs, using a rational polynomial mapping between parameters. For example, the top inner-lip FAPs can be sent and then used to determine the top outer-lip FAPs. The inner-lip FAPs would be mapped to the outer-lip FAPs using a rational polynomial function that is specified in the FIT.

Integration of Face Animation and Text-to-Speech (TTS) Synthesis

A block diagram showing the integration of TTS synthesizer into an MPEG-4 face animation system is shown in Figure 5.56. Synchronization of a FAP stream with TTS synthesizers using the TTSI is only possible if the encoder sends timing information. This is due to the fact that a conventional TTS system driven by text only behaves as an asynchronous source. Given a TTS stream that contains text in binary form, the MPEG-4 TTSI decoder decodes the text and prosody information according to the interface defined for the TTS synthesizer. The synthesizer creates speech samples that are handed to the compositor. The compositor presents audio and, if required, video to the user. The second output interface of the synthesizer sends the phonemes of the synthesized speech as well as the start time and duration information for each phoneme to the phoneme/bookmark-to-FAP converter [5.77, 5.84]. The converter translates the phonemes and timing information into FAPs that the face rendered uses in order to animate the face model. The precise methods of how the converter derives visemes from phonemes is left to the implementation of the decoder. This allows a coarticulation model at the decoder that uses the current, previous and next phonemes in order to derive the current mouth shape. Bookmarks in the text of TTS are used to animate facial expression and non-speech related parts of the face [5.77, 5.85]. The start time of a bookmark is derived from its position in the text. When the TTS finds a bookmark in the text, it sends this bookmark to the

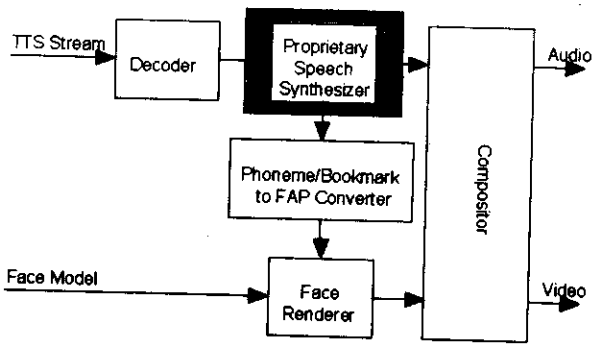


Figure 5.56 Integration of TTS synthesizer into an MPEG-4 face animation system. ©1999 ISO/IEC.

phoneme/bookmark-to-FAP-converter at the same time as it sends the first phoneme of the following word. The bookmark defines the start point and duration of the transition to new FAP amplitude.

BIFS for Facial Animation

In MPEG-4 the scene description information is represented using a parametric methodology [5.39]. The description consists of an efficiently encoded hierarchy (tree) of nodes with attributes and other information, including event sources and targets [5.86]. Leaf nodes in the tree correspond to particular audio or visual objects, and intermediate nodes perform grouping, transformation and other operations. The MPEG-4 scene description framework is partly based on the VRML, which has been significantly extended to address streaming and synchronization issues. To offer complete support for face and body animations, BIFS defines a set of face and body nodes. The most important BIFS nodes for facial animation are shown in Figure 5.57. In order to use face animation in the context of MPEG-4 Systems, a BIFS scene graph has to be transmitted to the decoder. The minimum scene graph contains a face node and an FAP node. The FAP decoder writes the amplitude of the FAP into fields of the FAP node. The FAP node might have the children viseme and expression, which are FAPs requiring a special syntax. This scene graph would enable an encoder to animate the proper face model of the decoder. If a face model is to be controlled from a TTS system, an audio source node needs to be attached to the face node. In order to download a face model to the decoder, the face node requires an FDP node as one of its children. This FDP node contains the position of the feature points in the downloaded model, the scene graph of the model and the face definition table, the face definition mesh and face definition transform nodes required to define the action caused by FAPs. Figure

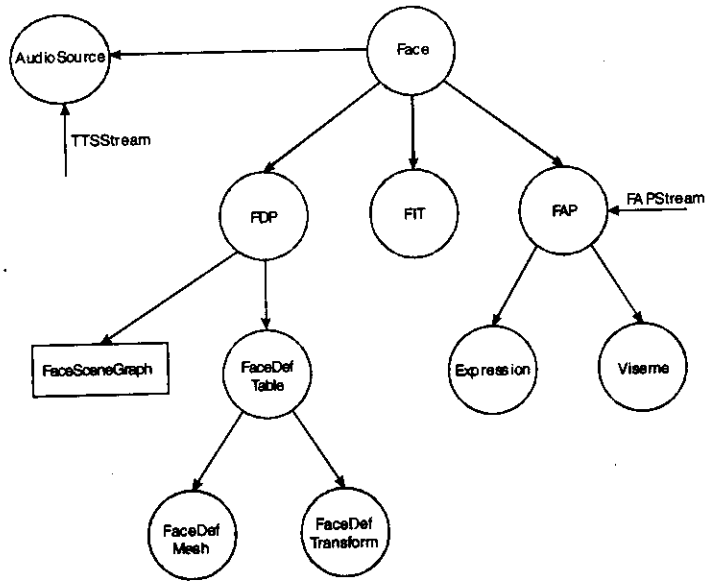


Figure 5.57 Nodes of a BIFS scene graph that are used to describe and animate a face. ©1999 ISO/IEC.

5.57 shows how these nodes relate to each other. The face graph contains the scene graph of the static face. Here, it is assumed that the streams are already decoded. The FIT node, when specified, allows a set of unreceived FAPs to be defined in terms of a set of received FAPs. The standard specifies processes that involve the reading of node values, for example, FAPs and then the writing of output values to nodes in the face hierarchy.

2D Mesh Coding

MPEG-4 version 1 supports 2D uniform or content-based (nonuniform) mesh representation of arbitrary visual objects, which includes an efficient method for animation of such meshes [5.87, 5.88].

In content-based video compression, manipulation, and indexing, the shape, motion and texture of each arbitrary-shaped VO need to be modeled and encoded independently. In the MPEG-4 video verification model, the shape of a VOP is represented by a bitmap, called alpha plane (binary or gray scale), and the text (color) of VOP is represented by a texture plane. The values of pixels in the texture plane are defined in the corresponding alpha plane pixel as nonzero. The motion of the VO is represented by a translational block model. A 2D mesh object is a representation of 2D deformable geometric shape, with which synthetic VOs may be created during a composition process at the decoder by spatially piecewise warping of existing VOPs or still texture objects. The instances of mesh objects at a given time are called Mesh Object Planes (MOPs). The geometry of MOPs is coded lossless [5.77]. Temporally and spatially predictive techniques and variable-length coding are used to compress 2D mesh geometry. The coded representation of a 2D mesh object includes representation of its geometry and motion.

The mesh model offers a versatile alternative, whereby the motion and shape of a VO are modeled in a unified framework, which can also be extended to the 3D object modeling. The 2D mesh modeling corresponds to nonuniform sampling of the motion field at a number of salient feature points (node points) along the contour and interior of the VO. Content-based mesh modeling may require transmission of geometry overhead, unlike block modeling, which requires no such overhead. If the first content-based mesh is designed based on the original VOP, the initial mesh geometry has to be transmitted in addition to all node motion vectors. The mesh geometry needs to be transmitted only once, because subsequent forward motion estimation is based on the most recently updated mesh [5.88].

The 2D mesh representation of a VO enables the following:

- VO compression
- VO manipulation
- Content-based video indexing

Mesh modeling may improve compression efficiency in two ways. Namely, the mesh model provides better motion compensation than the translational block model, which may result in less blocking artifacts at lower bit rates. Alternatively, we can choose to transmit texture

maps only at selected key frames and to animate these texture maps without sending any prediction error image for the intermediate frames. This is also known as self-transfiguration of selected key frames using 2D mesh information. VO manipulation deals with augmented reality, synthetic-object transfiguration/animation and spatiotemporal interpolation. On the other hand, in content-based video indexing, mesh representation does the following: enables animated key snapshots for a moving visual synopsis of objects; provides accurate object trajectory information that can be used to retrieve VOs with specific motion and provides vertex-based object shape representation, which is more efficient than the bitmap representation for shape-based object retrieval.

VO Tracking

A VO tracking procedure is shown as a block diagram in Figure 5.58. The 2D mesh design is also presented. Dotted boxes denote optional steps. A feedback loop is designed for the initial VOP. The initial mesh can have a uniform or content-based geometry. A 2D triangular mesh or a MOP is a planar graph that partitions a VOP or its bounding box into triangular patches. The vertexes of each patch are called node points. A 2D mesh object, which consists of a sequence of MOPs, is compactly represented by mesh geometry at some key (intra) MOPs and mesh motion vectors at all other (inter) MOPs. The mesh geometry refers to the location of the node points in the key MOPs. The 2D mesh animation is accomplished by propagating the 2D mesh defined on key MOPs using one motion vector per node point per object plane until the next key MOP. Both mesh geometry and motion (animation) information are predictively coded for an efficient binary representation [5.77]. Mesh-based motion modeling differs from block-based motion modeling (that is used in natural video object coding) in that the triangular patches overlap neither in the reference frame nor in the current frame. Triangular patches in the current frame are mapped onto triangular patches in the reference frame. On the other hand, the texture inside each patch in the reference frame is warped onto the current frame using a parametric mapping, such as affine mapping, as a function of the node point motion vectors. This process is called texture mapping, which is an integral part of mesh animation [5.77]. A uniform mesh is designed over a rectangular region, which is generally the bounding box of the VOP [5.78]. It is specified in terms of five parameters: the number of nodes in the horizontal and vertical directions, the horizontal and vertical dimensions of each rectangular cell in half pixel units and the triangle split code that specifies how each cell is divided into two triangles. A content-based mesh may be designed to fit exactly on the corresponding VOP. The procedure consists of three steps: (1)

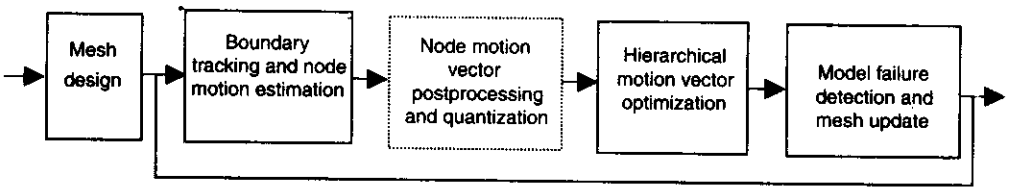


Figure 5.58 The 2D mesh design and tracking procedure. ©1999 ISO/IEC.

approximation of the VOP contour by a polygon through selection of N_b boundary node points, (2) selection of N_i interior node points and (3) Delaunay triangulation to define the mesh topology [5.89]. There are various methods for approximation of arbitrary-shaped contours by polygons [5.74].

Motion data of the 2D mesh may represent the motion of a real VO for natural VO compression and manipulation applications or may be synthetic for animation of a still texture map. In the former case, the motion of a natural VO may be estimated by forward mesh tracking. The latter requires special-purpose tools and/or artistic skills. In forward mesh tracking, we search the current VOP for the best matching locations of the node points of the previous (intra or inter) mesh, thus tracking image factures until the next intra MOP. This procedure applies for both uniform and content-based meshes. Various techniques have been proposed for node motion estimation for forward mesh tracking. The simplest method is to form blocks that are centered around the node points and then employ a closed-form solution or block matching to find motion vectors at the node points independently [5.87, 5.90]. Alternatively, hexagonal matching [5.90] and closed-form matching [5.91] techniques find the optimal motion vector at each node under the parametric warping of all constraints at the expense of more computational complexity [5.92]. Another method is iterative gradient-based optimization of node point locations, taking into account that image features and mesh provide significantly improved performance and robustness in enforcing constraints to avoid foldovers [5.93, 5.94].

2D-Mesh Object Encoder/Decoder

A simplified architecture of an encoder/decoder supporting a 2D-mesh object is depicted in Figure 5.59. A mesh analysis module extracts the 2D mesh data, which is then encoded by the mesh encoder. The coded mesh representation is embedded in a BIFS ES. At the receiver, the 2D mesh decoder is invoked automatically by the BIFS animation code [5.77]. A 2D mesh object can be used together with a VO or a still-texture object encoder/decoder.

Mesh data consists of a list of node locations (x_n,y_n) where n is the node index (n=0,...,N-1) and a list of triangles t_m where m is the triangle index (m=0,...,M-1). Each triangle t_m is specified by a triplet <i,j,k> of the indexes of the node points that are vertexes of that triangle. The syntax of the compressed binary representation of intra and inter MOPs and the semantics of the

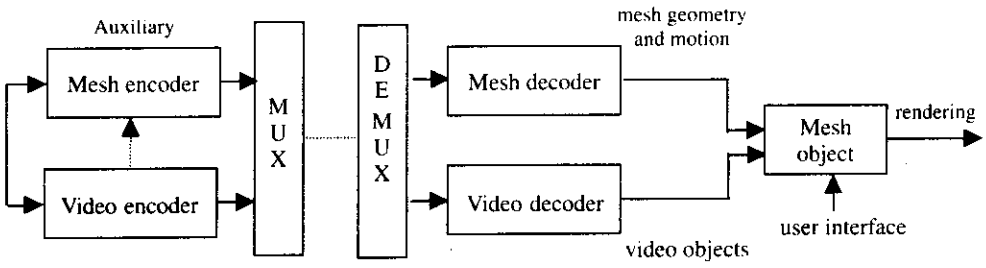


Figure 5.59 Scalable architecture with a video and mesh encoder. ©1999 ISO/IEC.

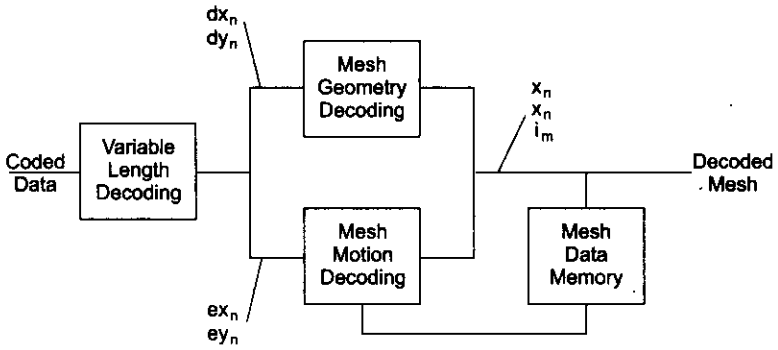


Figure 5.60 Block diagram of 2D mesh object decoding. ©1999 ISO/IEC.

decoding process is normative in MPEG-4. A block diagram of the decoding process is shown in Figure 5.60.

Variable length decoding takes the coded data and decodes either node point location data or node point motion data. Node point location data is denoted by dx_n, dy_n and node point motion data is denoted by ex_n, ey_n, where n is the node point index (n=0,...,N-1). Next, either mesh geometry decoding or mesh motion decoding is applied. Mesh geometry decoding computes the node point locations from the location data and reconstructs a triangular mesh from the node point locations. Mesh motion decoding computes the node point motion vectors from the motion data and applies these motion vectors to the node points of the previous mesh to reconstruct the current mesh.

The reconstructed mesh is stored in the mesh data memory so that it may be used by the motion decoding process for the next mesh. Mesh data consists of node point locations (x_n, y_n) and triangles t_m, where m is the triangle index (m=0,...,M-1) and each triangle t_m contains a triplet $<i,j,k>$, which stores the indexes of the node points that form the three vertexes of that triangle.

After the mesh object start code has been decoded, a sequence of MOPs is decoded until a mesh object end code is detected. The new mesh flag of the MOP class determines whether the data that follows specifies the initial geometry of a new dynamic mesh or if it specifies the motion of the previous mesh relative to the current mesh in a sequence of meshes. We describe the decoding of mesh geometry first; then, we describe the decoding of mesh motion. In this specification, a pixel-based coordinate system is assumed, with the x-axis pointing to the right from the origin, and the y-axis pointing down from the origin.

Encoding or decoding of mesh geometry. The flag mesh type code specifies whether the topology of an intra MOP is uniform or Delaunay. A 2D uniform mesh can be viewed as a set of rectangular cells, where each rectangle is split into two triangles. Five parameters are used to specify the node point locations and the topology of a uniform mesh. The top-left node point of the mesh always coincides with the origin of a local coordinate system. The first two parameters specify the number of nodes in the horizontal and vertical directions of the mesh, respectively. The next

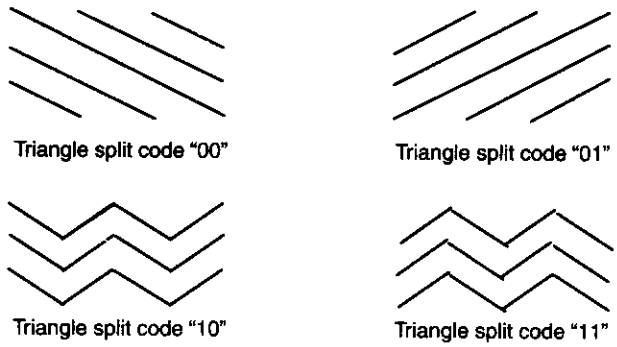


Figure 5.61 Types of uniform mesh topology [5.39]. ©1998 ISO/IEC.

two parameters specify the horizontal and vertical sizes of each rectangular cell in half pixel units.This completes the layout and dimensions of the mesh. The last parameter specifies how each rectangle is split to form two triangles. Four choices are allowed. Types of uniform mesh topology are illustrated in Figure 5.61. Code "00" is top left to right bottom, and code "01" is bottom left to top right. Code "10" alternates between top left to bottom right and bottom left to top right. Finally, code "11" alternates between bottom left to top right and top left to bottom right.

Example 5.14 An example of 2D uniform mesh is given in Figure 5.62. A uniform 2D mesh is specified by five parameters, where no_mesh_node_hor is equal to 5, no_mesh_nodes_vert is equal to 4 and mesh *rect*size_hor and mesh_rect_size_vert are specified as shown. The triangle_split_code is equal to "00."

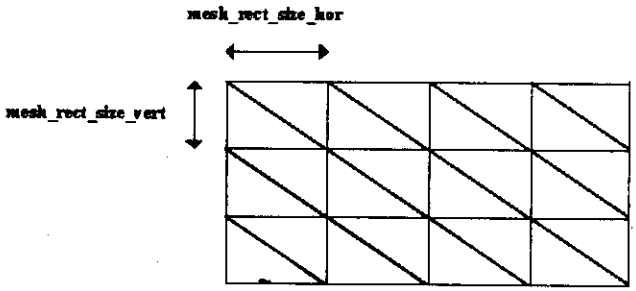


Figure 5.62 A uniform 2D mesh [5.62]. ©1998 ISO/IEC.

A 2D Delaunay mesh is specified by the following parameters:

- The total number of node points N
- The number of node points N_b that are on the boundary of the mesh
- The coordinated $p_n = (x_n, y_n)$, $n = 0,\ldots,N\text{-}1$, of all node pints

The origin of the local coordinate system is assumed to be at the top-left corner of the bounding box of the mesh. The number of nodes in the interior of the mesh N_i can be computed as

$$N_i = N - N_b \tag{5.1}$$

The first node point, $p_0 = (x_0, y_0)$ is decoded directly where the coordinates x_0 and y_0 are specified with respect to the origin of the local coordinate system. All other node points are computed by adding the decoded values dx_n and dy_n to the x and y coordinates, respectively, of the last decoded node point

$$x_n = x_{n-1} + dx_n \quad \text{and} \quad y_n = y_{n-1} + dy_n \tag{5.2}$$

The first N_b node point coordinates that are encoded or decoded must correspond to the boundary nodes in order to allow their identification without additional overhead. After receiving the first N_b locations, the decoder can reconstruct the boundary of the mesh by connecting each pair of successive boundary nodes, as well as the first and the last, by straight-line edge segments. Decoded node points and reconstruction of a mesh boundary are illustrated in Figure 5.63.

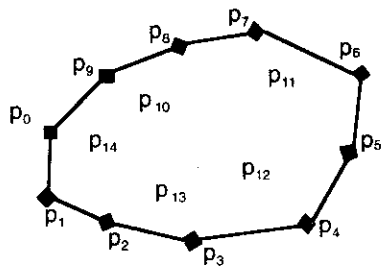


Figure 5.63 Decoded node points and reconstruction of a mesh boundary [5.62]. ©1998 ISO/IEC.

The next N_i coordinate values define the interior node points. The mesh is reconstructed by applying constrained Delaunay triangulation to all node points. The boundary polygon forms the constraint. Each triangle $t_k = \langle p_l, p_m, p_n \rangle$ of a constrained Delaunay triangulation satisfies the property that the circumcircle of t_k does not contain any node point p_k visible from all three vertexes of t_k. A node line between them falls entirely inside or exactly on the constraining polygonal boundary. A decoded triangular mesh obtained by constrained Delaunay triangulation is shown in Figure 5.64.

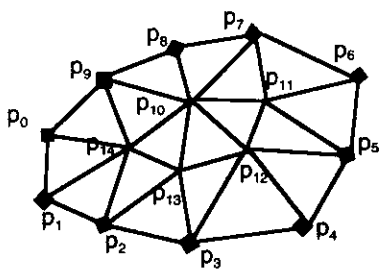


Figure 5.64 Decoded triangular mesh obtained by constrained Delaunay triangulation [5.62]. ©1998 ISO/IEC.

Encoding or decoding of mesh motion. An inter MOP is defined by a set of 2D motion vectors, $v_n = \{vx_n, vy_n\}$, that are associated with each node point p_n of the previous MOP. We can then reconstruct the locations of node points in the current MOP by propagating the corresponding node p_n of the previous MOP. The triangular topology of the mesh remains the same until the next intra MOP. Node point motion vectors are decoded predictively, that is, the components of each motion vector are predicted using those of two previously decoded points.

Recent and future research in the area of 2D mesh-based coding covers occlusion-adaptive mesh modeling and mesh mosaicking, hierarchical mesh modeling and extensions to 2D and 3D mesh modeling.

5.5.5 MPEG-4 Audio

There are two groups of sound-coding tools in MPEG-4: the natural tools which allow digital audio to be compressed and transmitted and the synthetic tools, which allow parametric descriptions of sounds to be transmitted and used to drive synthesis upon receipt [5.95, 5.96, 5.97, 5.98]. The natural audio tools enable the compressed transmission of speech and wideband audio at ranges from 6 Kb/s for low-bit-rate speech coding to 64 Kb/s per channel for high-quality multichannel sound. At the upper end of this range, the MPEG-4 tools have been demonstrated in psychoacoustic evaluation to be nearly perceptually transparent [5.99]. MPEG-4 has three main audio-coding tools. The General Audio (GA) coder allows the transmission of high-quality broadband multichannel signals, such as music, at bit rates from 16 to 64 Kb/s channel. This coder is a state-of-the-art, scalable version of well-known perceptual compression techniques [5.100]. It is based on the MPEG-2 advanced audio-coding standard [5.101] with additional improvements in quality and functionality for MPEG-4. The CELP coder uses codebook excitation linear prediction techniques to enable highly compressed speech coding between 16 and 24 Kb/s [5.102, 5.103]. The parametric speech coding is based on the harmonic vector excitation coding method and provides toll-quality speech down to 6 Kb/s [5.104].

MPEG-4 has two synthetic audio coders. One provides an interface to text-to-speech systems. The so-called TTSI receives a bit stream that contains phonemic and prosodic data and controls an external speech synthesizer [5.98]. No particular method of speech synthesis is specified in the standard. Only the interface and bit stream format are standardized in MPEG-4 TTSI. The second is a very general music-and-sound-effects synthesis tool set called Structured Audio (SA). The SA coder allows transmission of sound synthesis algorithms in a new Music V language called Structured Audio Orchestra Language (SAOL) [5.105]. Transmitting sound as synthesis algorithms is a recent development, and MPEG-4 is the first standard to make use of this capability [5.106]. The music language SAOL is also important to the audio-compositing tools. In MPEG-4, SAOL is used for downloading user-definable effects processing algorithms. The convergence between the coding techniques for structured audio and effects processing in MPEG-4 is one of the important aspects of the standard [5.107]. The sounds transmitted and decoded using the MPEG-4 audio tools are not immediately played back for the listener. Rather, thay are composited together into a soundtrack. The soundtrack, not the component parts, is pre-

sented. The composition process may be very simple, as in direct linear mixing, or very complex, with arbitrary effects-processing code downloaded and multiple sound objects presented spatially using 3D audio.

MPEG-4 Natural Audio Coding

The tools defined by MPEG-4 natural audio coding can be combined to implement different audio-coding algorithms. A set of different algorithms has been defined to establish optimum coding efficiency for the broad range of anticipated applications [5.108]. Figure 5.65 shows assignment of codecs to bit-rate ranges. The following lists the major algorithms of MPEG-4 natural audio [5.109]:

- General audio coding for medium and high qualities
- Twin VQ additional coding tools to increase the coding efficiency at very low bit rates
- HVXC low-rate clean speech coder
- CELP telephone speech/wideband speech coder

In addition to the coding tools used for the basic coding functionality, MPEG-4 provides techniques for additional features like bit-rate stream scalability.

General Audio Coding (Advanced Audio Coding Based)

This key component of MPEG-4 Audio covers the bit rate range of 16 Kb/s per channel up to bit rates higher than 64 Kb/s per channel. Figure 5.66 shows the arrangement of the building blocks of an MPEG-4 GA encoder in the processing chain. The same building blocks are present in a decoder implementation, performing the inverse processing steps.

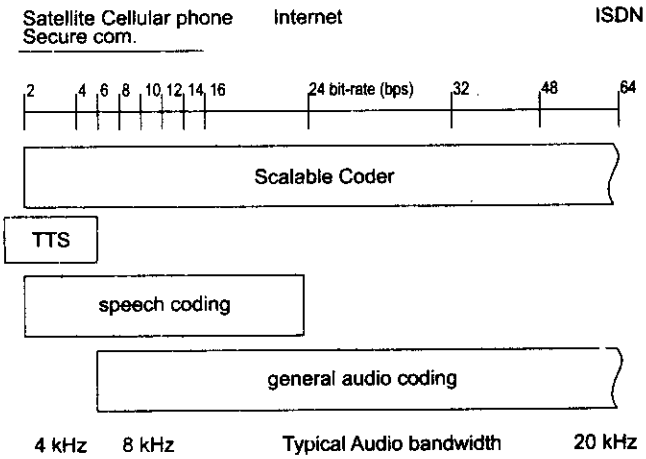


Figure 5.65 Assignment of codecs to bit-rate ranges.
©2000 ISO/IEC.

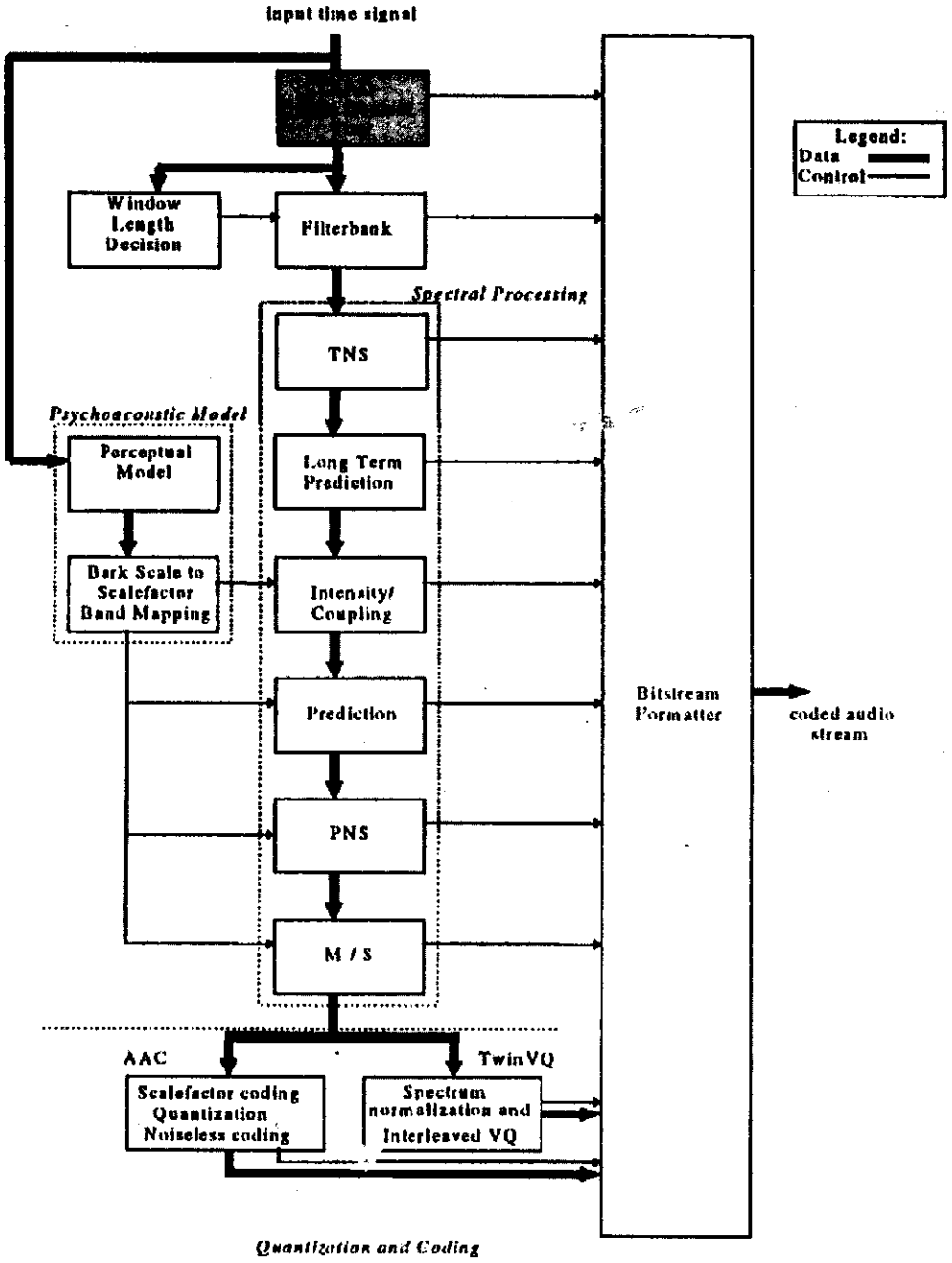


Figure 5.66 Building blocks of the MPEG-4 GA coder. ©2000 ISO/IEC.

The filter bank in MPEG-4 GA is derived from MPEG-2 AAC, that is, it is an MDCT supporting block lengths of 2,048 points and 256 points, which can be switched dynamically. Compared to previously known transform-coding schemes, the length of the long block transform is rather high, offering improved coding efficiency for stationary signals. MPEG-4 GA supports an additional mode with a block length of 1,920/240 points to facilitate scalability with the speech-coding algorithms in MPEG-4 Audio. All blocks are overlapped by 50% with the preceding and the following blocks. For improved frequency selectivity, the incoming audio samples are windowed before applying the transform. MPEG-4 AAC supports two different window shapes that can be switched dynamically. The two different window shapes are a sine-shaped window and a Kaiser-Bessel-Derived (KBD) window offering improved far-off rejection compared to the sine-shaped window. An important feature of the time-to-frequency transform is the signal adaptive selection of the transform length. This is controlled by analyzing the short time variance of the incoming time signal. To ensure block synchronization between two audio channels with different block-length sequences, eight short transforms are performed in a row using 50% overlap each and using specially designed transition windows at the beginning and the end of a short sequence. This keeps the spacing between consecutive blocks at a constant level of 2,048 input samples. For further processing, the spectral data in the quantization and coding parts of the spectrum are arranged in the so-called scale factor bands roughly reflecting the bark scale of the human auditory system. The frequency-domain prediction improves redundancy reduction of stationary signal elements. Because stationary signals can nearly always be found in long transform blocks, it is not supported in short blocks. The actual implementation of the prediction is a second-order backward adaptive lattice structure, independently calculated for every frequency line. The use of the predicted values instead of the original ones can be controlled on a scale factor bound-basis and is decided based on the achieved prediction gain in that band. To improve stability of the predictors, a cyclic-reset mechanism is applied that is synchronized between encoder and decoder using a dedicated bit-stream element. The required processing power of the frequency-domain prediction and the sensitivity to numerical imperfections make this tool hard to use on fixed-point platforms. Additionally, the backward adaptive structure of the predictor makes such bit streams quite sensitive to transmission errors.

Long-Term Prediction (LTP), newly introduced in MPEG-4, is an efficient tool for reducing the redundancy of a signal between successive coding frames. This tool is especially effective for the parts of the signal that have a clear pitch property. Because the long-term predictor is a forward adaptive predictor (prediction coefficients are sent as side information), it is inherently less sensitive to round off numerical errors in the decoder or bit errors in the transmitted spectral coefficients. The adaptive quantization of the spectral values is the main source of the bit-rate reduction in all transform coders. It assigns a bit allocation to the spectral values according to the accuracy demands determined by the perceptual model, realizing the irrelevancy reduction. The key components of the quantization process are quantization function and the noise shaping that is achieved through the scale factors. The quantizer used in MPEG-4 GA has been designed

similar to the one used in MPEG 1/2 Layer 3. It is a nonuniform quantizer. The main advantage over a conventional uniform quantizer is the implicit noise shaping that this quantization creates. The absolute quantizer step size is determined by a specific bit-stream element. It can be adjusted in 1.5 dB steps. To improve the subjective quality of the coded signal, the noise is further shaped by scale factors. They are used to amplify the signal in certain spectral regions (the scale factor bands) to increase the SNR in these bands. Thus the spectral values usually need more bits to be coded afterward. Like the global quantizer, the step size of the scale factors is 1.5 dB. To reconstruct the original spectral values in the decoder properly, the scale factors have to be transmitted within the bit stream. MPEG-4 GA uses an advanced technique to code the scale factors as efficiently as possible. It exploits the fact that scale factors usually do not change too much from one scale-factor band to another. Thus, differential encoding provides some advantage. It also uses a Huffman code to reduce the redundancy further within the scale-factor data.

The noiseless coding kernel within an MPEG-4 GA encoder tries to optimize the redundancy reduction within the spectral data coding. The spectral data is encoded using a Huffman code that is selected from a set of available codebooks according to the maximum quantized value. The set of available codebooks includes one to signal that all spectral coefficients in the respective scale factor band are 0, implying that neither spectral coefficients nor a scale factor are transmitted for that band. To find the optimum trade-off between selecting the optimum table for each scale factor band and minimizing the number of data elements to be transmitted, an efficient grouping algorithm is applied to the spectral data.

The basic idea of Temporal Noise Shaping (TNS) relies on the duality of time and frequency domains. TNS uses a prediction approach in the frequency domain to shape the quantization noise over time. It applies a filter to the original spectrum and quantizes this filtered signal. Additionally, quantized filter coefficients are transmitted in the bit stream. These are used in the decoder to undo the filtering performed in the encoder, leading to a temporally shaped distribution of quantization noise in the decoded audio signal. TNS can be viewed as a postprocessing step of the transform, creating a continuous signal adaptive filterbank instead of the conventional two-step switched-filterbank approach. The actual implementation of the TNS approach within MPEG-4 GA allows for up to three distinct filters applied to different spectral regions of the input signal, further improving the flexibility. A feature newly introduced into MPEG-4 GA is the Perceptual Noise Substitution (PNS) [5.110]. The technique of PNS is based on the observation that one noise sounds like the other. This means that the actual fine structure of a noise signal is of minor importance for the subjective perception of such a signal. Instead of transmitting the actual spectral components of a noisy signal, the bit stream would just signal that this frequency region is noiselike and gives some additional information on the total power in that band. PNS can be switched on a scale-factor band basis. Even if there are just some spectral regions with a noisy structure, PNS can be used to save bits. In the decoder, a randomly generated noise will be inserted into the appropriate spectral region according to the power level signaled with the bit stream.

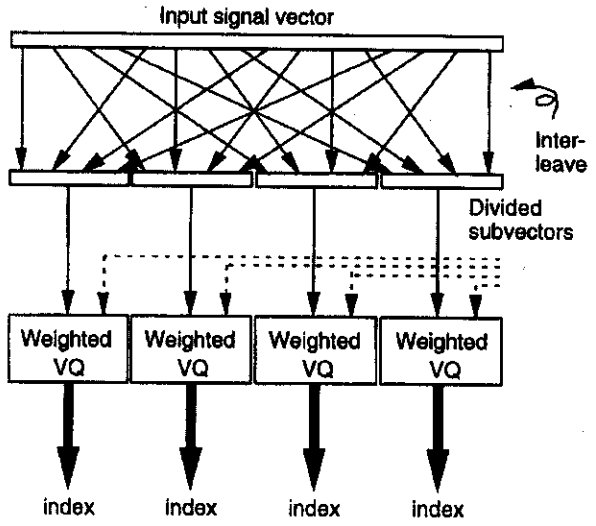


Figure 5.67 Twin VQ [5.40]. ©1998 ISO/IEC.

Twin VQ

To increase coding efficiency for coding of musical signals at very low bit rates, twin VQ-based coding tools are part of the GA coding systems in MPEG-4 Audio. The basic idea is to replace the conventional encoding of scale factors and spectral data used in MPEG-4 AAC by an interleaved VQ applied to a normalized spectrum [5.111, 5.112]. The basic idea of the weighted interleaved vector quantization (Twin VQ) scheme is represented in Figure 5.67. The input signal vector (spectral coefficients) is interleaved into subvectors. These subvectors are quantized using vector quantizers. Twin VQ can achieve a higher coding efficiency at the cost of always creating a minimum amount of loss in terms of audio coding.

Speech Coding in MPEG-4 Audio

Most of the recent speech-coding algorithms can be categorized as spectrum coding or hybrid coding. Spectrum coding models the input speech signal based on a vocal tract model, which consists of a signal source and a filter as shown in Figure 5.68. A set of parameters obtained by analyzing the input signal is transmitted to the receiver. Hybrid coding synthesizes an approximate speech signal based on a vocal tract model. A set of parameters used for this synthesis is modified to minimize the error between the original and the synthesized speech signals. A best parameter set can be searched for by repeating this analysis-by-synthesis procedure. The obtained set of parameters is transmitted to the receiver as the compressed data after quantization. In the decoder, a set of parameters for source and LP synthesis filtering is recovered by inverse quantization. These parameter values are used to operate the same vocal tract model as in the encoder. Figure 5.69 gives a block diagram of hybrid speech coding. Source and LP synthesis filters correspond to those in Figure 5.68. The error between the input signal and the synthesized signal is weighted by a Perceptual Weighted (PW) filter. The filter has a frequency

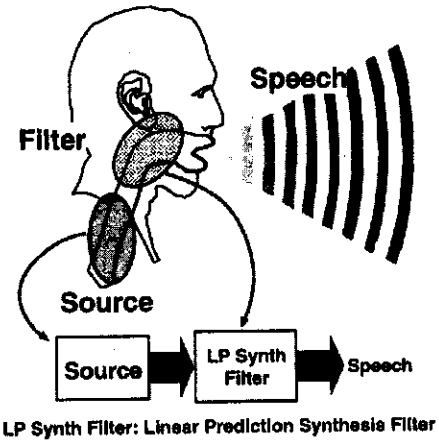


Figure 5.68 Vocal tract model. ©2000 ISO/IEC.

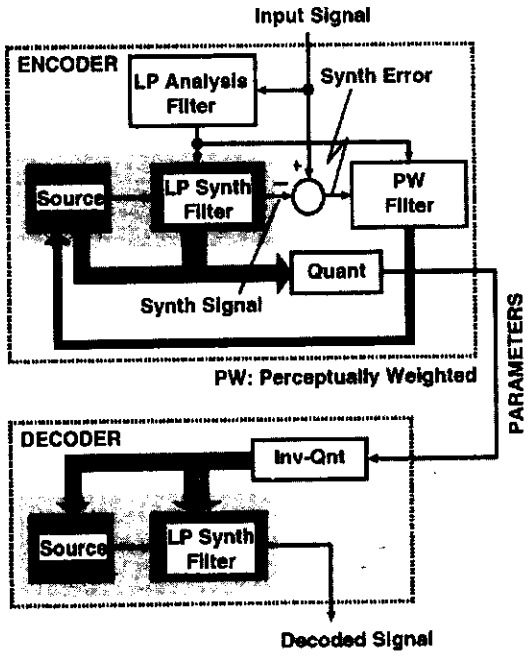


Figure 5.69 Hybrid speech coding. ©2000 ISO/IEC.

response that takes the human auditory system into consideration. Thus, a perceptually best parameter selection can be achieved.

The MPEG-4 natural speech-coding toolset provides a generic coding framework for a wide range of applications with speech signals. Two different bandwidths are covered [5.58]: 4 KHz and 7 KHz. The MPEG-4 natural speech-coding toolset contains two algorithms: HVXC

Table 5.14 Specifications of MPEG-4 natural speech-coding tools [5.113].

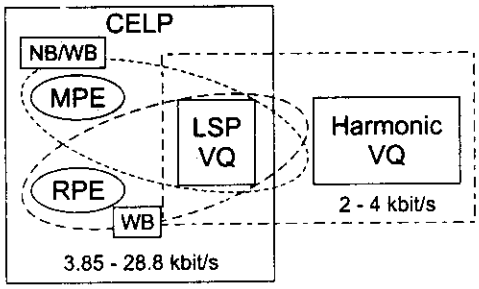
HVXC	Parameter	
Sampling frequency	8 KHz	
Bandwidth	300-3400 Hz	
Bit rate [bit/s]	2,000 and 4,000	
Frame size	20 ms	
Delay	33.5-56 ms	
Features	Multi-bit-rate coding Bit-rate scalability	
CELP	**Parameter**	**Parameter**
Sampling frequency	8 KHz	16 KHz
Bandwidth	300-3,400 Hz	50-7,000 Hz
Bit rate [bit/s]	3,850-12,200 (28 bit rates)	10,900-23,800 (30 bit rates)
Frame size	10-40 ms	10-20 ms
Delay	15-45 ms	15-26.75 ms
Features	Multi-bit-rate coding Bit-rate scalability Bandwidth scalability	

©1998 ISO/IEC.

and CELP. The specifications of the MPEG-4 natural speech-coding toolset are summarized in Table 5.14, and Figure 5.70 represents the corresponding toolset.

MPEG-4 is based on tools that can be combined according to the user needs. HVXC consists of the Line Spectral Pair (LSP), VQ and harmonic VQ tool. The Regular Pulse Excitation (RPE) tool, Multipulse Excitation (MPE) tool and LSP VQ tool form CELP. The RPE tool is allowed only for the wideband mode because of its simplicity at the expense of the quality. The LSP VQ tool is common both in HVXC and CELP.

HVXC. A basic block diagram of HVXC is presented in Figure 5.71. LP analysis to find the LP coefficients is first performed. Quantized LP coefficients are supplied to the inverse LP filter to find the prediction error. The prediction error is transformed into a frequency domain,



NB: Narrow Band RPE: Regular Pulso Excitation
WB: Wide Band MPE: Multipulse Excitation
VQ: Vector Quantization

Figure 5.70 MPEG-4 natural-speech coding toolset. ©2000 ISO/IEC.

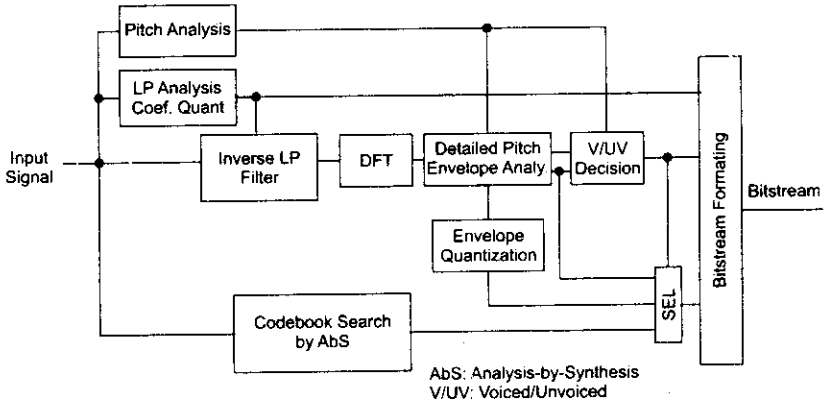


Figure 5.71 HVXC block diagram. ©2000 ISO/IEC.

and the pitch and the envelope of the spectrum are analyzed. The envelope is quantized by weighted VQ in voiced sections. In unvoiced sections, a closed-loop search of an excitation vector is arrived at.

CELP. Figure 5.72 shows a block diagram of CELP. The LP coefficients of the input signal are first analyzed and then quantized to be used in an LP synthesis filter driven by the output of the excitation codebooks. Encoding is performed in two steps. LTP coefficients are calculated in the first step. In the second step, a perceptually weighted error between the input signal and the LP synthesis filter is minimized. This minimization is achieved by searching for an appropriate code vector for the excitation codebooks. Quantized coefficients, as well as indexes to the code vectors of the excitation codebooks and the LTP coefficients, form the bit stream. The LP coefficients are quantized by vector quantization and the excitation can be either MPE or regular pulse excitation RPE [5.114]. MPE and RPE both model the excitation signal by multiple pulses. However, a distance exists in the degrees of freedom for pulse positions. MPE allows more freedom of the interpulse distance than RPE, which has a fixed-interpulse distance. Thanks to such a flexible interpulse distance, MPE achieves better coding quality than RPE [5.113]. On

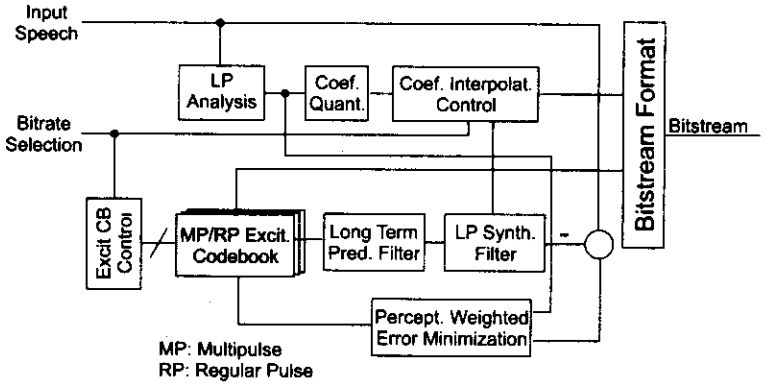


Figure 5.72 Block diagram of coding CELP. ©2000 ISO/IEC.

Table 5.15 The excitation signal types of MPEG-4/CELP.

Excitation	Bandwidth	Features
MPE	Narrow, wide	Quality, scalability
RPE	Wide	Complexity

the other hand, RPE requires fewer computations than MPE by trading off its coding quality. Such a low computational requirement is useful in the wideband coding where the total computation should naturally be higher than in the narrowband coding. The excitation signal types of MPEG-4/CELP are summarized in Table 5.15.

Scalability in MPEG-4 Natural Audio

Bit-stream scalability is the ability of an audio codec to support an ordered set of bit streams that can produce a reconstructed sequence. Moreover, the codec can output useful audio when certain subsets of the bit stream are decoded. The minimum subset that can be decoded is called the base layer. The remaining bit streams in the set are called enhancement or extension layers. Depending on the size of the extension layers, there exists large-step or small-step (granularity) scalability. Small-step scalability denotes enhancement layers of around 1 Kb/s or smaller. Typical data rates for the extension layers in a large-step scalable system are 16 Kb/s or more. A scalability in MPEG-4 Natural Audio largely relies on differences, either in time domain, or, as in the case of AAC layers of the spectral lines, it is in frequency domain.

Synthetic Audio in MPEG-4

Natural and synthetic audio are not unrelated methods for transmitting sound. As sound models in perceptual coding grow more sophisticated, the boundary between decompression and synthesis becomes somewhat blurred [5.115]. In [5.106, 5.116], the relationships among parametric models of sound, digital sound creation and transmission, perceptual coding, parametric compression and various techniques for algorithmic synthesis have been discussed.

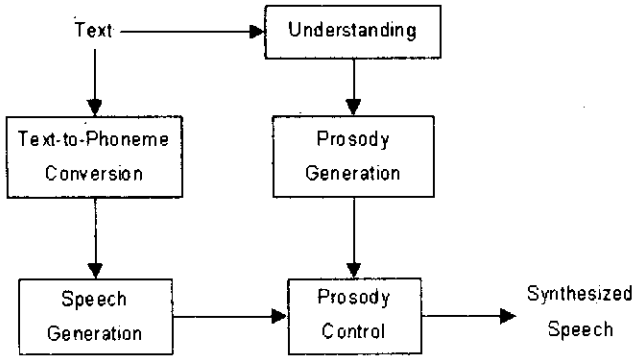


Figure 5.73 The interaction between text-to-phoneme conversion text understanding and prosody generation and application. ©2000 ISO/IEC.

TTS systems generate speech sound according to given text. This technology enables the translation of text information into speech so that the text can be transferred through speech channels such as telephone lines. TTS systems consist of the multiple processing modules shown in Figure 5.73 [5.117]. Such a system accepts text as input and generates a corresponding phonemes sequence. Phonemes are the smallest units of human language. Each phoneme corresponds to one sound used in speech. About 120 phonemes are sufficient to describe all human languages. The phoneme sequence is used to generate a basic speech sequence without prosody, that is, without pitch, duration and amplitude variations. In parallel, a text-understanding module analyzes the input for phrase structure and inflections. Using the result of this processing, a prosody-generation module creates the proper prosody for the text [5.118]. Finally, a prosody-control module changes the prosody parameters of the basic speech sequence according to the results of the text-understanding module, yielding synthesized speech. Today, TTS systems are used for many applications, including automatic voice-response systems (the "telephone men" systems), email reading and information services for the visually handicapped [5.118]. The applications of TTS are expanding in telecommunications, personal computing and the Internet. Current research in TTS includes voice conversion (synthesizing the sound of a particular speaker's voice), multilanguage TTS and enhancement of the naturalness of speech through modern sophisticated voice models and prosody generators.

Text, that is, a sequence of words written in some human language, is a widely used representation for speech data in standalone applications. However, it is difficult with existing technology to use text as a speech representation in a multimedia bit stream for transmission. The MPEG-4 TTSI is defined so that speech can be transmitted as a bit stream containing text. It also provides interoperability among TTS synthesizers by standardizing a single bit stream format for this purpose.

Synthetic speech is becoming a rather common media type. It plays an important role in various multimedia application areas. For instance, by using TTS functionality, multimedia content with narration can be easily created without recording natural speech. In MPEG-4, a single common interface for TTS systems is standardized. This interface allows speech information to be transmitted in the International Phonetic Alphabet (IPA) or in a textural (written) form of any language. The MPEG-4 TTSI is a hybrid/multilevel scalable TTSI that can be considered a

superset of the conventional TTS framework. This extended TTSI can use prosodic information taken from natural speech in addition to input text and can thus generate much higher quality synthetic speech. As well as an interface to TTS synthesis systems, MPEG-4 specifies a joint coding method for phonemic information and FAPs. The MPEG-4 TTSI has important functionalities both as an individual code and in synchronization with the facial animation techniques described in Tekalp and Ostermane [5.77]. The basic TTSI format is extremely low bit rate. The synthesized speech with predefined prosody will deliver emotional content to the listener. One of the important features of the MPEG-4 TTSI is the ability to synchronize synthetic speech with the lip movements of a computer generated talking head. In this technique, the TTS synthesizer generates phoneme sequences and their duration and communicates them to the facial animation visual object decoder so that it can control the lip movement. Through the MPEG-4 elementary synchronization capabilities, the MPEG-4 TTSI can perform synthetic motion picture dubbing [5.52]. The MPEG-4 TTSI decoder can use the system clock to select an adequate speech location in a sentence and communicates this to the TTS synthesizer, which assigns appropriate duration for each phoneme. Using this method, synthetic speech can be synchronized with the lip shape of the moving image. An overview of the MPEG-4 TTSI decoding process showing the interaction between the syntax parser, the TTS synthesizer, and the face animation decoder is given in Figure 5.74. The TTS synthesizer and face-decoder blocks are not normatively described and operate in a terminal-dependent manner [5.117].

The architecture of the decoder can be described as a collection of interfaces. Upon receiving a multiplexed MPEG-4 bit stream, the demux passes coded MPEG-4 TTSI ESs to the syntactic decoder. Other ESs are passed to other decoders. Receiving a coded MPEG-4 TTSI bit stream, the synthetic decoder passes a number of different pieces of data to the TTS synthesizer. The input type specifies whether TTS is being used as a standalone function or in the synthetizer with facial animation or motion-picture dubbing. The control commands sequence specifies the language, gender, age and speech rate of the speaking voice. The input text specifies the character string for the text to be synthesized. Auxiliary information, such as IPA phoneme symbols

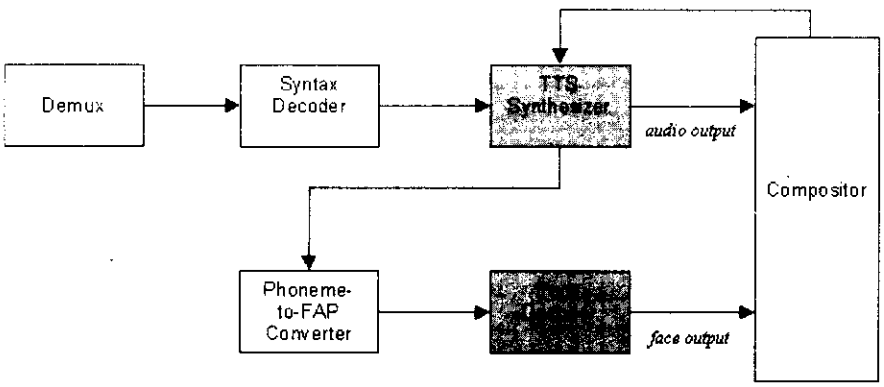


Figure 5.74 Block diagram of the MPEG-4 TTSI decoder. ©2000 ISO/IEC.

(which allow text in a language foreign to the decoder to be synthesized), lip shape patterns and trick-mode commands, is also passed along the interface between the synthetic decoder and the TTS synthesizer. The synthesizer constructs a speech sound and delivers it to the audio composition system. The interface from the compositor to the TTS synthesizer allows the local control of the synthesized speech by users. Using this interface and an appropriate interactive scene, users can start, stop, rewind and fast-forward the TTS system. Controls can also allow changes to the speech rate, pitch range, gender, and age of the synthesized speech by the user. The TTS synthesizer and the face animation can be driven synchronously by the same input control stream, which is the text input to the MPEG-4 TTSI. From this input stream, the TTS synthesizer generates synthetic speech, and, at the same time, phoneme symbols, phoneme durations and word boundaries generate the phoneme/bookmark-to-FAP converter, which generates relevant facial animation. In that way, the synthesized speech and facial animation are synchronized when they enter the scene composition framework.

The tool that provides audio synthesis capability in MPEG-4 is termed the structured audio coder [5.106, 5.119]. MPEG-4 structured audio is a codec like the other audio tools in MPEG-4. A sound transmission is decomposed into two pieces: a set of synthesis algorithms that describe how to create sound and a sequence of synthesis controls that specifies which sounds to create. The synthesis model is not fixed in the MPEG-4 terminal. The standard specifies a framework for reconfigurable software synthesis. Like the other MPEG-4 media types, a structured audio bit stream consists of a decoder configuration header that tells the decoder how to begin the decoding process and then a stream of bit-stream access units that contain the compressed data. In structured audio, the decoder configuration header contains the synthesis algorithms and auxiliary data, and the bit-stream access units contain the synthesis control instructions.

Audio BIFS

The part of BIFS controlling the composition of a sound scene is called audio BIFS. It provides a unified framework for sound scenes that use streaming audio, interactive presentation, 3D spatialization and dynamic download of custom signal-processing effects [5.55, 5.120].

Audio BIFS contains significant advances in quality and flexibility over VRML audio. There are two main modes of operation that audio BIFS is intended to support: virtual-reality and abstract-effects compositing. In virtual-reality compositing, the goal is to re-create a particular acoustic environment as accurately as possible. Sound should be presented spatially according to its location relative to the listener in a realistic manner. Moving sounds should have a Doppler shift. Distant sounds should be attenuated and low-pass filtered to simulate the absorptive properties of air. Sound sources should radiate sound unevenly, with a specific frequency-dependent directivity pattern. This type of scene composition is most suitable for virtual world applications and video games where the application goal is to immerse the user in a synthetic environment. In abstract-effects compositing, the goal is to provide content authors with a rich suite of tools from which artistic considerations can be used to choose the right effect for a given situation.

A schematic diagram for the overall audio system in MPEG-4 is shown in Figure 5.75. Sound is conveyed in the MPEG-4 bit stream as several ESs that contain coded audio. There are four ESs in the sound scene. Each of these ESs contains a primitive media object, which, in the case of audio, is a single-channel or multichannel sound that will be composited into the overall scene. The MPEG-4 audio system shows the interaction between decoding, scene description and audiovisual synchronization. The conceptual flow is from the bottom of the figure to the top. At the bottom, two multiplexed MPEG-4 bit streams, each from a different server, convey several ESs containing compressed data. Each bit stream is demultiplexed. A total of four ESs is produced. The ESs are decoded using various MPEG-4 decoders into four primitive media objects containing uncompressed PCM scene graphs and are presented to the listener as though they emanate from the sound nodes. The BIFS part and the audio BIFS part of the scene graph are separated, but there is no technical difference between them. Namely, audio BIFS is just a subset of BIFS. Audio BIFS consists of a number of nodes that are interlinked in a scene graph. An audio BIFS scene graph is termed an audio subgraph. The audio BIFS nodes are summarized in Table 5.16, which describes their function in an audio scene. Each node has several fields that specify the parameters of operation of the node. In MPEG-4 BIFS, these fields and their operation are carefully quantized and transmitted in a binary data format for maximum compression of the scene graph.

5.5.6 Profiles and Levels in MPEG-4

Profiles and levels in MPEG-4 serve two main purposes:

- Ensuring interoperability between MPEG-4 implementations
- Allowing conformance to the standard to be tested

Profiles exist not only for the audio and visual parts of the standard (audio profiles and visual profiles), but also for the systems part of the standard, in the form of graphics profiles, scene graph profiles and an object descriptor profile. Different profiles are created for different application environments. The policy for defining profiles is that they should enable as many applications as possible while keeping the number of different profiles low [5.7].

Media profiles describe the object types that can be used to create the scene and tools that can be used to create those object types.

Visual Object Types

Five different object types represent video information:

- The Simple object type is an error-resilient rectangular natural VO of arbitrary height/width ratio, developed for low bit rates. It uses relatively simple and inexpensive coding tools, based on I-VOPs and P-VOPs.
- The Simple Scalable object type is a scalable extension of Simple, which gives temporal and spatial scalability using Simple as the base layer. The enhancement layer is still rectangular.

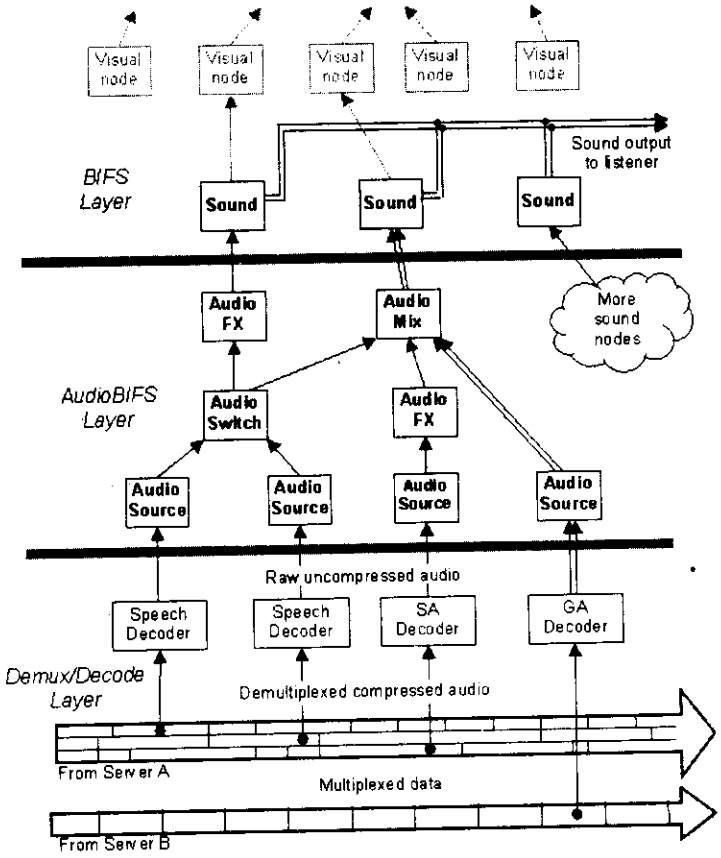


Figure 5.75 The MPEG-4 audio system.

Table 5.16 The audio BIFS nodes.

Name	Function
AudioSource	Attach sound decoder to scene graph
AudioMix	Mix M channels of sound into N channels
AudioSwitch	Select subset of M input channels of sound
AudioDelay	Delay sounds for synchronization
AudioFX	Apply algorithmic signal-processing effects

© 1998 ISO/IEC.

Table 5.16 The audio BIFS nodes. (Continued)

Name	Function
AudioBuffer	Cache sound for use in interactive playback
Sound	Position sound in 3D virtual environment
Sound2D	Position sound in 2D scene
Group	Group multiple nodes together for hierarchical transformation
ListeningPoint	Specify location of virtual listener in scene
TermCap	Query terminal for available playback resources

© 1998 ISO/IEC.

- The Core object type uses a tool superset of Simple, giving better quality through the use of bidirectional interpolation, and it has binary shape. It supports scalability based on sending extra P-VOPs.
- The Main object type is the VO that gives the highest quality. Compared to Core, it also supports gray-scale shape, sprites and interlaced content in addition to progressive material.
- The N-bit object type is equal to the Core object type, but it can vary the pixel depth from 4 to 12 bits for the luminance as well as the chrominance planes.

One special object type represents still natural visual information. This is the Still Scalable Texture object type. It gives an arbitrary shape still image that uses wavelet coding for scalability and incremental download and build up.

The following object types use synthetic tools, some of which are in combination with natural video texture:

- The Animated 2D Mesh object type combines the synthetic mesh with natural video. The natural video coding uses the same tools as the Core object type. This video can be mapped onto the mesh and deformed by moving the points in the mesh.
- The Basic Animated Texture object type allows mesh animation with arbitrary shape still images.
- The Simple Face object type has the tools for facial animation. This object type does not define what the face looks like. The animation can be applied to any local model of choice.

Visual Profiles

The visual profiles determine which visual object types can be present in the scene:

- The Simple Profile accepts only objects of type Simple and was created with low-complexity applications in mind. The first use is mobile use of (audio) visual services, and the second is putting very low-complexity video on the Internet. Also, small camera devices recording moving video to, for example, disk or memory chips, can make good use of this profile. The Simple Profile has three levels with bit rates from 64 to 384 Kb/s. The levels also define the maximum total surface for the objects and the amount of macroblocks per second that the decoder needs to be able to decode. Further, they define the size of various (hypothetical) buffers needed for decoding.
- The Simple Scalable Profile can supply scalable coding in the same operational environments as foreseen for Simple and has two levels defined.
- The Core Profile accepts Core and Simple objects types. It is useful for higher quality interactive services, combining good quality with limited complexity and supporting arbitrary shape objects. Also, mobile broadcast services should be supported by this profile. The maximum bit rate is 384 Kb/s in Level 1 and 2 Mb/s in Level 2. The number of macroblocks is chosen such that a scene using this typical session size can have overlapping objects and still be filled.
- The Main Profile was created with broadcast services in mind, addressing progressive as well as interlaced material. It combines the highest quality with the versatility of arbitrarily shaped objects using gray-scale coding. The highest level accepts up to 32 objects (of Simple, Core or Main type) for a maximum total bit rate of 38 Mb/s.
- The N-bit profile is useful for areas that use terminal imagers, such as surveillance applications. Medical applications may also want to use enhanced pixel depth giving a larger dynamic range in color and luminance. It accepts objects of types Simple, Core and N-bit profiles.
- The Simple Face Profile accepts only objects of type Simple Face. Depending on the level, either one or a maximum of four faces can appear in the scene, for example, for a virtual meeting. Bit rates remain very low. Even for the second level, 32 Kb/s is more than adequate for driving the four faces.
- The Hybrid Profile allows combining both natural and synthetic objects in the same scene while keeping complexity reasonable. On the natural side, it compares to the Core Profile, and on the synthetic side, it adds animated meshes, scalable textures and animated faces. This is a rich set of tools for creating attractive hybrid natural and synthetic content. This profile can be used to place real objects into a synthetic world and to add synthetic objects to a natural environment.
- The Basic Animated Texture Profile allows animation of still pictures and facial animation. Attractive content can be created at very low bit rates.

Audio Object Types

For coding natural sound, MPEG-4 includes AAC and twin VQ algorithms. The following object types exist:

- The AAC Main object type has multichannel capability to give five full channels plus a separate low-frequency channel in one object. It is very similar and compatible with the AAC Main profile that is defined in MPEG-2 (ISO/IEC 13818-7). MPEG-4 AAC adds the perceptual noise-shaping tool.
- The MPEG-4 AAC Low Complexity object type is a low complexity version of the AAC Main Object type.
- The MPEG-4 AAC Scalable Sampling Rate object type is the counterpart to the MPEG-2 AAC Scalable Sampling Rate profile.
- The MPEG-4 AAC LTP object type is similar to the AAC Main object type with the LTP replacing the MPEG-2 AAC predictor. This gives the same efficiency with significantly lower implementation costs.
- The AAC Scalable object type allows a large number of scalable combinations, including combinations with twin VQ and CELP coder tools as the core coders. It supports only mono or two-channel stereo sound.
- The Twin VQ object type is based on fixed-rate VQ instead of the Huffman coding used in AAC. It operates at lower bit rates than AAC and supports mono and stereo sound.
- The CELP object type uses CELP. It supports 8 KHz and 16 KHz sampling rates at bit rates from 4 to 24 Kb/s. CELP bit streams can be coded in a scalable way using bit-rate scalability and bandwidth scalability.
- The HVXC object type gives a parametric representation of 8 KHz mono speech at fixed bit rates between 2 and 4 Kb/s and below 2 Kb/s using a variable bit-rate mode, and supports pitch and speed changes.
- The TTSI object type gives an extremely low bit-rate phonemic representation of speech. Bit rates range from 0.2 to 1.2 Kb/s. The synthesized speech can be synchronized with a facial animation object.

A number of different object types exists for synthetic sound:

- The Main Synthetic object type collects all MPEG-4 structured audio tools. Structured audio is a way to describe methods of synthesis [5.117]. Sound can be described at 0 Kb/s to 3 to 4 Kb/s for extremely expressive sounds in MPEG-4 Structured Audio format.
- The Wavetable Synthesis object type is a subset of the Main Synthetic object type. It provides relatively simple sampling synthesis.
- The General Musical Instrument Digital Interface (MIDI) object type gives interoperability with existing content. Unlike the Main Synthetic or Wavetable

Synthesis object types, it does not give completely predictable sound quality and decoder behavior.

- The Null object type provides the possibility to feed raw PCM data delivery to the MPEG-4 audio compositor in order to allow mixing in of local sound at the decoder. The support for this object type is in the compositor.

Audio Profiles

There are only four different audio profiles. The application area for the Speech profile can be deduced from its name. Two levels are defined, determining whether either 1 or a maximum 20 objects can be present in the audio scene. A prime reason for defining the Scalable profile was to allow good quality, reasonable complexity, low-bit-rate audio on the Internet and an environment in which bit rate varies from user to user and from one minute to the next. Scalability allows making optimal use of available, and even dynamically changing, bandwidth while having to encode and store the material only once. The Scalable profile has four levels that restrict the amount of objects in the scene, the total amount of channels and the sampling frequency. The highest level employs the novel concept of complexity units. The Synthetic profile groups all the synthetic object types. The main application areas are found where good quality sounds are needed at very low data rates while the sound source usually uses microphones. Three levels define the amount of memory for the data, the sampling rates, amount of TTSI objects and some further processing restrictions. The Main profile includes all object types. It is useful in environments where processing power is available to create very rich, highest quality audio scenes that may combine microphone-recorded sources with synthetic ones. Example applications are the DVD and multimedia broadcast. This profile has four levels, defined in terms of complexity units. There are two different types of complexity units: Processor Complexity Units (PCU) specified in millions of operations/s, and RAM Complexity Units (RCU) specified in terms of number of kwords.

Graphics

Graphics profiles regulate which of the graphics and textual elements can be used to build a scene. They are expressed in terms of BIFS nodes. Three hierarchical graphics profiles are defined in MPEG-4: Simple 2D, Complete 2D and Complete. Simple 2D provides the basic functionalities needed to create a visual scene with visual objects without giving additional graphical elements. Complete 2D allows elements like bitmaps, backgrounds, circles, boxes and lines, all in a flat space. These elements have characteristics like line width and color. Complete contains the full set of BIFS graphics nodes with which complete and elaborate 3D graphics can be created. It adds to Complete 2D, for example, the sphere, the cone, 3D boxes, directional lighting, and so forth.

Systems Profiles

Here, we present the Scene Graph profiles and Object Descriptor profiles. The Scene Graph profile defines what types of transformational capabilities need to be supported to the terminal. This

is defined in terms of the scene graph elements (BIFS nodes) that the decoding terminal needs to be able to understand. Examples are translations, 3D rotations and elements like input sensors with which interactive behavior can be created. The Scene Graph profiles follow a structure similar to the graphics profiles, with one profile added for audio-only scenes. Thus, we have Audio, Simple 2D, Complete 2D, and Complete. The target applications are very much the same as those for the Graphics profiles with Simple 2D providing placement capabilities, Complete 2D adding for instance rotation and Complete giving the capability to do arbitrary transformations in a 3D space.

The Object Descriptor profile specifies allowed configurations of the object descriptor and Sync Layer tools [5.121]. The object descriptor contains all descriptive information, and the Sync Layer tool provides the syntax to convey, among others, tuning information for ESs. The main reason for wanting to subject the Object Descriptor to profiling lies in reducing the amount of asynchronous operations and the necessary permanent storage.

5.6 MPEG-4 Visual Texture Coding (VTC) and JPEG 2000 Image Compression Standards

With the increasing use of multimedia communication systems, image compression requires higher performance and new features. JPEG 2000 is an emerging standard for still-image compression. It is not only intended to provide rate distortion and subject image quality performance superior to existing standards, but also to provide functionality that current standards can either not address efficiency or not address at all [5.122]. The compression advantages of JPEG 2000 are a direct result of the inclusion into the standard of a number of advanced and attractive features, including progressive recovery, lossy/lossless compression and region of interest capabilities. These features lay the foundation for JPEG 2000 to provide tremendous benefits to a range of industries. Some of the applications that will benefit directly from JPEG 2000 are image archiving, Internet, Web browsing, document imaging, digital photography, medical imaging and remote sensing.

Functionally, JPEG 2000 includes many advanced features:

- Component precision of 1 to 127 bits/sample (signed or unsigned)
- Components that may each have a different precision and subsampling factor
- Use of image data that may be stored compressed or uncompressed
- Lossy and lossless compression
- Progressive recovery by fidelity or resolution
- Tiling
- Error resilience
- Region-of-interest coding
- Random access to an image in a spatial domain
- Security

Image compression must not only reduce the necessary storage and bandwidth requirements, but also must allow extraction for editing, processing and targeting particular devices and applications. JPEG 2000 allows extraction of different resolutions, pixel fidelities, regions of interest, components and more, all from a single compressed bit stream. This allows an application to manipulate or transmit only the essential information for any target device from any JPEG 2000 compressed source image.

Some of the technology highlights for JPEG 2000 are the following:

- Wavelet subband coding
- Reversible integer-to-integer and nonreversible real-to-real wavelet transforms
- Reversible integer-to-integer and nonreversible real-to-real multicomponent transforms
- Bit-plane coding
- Arithmetic coding
- Use of Embedded Block Coding with Optimized Truncation (EBCOT) coding scheme
- Code-stream syntax similar to JPEG
- File format syntax

In what follows, the structure of the JPEG 2000 standard is presented together with performance and the complexity comparisons with existing standards.

5.6.1 JPEG 2000 Development Process

The JPEG 2000 project was motivated by submission of the Compression with Reversible Embedded Wavelet (CREW) algorithms to an earlier standardization effort for lossless and near-lossless compression (known as JPEG-LS) [5.123, 5.124].

With the continual expansion of multimedia and Internet applications, the needs and requirements of the technologies used grew and become quite complex. A call for technical contributions was issued in March 1997, requesting compression technologies be submitted to an evaluation during the November 1997 WG1 meeting in Sydney, Australia [5.125]. The Wavelet/Trellis Coded Quantization (WTCQ) algorithm ranked first overall in both the subjective and objective evaluations. It was further decided that a series of core experiments would be conducted to evaluate WTCQ and other technologies in terms of JPEG 2000 desired features and in terms of algorithm complexity. Results from the first round of core experiments were presented at the March 1998 WG1 meeting in Geneva. Based on these experiments, it was decided to create a JPEG 2000 VM, which would lead to a reference implementation of JPEG 2000. The VM was modified in each meeting based on experiments performed between meetings. Results from Round 1 core experiments were selected to modify WTCQ in the first release of the VM (VM0).

The basic ingredients of the WTCQ algorithm are the discrete wavelet transform, Trellis Coded Quantization (TCQ) using step sizes chosen with a Lagrangian rate-allocation procedure [5.126, 5.127] and binary arithmetic coding. The embedding principle asserts the encoded bitstream should be ordered in a way that maximally reduces Mean Square Error (MSE) per bit

transmitted [5.123, 5.128, 5.129, 5.130, 5.131]. In WTCQ, embedding is provided by the bit-plane coding. The bit-plane coding operates on TCQ indexes (trellis quantized wavelet coefficients) in a way that enables successive refinement. This is accomplished by sending bit planes in decreasing order from most to least significant. To exploit spatial correlation within bit planes, spatial context models are used. In general, the context can be chosen within a subband and across subbands. The WTCQ bit-plane coders avoid the use of intersubband contexts to maximize flexibility in scalable decoding and to facilitate parallel implementation. WTCQ also includes a binary mode, a classification of coefficients, multiple decompositions (dyadic, packet and others) and difference images to provide lossless compression [5.132].

Additions and modifications to VM0 continued for several meetings. VM2 supported user-specified floating point and integer transforms, as well as user-specified decompositions (dyadic, uniform and so forth). As a simpler alternative to the Lagrangian rate allocation, a fixed quantization table (Q-table) was included. This is analogous to the current JPEG standard [5.133, 5.134, 5.135]. When a Q-table is used, precise rate control can still be obtained by truncating the (embedded) bit stream. In addition to TCQ, scalar quantization was included in VM2. For integer wavelets, scalar quantization with the step size 1 was employed (no quantization), which allowed progression to lossless in the manner of CREW. Rate control for integer wavelets was accomplished by embedding, and a lossless compression scheme was available from the fully decoded embedded bit stream. Other features, such as tiling, region of interest coding and decoding, error resilience and approximate wavelet transforms with limited spatial support were added to the VM. Several refinements were made to the bit-plane coder. The major changes were the deinterleaving of bit planes and improvements to the context modeling. Within a given bit plane of each subband, the bits were deinterleaved into three subplanes of the following types: bits predicted to be newly significant, refinement bits and bits predicted to remain insignificant. The idea of subplanes was first reported in 1998 and was motivated by rate-distortion concerns [5.131]. Also, it is desirable to have the bits with the steepest rate-distortion slopes appear first in an embedded bit stream. The VM2 bit-plane coder has no intersubband dependencies such as those used in zero tree-based schemes [5.128, 5.130]. In VM2, all coding was carried out using context-dependent binary arithmetic coding [5.136]. It should be noted that, when encoding a particular bit, neither significance prediction nor context-modeling stages can use any information that would not be available at the decoder when that bit needs to be decoded. Thus, for wavelet coefficients that are noncausal with respect to the scan pattern, only information from more significant bit planes is used.

EBCOT included the idea of dividing each subband into rectangular blocks of coefficients and performing the bit-plane coding independently on these codeblocks. This partitioning reduces memory requirements in both hardware and software implementations, and it also provides a certain degree of (spatial) random access to the bit stream. EBCOT also included an efficient syntax for forming the sub-bit plane of multiple code blocks into packets. EBCOT was adopted for inclusion in VM3 [5.137].

During the March 1999 WG1 meeting in Korea, the MQ coder (submitted by Mitsubishi) was adopted as the arithmetic coder for JPEG 2000. This coder is functionally similar to the QM coder available as an option of the original JPEG standard. The MQ coder has some useful bit-stream creation properties, is used in the JBIG-2 standard, and should be available on a royalty and fee-free basis for ISO standards.

In fact, one goal of WG1 has been the creation of a Part 1, which could be used entirely on a royalty and fee-free basis. This is essential for the standard to gain wide acceptance as an interchange format. At the same time, as changes were being made to the internal coding algorithms, syntax wrapping of the compressed data was developed. This syntax is made up of a sequence of markers compatible with those of the original JPEG [5.135] and with features added to allow the identification of relevant portions of the compressed data. One Annex of the JPEG 2000 standard contains an optional minimal file format to include information such as the color space of the pixels and intellectual property (copyright) information for images. The inclusion of this Annex will prevent the proliferation of the property file format that happened with the original JPEG. This optional file format is extensible, and Part 2 defines storage of many additional types of metadata. The standardization process has already produced the WD and the CD documents [5.138, 5.139]. Final Draft International Standard (FDIS) was produced in August 2000, and finally JPEG 2000 was produced in December 2000.

Part 2 became an International Standard in October 2001. Division of the standard between Part 1 and Part 2 is shown in Table 5.17. It lists the various components of the compression system and the extensions likely for Part 2. For example, Part 1 will require one floating-point wavelet (9,7) and one integer wavelet (5,3), and Part 2 will allow multiple wavelets, including user-defined ones [5.140].

Part 3 of JPEG 2000 is known as MotionJPEG 2000. MotionJPEG has been a commonly used method of editing high-quality video without the existence of an ISO standard. This technology became an International Standard in November 2001 and should have important application in the next generation of digital cameras and elsewhere. In addition, MotionJPEG 2000 will allow support for both lossless and lossy compression in a single codec.

Part 4 of the JPEG 2000 Standard addresses conformance-testing issues, which is a key function to assure the interoperability of various implementations of the standard by the widest community of developers.

Part 5 of the JPEG 2000 standard contains reference software that implements the basic features of Part 1 and is due to be disseminated for both noncommercial and commercial users. A CD of Part 5 of the standard was also produced, together with two releases of reference code: one Image Power' JasPer codec written in the C language and the other in the Java programming language. Due to increased interest in JPEG 2000 and its growing importance in various applications and business models, two more parts were added to the standard. Part 6 became an IS in May 2002. Part 7 is scheduled for a later date.

Part 6, Mixed Raster Content (MRC), will deal with file format issues for compound images based on a mixed-raster approach, allowing JPEG 2000, JBIG 2 and other coding schemes to be mixed in a common environment.

Part 7 will contain a Technical Report (TR) outlining guidelines for minimum support of Part 1 of the standard. It addresses issues facing hardware implementations of the Part 1 standard in Application-Specific Integrated Circuits (ASIC) and Field Programmable Gate Array (FPGA) applications.

Table 5.17 Division of the JPEG-2000 standard between Part 1 and Part 2 [5.140].

Technology	Part 1	Part 2
Bit stream	Fixed and variable length markers.	New markers can be skipped by a Part 1 decoder.
File format	Optional. Provide intellectual property (for example, copyright) information, color or tone-space for image and general method of including metadata.	Allow metadata to be interleaved with coded data. Define types of metadata.
Arithmetic coder	MQ coder.	MQ coder.
Coefficient modeling	Independent coding of fixed-size blocks within subbands. Division of coefficients into three sub-bit planes. Grouping of sub-bit planes into layers.	Special models for binary or graphic data.
Quantization	Scalar quantizer with dead-zone and truncation of code blocks.	Trellis coded quantization.
Transformation	Low complexity (5,3) and high performance Daubechies (9,7). Mallat decomposition.	Many more filters, perhaps user-defined filters. Packet and other decompositions.
Component decorrelation	Reversible Component Transformation (RCT), YCrCb transform.	Arbitrary point transform or reversible wavelet transform across components.
Error resilience	Resynchronization markers.	Fixed-length entropy coder, repeated headers.
Bit-stream ordering	Progressive by tile part, then SNR, or resolution or component.	Out of order tile parts.

©2000 IEEE.

5.6.2 Overview of Still-Image Coding Standards

In order to present an analytical study of JPEG 2000 functionalities, the following standards will be overviewed: MPEG-4 VTC [5.141], JPEG [5.136], JPEG-LS [5.142], and Portable Network Graphics (PNG) [5.143]. JPEG is one of the most popular coding techniques in imaging applications ranging from Internet to digital photography. Both MPEG-4 VTC and JPEG-LS are very recent standards that have started appearing in various applications. Although PNG is not formally a standard and is not based on the state-of-the-art techniques, it is becoming increasingly popular for Internet-based applications. Although JPEG 2000 supports coding of bilevel and paletted color images, we restrict ourselves to continuous tone because it is one of the most popular image types. Other image coding standards are JBIG [5.144] and JBIG 2 [5.145]. These are known for providing good performance for bilevel images, but they do not support an efficient coding of continuous tone images with a large enough number of levels [5.146].

MPEG-4 VTC

MPEG-4 VTC is the algorithm used in the MPEG-4 standard in order to compress the texture information in photo-realistic 3D models. Because the texture in a 3D model is similar to a still picture, this algorithm can also be used for compression of still images [5.141]. It is based on the DWT, scalar quantization, zero-tree coding and arithmetic coding. MPEG-4 VTC supports SNR scalability through the use of different quantization strategies: Single Quantization (SQ), Multiple Quantization (MQ) and Bilevel Quantization (BQ). SQ provides no SNR scalability. MQ provides limited SNR scalability, and BQ provides generic SNR scalability. Resolution scalability is supported by the use of band-by-band (BB) scanning instead of traditional zero-tree scanning, or Tree-Depth (TD), which is also supported. MPEG-4 VTC also supports coding of arbitrary-shaped objects by means of a shape-adaptive DWT, but does not support lossless coding. Several objects can be encoded separately, possibly at different qualities, and then composited at the decoder to obtain the final decoded image.

JPEG

This is a very well known ISO/ITU-T standard created in the late 1980s. There are several modes defined for JPEG, including baseline, lossless, progressive and hierarchical. Baseline mode is the most popular and supports lossy coding only. It is based on the 8x8 block DCT, zigzag scanning, HVS weighting uniform scalar quantization and Huffman coding. The lossless mode is based on a predictive scheme and Huffman coding [5.135]. The progressive and hierarchical modes of JPEG are both lossy and differ only in the way that the DCT coefficients are coded or computed, respectively, when compared to the baseline mode. They allow a reconstruction of a lower quality or lower resolution version of the image by partial decoding of the compressed bit stream. Progressive mode encodes the quantized coefficients by a mixture of spectral selection and successive approximation, and hierarchical mode uses a pyramidal approach to computing the DCT coefficients in a multiresolution way.

JPEG-LS. JPEG-LS is the latest ISO/ITU-T standard for lossless coding of still images. It also provides for near-lossless compression. It is based on adaptive prediction, context model-

ing and Golomb coding. In addition, it features a flat-region detector to encode these in run-lengths. Near-lossless compression is achieved by allowing a fixed maximum sample error. This algorithm was designed for low complexity while providing high lossless compression ratios. However, it does not provide support for scalability, error resilience or any such functionality.

PNG

PNG is a World Wide Web Consortium (W3C) recommendation for coding of still images. It is based on a predictive scheme and entropy coding. The entropy coding uses the Deflate algorithm of the popular Zip file compression utility, which is based on LZ77 coupled with Huffman coding. PNG is capable of lossless compression only and supports gray scale, paletted color and true color, an optional alpha plane, interlacing and other features.

5.6.3 Significant Features of JPEG 2000

The JPEG 2000 standard provides a set of features that are of vital importance to many high-end and emerging applications by taking advantage of new technologies. It addresses areas where current standards fail to produce the best quality of performance and provides capabilities to markets that currently do not use compression. The most significant features are the possibility to define regions of interest in an image, the spatial and SNR scalabilities, the error resilience and the possibility of intellectual property rights' protection. All these features are incorporated within a unified algorithm [5.146].

Region of Interest (ROI) Coding

One of the features included in JPEG 2000 is the ROI coding. In accordance with this, certain ROIs of the image can be coded with better quality than the rest of the image (background). The ROI scaling-based method scales up the coefficients so that the bits associated with the ROI are placed in higher bit planes. During the embedded coding process, those bits are placed in the bit-stream before the non-ROI parts of the image. Thus, the ROI will be decoded, or refined, before the rest of the image. Regardless of the scaling, a full decoding of the bit stream results in a reconstruction of the whole image with the highest fidelity available. If the bit stream is truncated, or the encoding process is terminated before the whole image is fully encoded, the ROI will have a higher fidelity than the rest of the image [5.147]. The ROI approach defined in JPEG 2000 Part 1 allows ROI encoding of arbitrary-shaped regions without the need of shape information and shape decoding.

Scalability

In general, scalable coding of still images means the ability to achieve coding of more than one resolution and/or quality simultaneously. Scalable image coding involves generating a coded bit stream in a manner that facilitates the derivation of images of more than one resolution and/or quality by scalable decoding. Reasoning that many applications require images to be simultaneously available for decoding at a variety of resolutions or qualities, the architecture supports scalability. If a bit stream is truly scalable, decoders of different complexities, from low-performance decoders to high-performance decoders, can coexist. Although low-performance decod-

ers may decode only small partitions of the bit stream producing basic quality, high-performance decoders may decode much more and produce significantly higher quality. The most important types of scalability are SNR scalability and spatial scalability [5.138, 5.139].

SNR scalability is intended for use in systems with the primary common feature that a minimum of two layers of image quality are necessary. It involves generating at least two image layers of the same spatial resolution, but of different qualities, from a single image source. The lower layer is coded by itself to provide the basic image quality. The enhancement layer, when added back to the lower layer, generates a higher quality reproduction of the input image.

Spatial scalability involves generating at least two spatial resolution layers from a single source so that the lower layer is coded by itself to provide the basic spatial resolution but the enhancement layer employs the spatially interpolated lower layer and carries the full spatial resolution of the input image source. Both types of scalability are very important for Internet and database access applications and bandwidth scaling for robust delivery. The SNR and spatial scalability types include the progressive and hierarchical coding modes already defined in the current JPEG. An additional advantage of spatial and SNR scalability types is their ability to provide resilience to transmission errors because the most important data of the lower layer can be sent across a channel with better error performance, while the less critical enhancement layer data can be sent across a channel with poor error performance.

Error Resilience

To improve the performance of transmitting compressed images across the error-prone channels, error-resilient bit-stream syntax and tools are included in this standard. The error-resilience tools deal with channel errors using data partitioning and resynchronization, error detection and concealment and QoS transmission based on priority [5.138, 5.139]. Many applications require the delivery of image data across different types of communication channels. Typical wireless communication channels give rise to random and burst bit errors. Internet communications are prone to loss due to traffic congestion.

IPRs

An optional file format (JP2) for the JPEG 2000 compressed image data has been defined by the standard. This format has provisions for both image and metadata and a mechanism to indicate the tone scale or color space of the image. This is a mechanism by which readers may recognize the existence of IPR information in the file. Also, it is a mechanism by which metadata, including vendor-specific information, can be included in the file.

5.6.4 Architecture of JPEG 2000

A block diagram of the JPEG 2000 encoder and decoder is illustrated in Figure 5.76.

At first, the discrete transform is applied on the source image data. The transform coefficients are then quantized and entropy coded before forming the output bit stream. The decoder is the reverse of the encoder. The code stream is first entropy decoded, dequantized and inverse discrete transformed, resulting in the reconstructed image data. Depending on the wavelet trans-

form and the applied quantization, JPEG 2000 can be both lossy and lossless. This standard works on image tiles. The term "tiling" refers to the partition of the original (source) image into rectangular nonoverlapping blocks called tiles. They are compressed independently as though they were entirely distinct images. The process of tiling, DC level shifting and DWT of each image tile component is shown in Figure 5.77. This is the strongest form of spatial partitioning because all operations, including component mixing, wavelet transform, quantization and entropy coding, are performed independently on the different tiles of the image.

All tiles have exactly the same dimensions, except maybe those that abut the right and lower boundaries of the image. The nominal tile dimensions are exact powers of two. Tiling reduces memory requirements and constitutes one of the methods for the efficient extraction of a region of the image. Prior to computation of the forward DWT on each tile, all samples of the image tile component are DC level shifted by subtracting the same quantity (that is, the component depth) from each sample. The tile components are decomposed into different decomposition levels using a wavelet transform. These decomposition levels contain a number of subbands populated with coefficients that describe the horizontal and vertical spatial frequency character-

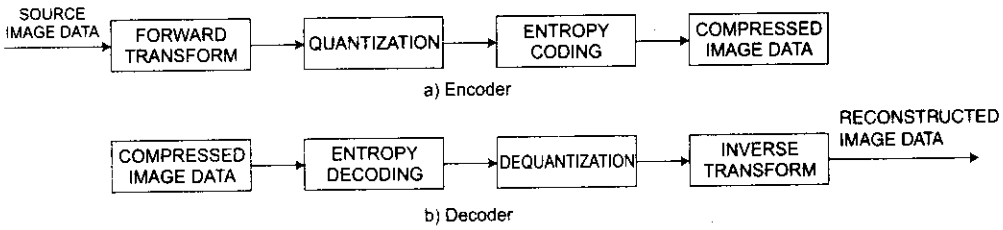


Figure 5.76 Block diagram of JPEG 2000 a) encoder and b) decoder architecture [5.35].

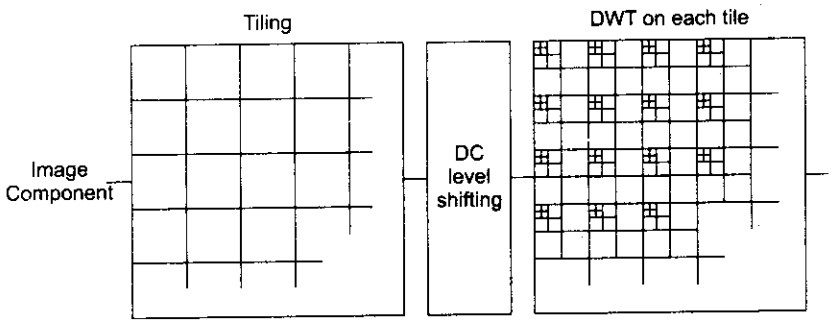


Figure 5.77 Tiling, DC level shifting and DWT on each image tile component [5.125]. ©2001 IEEE.

istics of the original tile component planes as shown in Figure 5.77. The coefficients provide local frequency information. A decomposition level is related to the next decomposition level by spatial powers of two. To perform the forward DWT, the standard uses a 1D subband decompo-

sition of a 1D set of samples into low-pass samples and high-pass samples, representing a down-sampled residual version of the original set needed for the perfect reconstruction of the original set from the low-pass set. Any user supplied wavelet filter banks may be used [5.147, 5.148]. The DWT can be irreversible or reversible. The default irreversible transformation is implemented by means of the Daubechies 9/7-tap filter. The analysis and the corresponding synthesis filter coefficients are given in Table 5.18. The default reversible transformation is implemented by means of the 5-tap/3-tap filter, the coefficients of which are given in Table 5.19.

Table 5.18 Daubechies 9/7 analysis and synthesis filter coefficients [5.154].

Analysis filter coefficients		
i	**Low-pass filter $h_L(i)$**	**High-pass filter $h_H(i)$**
0	0.6029490182363579	1.115087052456994
+/-1	0.2668641184428723	-0.5912717631142470
+/-2	-0.07822326652898785	-0.05754352622849957
+/-3	-0.01686411844287495	0.09127176311424948
+/-4	0.02674875741080976	N/A
Synthesis filter coefficients		
i	**Low-pass filter $h_L(i)$**	**High-pass filter $h_H(i)$**
0	1.115087052456994	0.6029490182363579
+/-1	0.5912717631142470	-0.2668641184428723
+/-2	-0.05754352622849957	-0.07822326652898785
+/-3	-0.09127176311424948	0.01686411844287495
+/-4	N/A	0.02674875741080976

©2000 IEEE.

Figure 5.78 Periodic symmetric extension of the signal ABCDEFG [5.154]. ©2000 IEEE.

The standard supports two filtering models: convolution based and lifting based. For both modes the signal should be first extended periodically as shown in Figure 5.78. The periodic symmetric extension is used to ensure that, for filtering operations that take place at both boundaries of the signal, one signal sample exists and spatially corresponds to each coefficient of the filter mask. For illustration purposes, it has been assumed that, for the signal ABCDEFG, we have A>B>...>G. The number of additional samples required at the boundaries of the signal is filter-length dependent [5.149].

Convolution-based filtering consists of creating a series of dot products between the two filter masks and the extended 1D signal. Lifting-based filtering consists of a sequence of very simple filtering operations for which alternatively odd sample values of the signal are updated with a weighted sum of even sample values. On the other hand, even sample values are updated with a weighted sum of odd sample values [5.139, 5.150]. For the reversible (lossless) case, the results are rounded to integer values.

Example 5.15 The lifting-based filtering for the 5/3 analysis filter is achieved as follows:

$$y(2n+1) = x_{ext}(2n+1) - \left\lfloor \frac{x_{ext}(2n) + x_{ext}(2n+2) - 1}{2} \right\rfloor \tag{5.3}$$

Table 5.19 5/3 analysis and synthesis filter coefficients [5.154].

Analysis filter coefficients		
I	**Low-pass filter $h_L(i)$**	**High-pass filter $h_H(i)$**
0	6/8	1
+/-1	2/8	-1/2
+/-2	-1/8	N/A
Synthesis filter coefficients		
I	**Low-pass filter $h_L(i)$**	**High-pass filter $h_H(i)$**
0	1	6/8
+/-1	1/2	-2/8
+/-2	N/A	-1/8

©2000 IEEE.

$$y(2n) = x_{est}(2n) + \left\lfloor \frac{y(2n-1) + y(2n+1) + 2}{4} \right\rfloor \tag{5.4}$$

where x_{est} is the extended input signal, y is the output signal, and $\lfloor a \rfloor$ and $\lceil a \rceil$ indicate the largest integer not exceeding a and the smallest integer not exceeded by a, respectively.

Quantization is the process by which the coefficients are reduced in precision. This operation is lossy unless the quantization step is 1 and the coefficients are integers as produced by the reversible integer 5/3 wavelet. Each of the transform coefficients $a_b(u,v)$ of the subband b is quantized to the value $q_b(u,v)$ according to the formula

$$q_b(u,v) = sign\left(a_b(u,v)\right) \left\lfloor \frac{|a_b(u,v)|}{\Delta_b} \right\rfloor \tag{5.5}$$

where Δ_b is the quantization step [5.122]. The dynamic range depends on the number of bits used to represent the original image tile component and on the choice of the wavelet transform. All quantized transform coefficients are signed values even when the original components are unsigned. These coefficients are expressed in a sign-magnitude representation prior to coding. Although Part 1 of the JPEG 2000 standard uses only simple scalar dead-zone quantization, significant data size reduction can also be obtained by throwing away portions of the data. Part 2 of the standard contains a TCQ [5.151, 5.152]. This technology has a fairly high encoding cost, but adds a minimal amount of complexity for a decoder and produces higher quality images, and sometimes does a better job visually. However, the improvement may not be noticeable in terms of SNR. Each subband of the wavelet decomposition is divided into rectangular blocks, called code blocks, which are coded independently using arithmetic coding. A binary arithmetic entropy coder called the MQ coder is used to provide compression of symbols' output by the context model. The complexity and compression are much higher than the typically used Huffman coder in JPEG [5.139]. The code blocks are coded one bit plane at a time, starting with the most significant bit plane with a nonzero element to the least significant bit plane. For each bit-plane in a code block, a special code-block scan pattern is used for each of three passes. Each coefficient bit in a bit plane is coded in only one of the three passes. A rate distortion optimization method is used to allocate a certain number of bits to each block.

JPEG 2000 supports multiple-component images. Different components need not have the same bit depths nor need they have all been signed or unsigned. The standard supports two different component transformations: one Irreversible Component Transformation (ICT) and one RCT. It is usual that the input image has three components: Red, Green and Blue (RGB). The block diagram of the JPEG 2000 multiple-component encoder is shown in Figure 5.79. Here, C_1, C_2 and C_3 represent, in general, the color-transformed output components. If needed, prior to applying the forward-color transformation, the image component samples are DC level shifted. The ICT may be used only for lossy coding. It can be seen as an approximation of a YCbCr transformation of the RGB components.

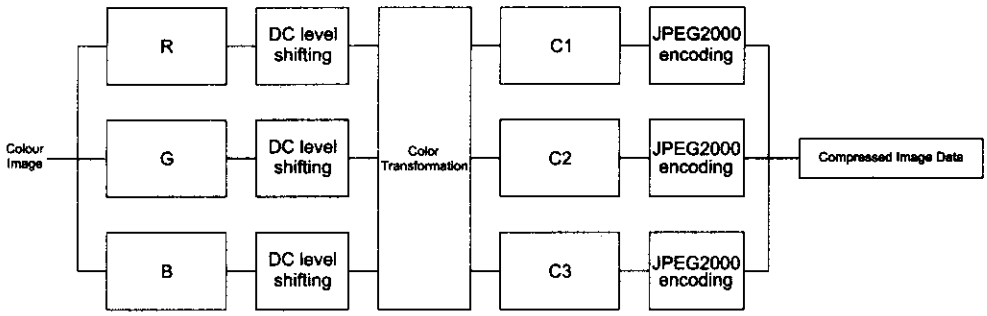


Figure 5.79 JPEG 2000 multiple-component encoder.

The forward and the inverse ICT transformations are achieved from the following equations:

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.1687 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{5.6}$$

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1.0 & 0 & 1.402 \\ 1.0 & -0.34413 & -0.71414 \\ 1.0 & 1.772 & 0 \end{pmatrix} \begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} \tag{5.7}$$

The RCT may be used for lossy or lossless coding. It is a decorrelating transformation, which is applied to the three first components of an image. Three goals are achieved by this transformation:

- Color decorrelation for efficient compression
- Reasonable color space with respect to the human visual system for quantization
- Ability of having lossless compression, that is, exact reconstruction with finite integer precision

For the RGB components, the RCT can be seen as an approximation of a YUV transformation. The forward and inverse RCT is performed by means of

$$\begin{pmatrix} Yr \\ Ur \\ Vr \end{pmatrix} = \begin{pmatrix} \left\lfloor \frac{R+2G+B}{4} \right\rfloor \\ R-G \\ B-G \end{pmatrix} \tag{5.8}$$

that is,

$$\begin{pmatrix} G \\ R \\ B \end{pmatrix} = \begin{pmatrix} Yr - \left\lfloor \frac{Ur+Vr}{4} \right\rfloor \\ Ur+G \\ Vr+G \end{pmatrix} \tag{5.9}$$

Part 1 of the JPEG 2000 standard contains the YCrCb transform used in the original JPEG standard. It also includes an RCT useful for lossless compression of three-component color imagery. Part 2 contains the ability to do an arbitrary joint transform to decorrelate components. This is essential for good compression on multi- and hyperspectral imagery. A standard block diagram of a JPEG 2000 coder and the correspondence with the Annexes of the WD of the standard are shown in Figure 5.80. An encoder starts at the left of the figure with an image and produces a code stream at the right. A decoder works in the opposite direction.

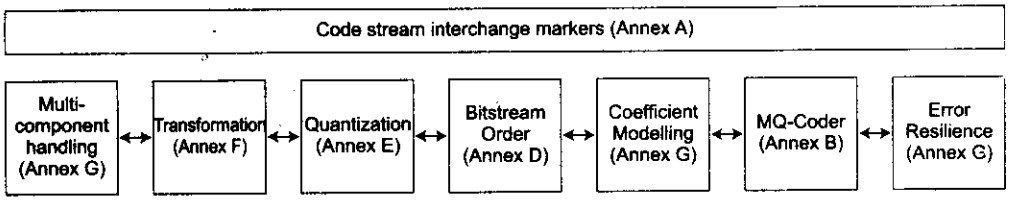


Figure 5.80 Standard block diagram of a JPEG 2000 coder and the correspondence with the Annexes of the working draft [5.159]. ©1998 ISO/IEC.

5.6.5 JPEG 2000 Bit Stream

JPEG 2000 provides better rate-distortion performance than the original JPEG standard for any given rate. The improvements in the near visually lossless realm are more modest, approximately 20% [5.140].

There are four basic dimensions of progression in the JPEG 2000 bit stream: resolution, quality, spatial location and component. Different types of progression are achieved by the ordering of packets within the bit stream. Although this provides an important mechanism for spatial progression, we assume for simplicity that the image consists of a single tile. Each packet is then associated with one component (say i), one layer (j), one resolution level (k) and one packet partition location (m).

Example 5.16 Let us say that an image is divided into tiles, and each tile is transformed. The subbands of a tile are divided into packet partition locations. Finally, each packet partition location is divided into code blocks. This is illustrated in Figure 5.81 where 12 code blocks of one packet partition location at resolution level 2 of a 3-level dyadic wavelet transform are given. The packet partition location is emphasized by heavy lines. The division of one packet partition location into 12 code blocks is also shown.

Figure 5.82 depicts one packet for the packet partition location illustrated in the previous figure. Each of the 12 code blocks can contribute a different number of sub-bit planes (possibly zero) to the packet. Empty packet bodies are allowed.

A packet can be interpreted as one quality increment for one resolution level at one spatial location. Packet partition locations correspond roughly to spatial locations. A layer is then a collection of packets: one from each packet partition location of each resolution level. A layer then can be interpreted as one quality increment for the entire image at full resolution. Each layer

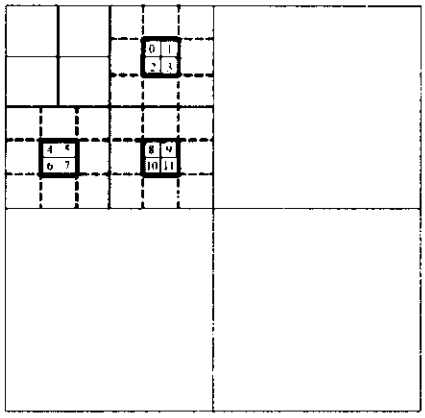


Figure 5.81 Twelve code blocks of one packet partition location at resolution level 2 of a three-level dyadic wavelet transform. The packet partition location is presented by heavy lines.

Packet Header	n_0 sub-bitplanes from code-block 0	n_1 sub-bitplanes from code-block 1	.....	n_{11} sub-bitplanes from code-block 11

Figure 5.82 The composition of one packet partition location with 12 code-blocks.

provides more bits of some of the wavelet coefficients. The role of layers in providing progression by SNR has been described [5.140]. The layers need not be designed specifically for optimal SNR progression. JPEG 2000 does not explicitly define a method of subsampling color components as JPEG does. A JPEG 2000 encoder could place all the high frequency bands of the color components in the last layer. A decoder that did not receive high frequency subbands could use a simplified transform to save computational complexity. For images with significant color edges, some bits of the color coefficients might be saved in earlier layers.

The JPEG 2000 bit stream contains markers that identify the progression type of the bit stream. Other markers may be written that store the length of every packet in the bit stream. To change a bit stream from progressive by resolution to progressive by SNR, a parser can read all the markers, change the type of progression in the markers, write the lengths of the packets out in the new order, and write the packets themselves out in the new order. There is no need to run the MQ coder or the context model or even to decode the block-inclusion information. The complexity is only slightly higher than a pure copy operation.

If the ROIs are known in advance, that is, at encode time, JPEG 2000 provides additional methods of providing greater image quality in the foreground verses the background. First, all the code blocks that contain coefficients affecting the ROI can be identified, and the bit planes of these coefficients can be stored in higher layers relative to other coefficients. Thus, a layer-progressive bit stream can naturally send the ROI with higher quality than the background. In addition, an explicit ROI can be defined, and those coefficients that affect the ROI can be shifted and coded as if they were in their own set of bit planes. For an encoder, this allows individual coefficients to be enhanced rather than entire code blocks, If the ROIs are not known at encoding time, there are still several methods for a smart server to provide

exactly the right data to a client requesting a specific region. The simplest method to provide access to spatial regions of the image (which are not known at encoding time) is for the encoder to tile the image. Because tiling divides the image spatially, any region desired by the client will lie within one or more tiles. Tiles as small as 64 by 64 are useable, but tiles this small increase the bit rate noticeably. Tiles greater than 256 by 256 samples have almost no compression performance impact, but offer less flexible access for small regions. All of the parsing operations on the whole image can selectively be applied to specific tiles. Other tiles could be discarded or transmitted at a much lower quality. The bitstream contains the length of each tile, so it is always possible to locate the desired tiles with minimal complexity. Similarly, packet partitions can be extracted from the bit stream for spatial access. The length information is still stored in the tile header, and the data corresponding to a packet partition location is easily extracted. Finer grain access is possible by parsing individual code blocks. As in the case of packet partition locations, it is necessary to determine which code blocks affect which pixel locations. The correct packets containing these code blocks can be determined from the progression-order information. Finally, the location of the compressed data for the code blocks can be determined by decoding the packet headers.

All uncompressed tiled image formats allow regions of an image to be edited, and only those tiles affected need to be rewritten to disk. With compression, the compressed size of an edited tile can change. Because of the flexibility in quantization in JPEG 2000, it is possible to truncate an edited tile to fit in the previous size. Alternatively, Part 2 of the standard allows out-of-order tiles within the bit stream, so an edited tile could be rewritten at the end of the bit stream. The main header of a JPEG 2000 bit stream contains the width and height of the image. It also contains a horizontal and vertical offset for the start of the image. This allows the image to be cropped to a subrectangle of the original without requiring a forward and inverse wavelet transform for recompression. All tiles inside the newly cropped image need not be changed at all, and tiles on the edge of the new image need only to have the code blocks on the edges recorded. New tile headers and packet headers are written to the bit stream (no wavelet transform). Finally, the integer nature (5.3) of the wavelet allows an image or partition of an image to be compressed multiple times with the same quantization with no additional loss. This is the only time that the decompressed sample values are not clipped when they fall outside the full dynamic range (for example, 0 to 255 for 8-bit images).

5.6.6 Compression Efficiency Comparisons

Compression efficiency is one of the top priorities in the design of image products [5.153]. In Santa-Cruz and Ebrahimi [5.146], lossless and lossy progressive compression and efficiency results to evaluate how well the algorithms code different types of imagery are presented. The support of progressive coding is included, too. The algorithms have been evaluated with seven images from the JPEG 2000 test set, covering various types of imagery. The images "Bike" (2048x2560) and "Café" (2048x2560) are natural, "Cmpnd" (512x768) and "Chart" (1688x2347) are compound documents consisting of the text, photographs and computer graph-

ics, "Aerial2" (2048x2048) is an aerial photography, "Target" is a computer-generated image and "US" (512x448) is an ultra scan. All these images have a depth of 8 bits per pixel.

Table 5.20 summarizes the lossless compression efficiency of lossless JPEG (L-JPEG), JPEG-LS, PNG and JPEG 2000 for all the test images. For JPEG 2000, the reversible DWT filter, refered to as $J2K_R$, has been used. In the case of L-JPEG, optimized Huffman tables and the predictor yielding the best compression performance have been used for each image. For PNG, the maximum compression setting has been used, and for JPEG-LS, the default options were chosen. MPEG-4 VTC is not considered because it does not provide a lossless functionality. It can be seen that, in almost all cases, the best performance is obtained by JPEG-LS. JPEG 2000 provides, in most cases, competitive compression ratios with the added benefit of scalability. PNG performance is similar to JPEG 2000. PNG provides the best results for the "Target" image. JPEG-LS and PNG achieve much larger compression ratios for the "Cmpnd" image. This image contains, for the most part, block text on a white background. PNG performs the best although this is solely due to the very large compression ratio that it achieves on target. However, JPEG-LS provides the best compression ratio for most images. To conclude, as far as lossless compression is concerned, JPEG 2000 seems to perform reasonably well in terms of its ability to deal efficiently with various types of images. However, in specific types of images such as "Cmpnd," JPEG 2000 is outperformed by far by JPEG-LS. This result is even more striking, noting that JPEG-LS is a significantly less complex algorithm.

Table 5.20 Lossless compression ratios for seven test images [5.146].

Image	$J2K_R$	JPEG-LS	L-JPEG	PNG
Bike	1.77	1.84	1.61	1.66
Café	1.49	1.57	1.36	1.44
Cmpnd1	3.77	6.44	3.23	6.02
Chart	2.60	2.82	2.00	2.41
Aerial2	1.47	1.51	1.43	1.48
Target	3.76	3.66	2.59	8.70
US	2.63	3.04	2.41	2.94
Average	2.50	2.98	2.09	3.52

©2000 IEEE.

The rate distortion behavior of lossy (nonreversible) JPEG 2000 and progressive JPEG is depicted in Figure 5.83 for a natural image. It is seen that JPEG 2000 significantly outperforms the JPEG scheme [5.122]. It can be calculated that, for similar Peak SNR (PSNR) quality, JPEG 2000 compresses almost twice as much as JPEG.

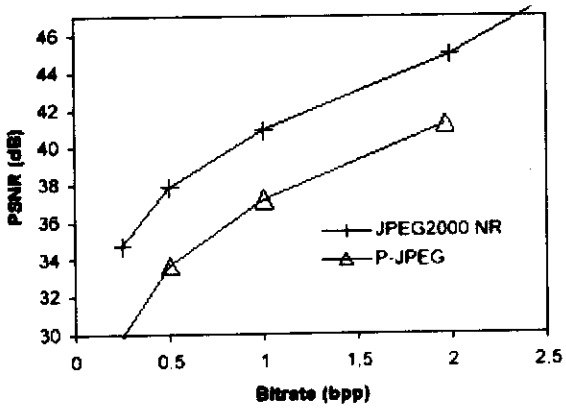
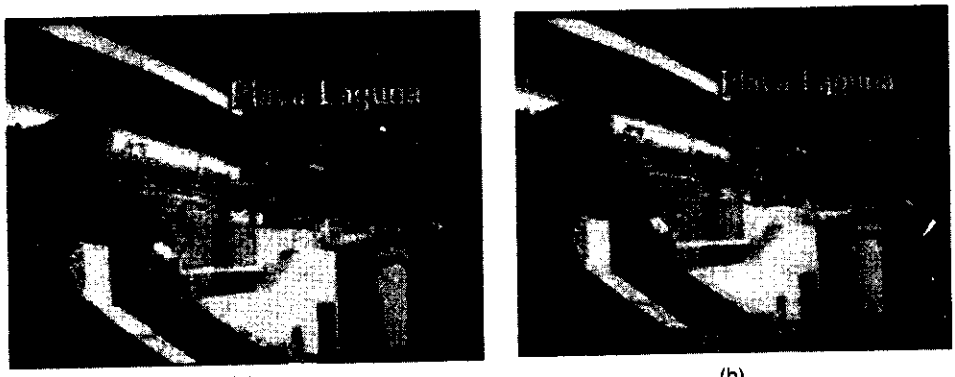


Figure 5.83 Rate distortion results for progressive JPEG 2000 versus progressive JPEG for a natural image [5.122]. ©2000 IEEE.



(a) (b)

Figure 5.84 Reconstructed images compressed at 0.125 bpp by means of (a) JPEG and (b) JPEG 2000 [5.122]. ©2000 IEEE.

The superiority of JPEG 2000 can be subjectively judged with the help of Figure 5.84 where the reconstructed image "Hotel" (720x576) is shown. Both images were compressed at a rate of 0.125 bpp using JPEG and JPEG 2000. The degradation of the image in Figure 5.84(a) is evident.

In order to evaluate the error-resilience features offered by the different standards, a transmission channel with random errors has been simulated [5.146] together with the evaluation of the average reconstructed image quality after decompression. Table 5.21 shows the results for JPEG 2000 with irreversible wavelet transform and JPEG baseline. Only the results of "Café" image are shown. The behavior is very similar for the other images. As it can be seen, the recon-

Table 5.21 PSNR, in dB, corresponding to average Root MSE (RMSE) of 200 runs of the decoded "Café" image when transmitted across a noisy channel with various bit error rates (BER) and compression bit rates for JPEG baseline and JPEG 2000 (J2K) [5.146].

bpp	IS	BER 0	BER 1E-06	BER 1E-05	BER 1E-04
0.25	J2K	23.06	23.00	21.62	16.59
	JPEG	21.94	21.79	20.77	16.43
0.5	J2K	26.71	26.42	23.96	17.09
	JPEG	25.40	25.12	22.95	15.73
1.0	J2K	31.90	30.75	27.08	16.92
	JPEG	30.34	29.24	23.65	14.80
2.0	J2K	38.91	36.38	27.23	17.33
	JPEG	37.22	30.68	20.78	12.09

©2000 IEEE.

structed image quality under transmission errors is higher for JPEG 2000 than JPEG across all encoding bit rates and error rates. However, at low bit rates (0.25 and 0.5 bpp), the quality of JPEG 2000 decreases more rapidly than JPEG as the errors increase, although the absolute quality is always higher. Concerning the visual quality at moderately low error rates (that is, 1E-06), for JPEG 2000 it is much higher when compared to JPEG. It should also be noted that at higher error rates (that is, 1E-04), the reconstructed image quality in JPEG 2000 is almost constant across all bit rates. This is due to the fact that, in JPEG 2000, each subband block is coded by bit planes. When the error rate is high, almost all blocks are effected in the most significant bit planes, which are transmitted first. When particular bit planes are affected in a block, lower bit planes cannot be decoded and are therefore useless. In the case of JPEG, the problem is even worse. The higher the encoding bit rate means the lower the decoded quality. This can be explained by the fact that, when an 8x8 block is affected by a transmission error, the entire block is basically lost. The higher the encoding bit rate means the more bits it takes to code a block. Therefore, the probability of a block being hit by an error and lost is higher for the same bit error rate. In other words, in JPEG, the density of error protection decreases with an increase in bit rate.

Many applications require features in a coding algorithm other than simple compression efficiency. This is often referred to as functionalities. Table 5.22 summarizes the comparison from a functionality point of view. A functionality matrix is provided in Santa-Cruz and Ebrahimi [5.146]. It indicates the set of supported features in each standard.

This table clearly shows that JPEG 2000 is the standard offering the richest set of features in an efficient manner and within an integrated algorithmic approach. MPEG-4 VTC (also JPEG 2000) is able to produce progressive bit streams without any noticeable overhead. However, the

Table 5.22 Functionality matrix. A + indicates that it is supported. The more +s means the more efficiently or better it is supported. A - indicates that it is not supported [5.146].

Functionality	JPEG 2000	JPEG-LS	JPEG	MPEG-4 VTC	PNG
Lossless compression performance	+++	++++	+[1]	-	+++
Lossy compression performance	+++++	+	+++	++++	-
Progressive bit streams	+++++	-	++[2]	+++	+
ROI coding	+++	-	-	+[3]	-
Arbitrary-shaped objects	-	-	-	++	-
Random access	++	-	_	-	-
Low complexity	++	+++++	+++++	+	+++
Error resilience	+++	++	++	+++	+
Noniterative rate control	+++	-	-	+	-
Genericity[4]	+++	+++	++	++	+++

[1]Only using the lossless mode of JPEG.

[2]Only in the progressive mode of JPEG.

[3]Tile-based only.

[4] Ability to compress different types of imagery efficiently across a wide range of bit rates.

©2000 IEEE.

latter provides more progressive options and produces bit streams that are parseable and that can be rather easily reorganized by a transcoder. Along the same lines, JPEG 2000 also provides random access to the block level in each subband, thus making it possible to decode a region of the image without having to decode it as a whole. These features could be very advantageous in applications such as digital libraries.

Error Resilience

Concerning error resilience, JPEG 2000 offers higher protection than JPEG, as shown in the previous section. MPEG-4 VTC also offers error-resilience features and, although it could not be evaluated, the support should be in-between JPEG and JPEG 2000. JPEG-LS does not offer any particular support for error resilience besides restart markers and has not been designed with it in mind. As for PNG, it offers error detection, but no concealment possibilities.

Overall, one can say that JPEG 2000 offers the richest set of features and provides superior rate-distortion performance. However, this comes at the price of additional complexity when compared to JPEG and JPEG-LS, which might be currently perceived as a disadvantage for some applications, as was the case for JPEG when it was first introduced.

5.7 MPEG-7 Standardization Process of Multimedia Content Description

MPEG-7, formally named Multimedia Content Description Interface, is the standard that describes multimedia content so that users can search, browse and retrieve the content more efficiently and effectively than they could by using existing mainly text-based search engines [5.155]. It is a standard for describing the features of multimedia content. The word "features" or "descriptions" represents a rich concept that can be related to several levels of abstraction. Descriptions vary according to the types of data. Furthermore, different types of descriptions are necessary for different purposes of categorization. MPEG-7 will specify a standard set of descriptors that can be used to describe various types of multimedia information. Also, MEPG-7 will standardize ways to define other descriptors as well as structures for the descriptors and their relationships. This description will be associated with the content to allow fast and efficient searching for material of the user's interest. A language to specify description schemes, that is, a DDL, will be standardized, too. Audiovisual material that has MPEG-7 data associated with it can be indexed and searched for. This material includes still pictures, graphics, 3D models, audio, speech, video and information about how these elements are combined in a multimedia presentation. Special cases of these general data types may include facial expressions and personal characters [5.155, 5.156].

Different people want to use the audiovisual information for various purposes. However, before the information can be used, it must be located. At the same time, the increasing availability of potentially interesting material makes this search more difficult. This challenging situation led to the need for a solution to the problem of quickly and efficiently searching for various types of multimedia material of interest to the user. The MPEG-7 standard wants to answer to this need and to provide the solution [5.157].

MPEG-7 is rather different from the other MPEG standards because it does not define a way to represent data with the objective to reconstruct the data as faithfully as possible like MPEG-1, MPEG-2 and MPEG-4 did. The increasingly pervasive role that audiovisual sources are destined to play in our lives and the growing need to have these sources further processed make it necessary to develop forms of audiovisual information representation that go beyond the simple waveform or pixel-based, frame-based (such as MPEG-1 and MPEG-2) or even object-based (such as MPEG-4) representations. This necessitates forms of representation that allow some degree of interpretation of the information's meaning, which can be passed on to, or accessed by, a device or a computer code. The people active in defining the MPEG-7 standard represent broadcasters, equipment and chip manufacturers, digital content creators and managers, telecommunication service providers, publishers, IPR managers and researchers.

5.7.1 Objective of the MPEG-7 Standard

The objective of MPEG-7 is to set a standard for the description of multimedia material. This includes speech, audio, video, still pictures and 3D models. It also includes information about how these elements are combined in a multimedia scene, presentation or document. MPEG-7

will define a number of elements: descriptors, description schemes, a DDL, system tools and coding schemes for the descriptions. These are the normative elements of MPEG-7. These parts need to be specified to ensure the interoperability between MPEG-7-enabled systems. Before defining these elements, we will first address the definitions of the key concepts of data and feature [5.158]. Data refers to Audio Visual information that will be described using MPEG-7, regardless of storage, coding, display, transmission medium or technology. Examples are an MPEG-4 stream; a video tape; a CD containing music, sound or speech; a picture printed on paper, and an interactive multimedia presentation on the Web.

A feature is a distinctive characteristic of the data that signifies something to somebody. Some examples are color of an image, pitch of a speech segment, rhythm of an audio segment, camera motion in a video, style of a video, the title of a movie, the actors in a movie, and so forth.

A Descriptor (D) is a representation of a feature. It defines the syntax and the semantics of the feature representation. Possible descriptors are the color histogram, the average of the frequency components, the motion field, the text of the title, and so forth. A D value is an instantiation of a D for a given dataset or subset.

A Description Scheme (DS) specifies the structure and semantics of the relationships between its components, which may be both Ds and DSs schemes. Examples are a movie, temporally structured as scenes and shots, including some textural descriptors at the scene level, and color, motion and audio Ds at the shot level. A description consists of a DS structure and the set of descriptor values that describe the data. A coded description is a description that has been encoded to fulfill relevant requirements, such as compression efficiency, error resilience, random access, and so forth. DDL is a language that allows the creation of new description schemes and possibly, descriptors. It also allows the extension and modification of existing description schemes. Figure 5.85 gives a graphical view of the relation between the different MPEG-7 elements and their relations [5.157].

MPEG-7 addresses many applications and many types of usage. The standard will address real-time and non-real-time applications, interactive and unidirectional (broadcast) and online as well as offline usage. In this context, a real-time environment means that descriptions are being associated with the content while it is being captured. MPEG-7 descriptions will support query modalities such as text-based only, subject navigation, interactive browsing, visual navigation and summarization, search by example, and using features and sketches.

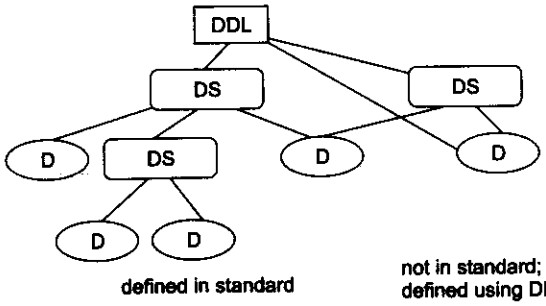


Figure 5.85 Elements of the MPEG-7 standard [5.156]. ©2000 ISO/IEC.

Which modality to use depends on the task at hand and the application environment. The description definition of a piece of multimedia material does not exist. First, more than one descriptor may exist to represent the same feature, fulfilling different requirements. Second, and more important in this context, the exact description depends very much on the application and the user. MPEG-7 will not define what description is right for a certain body of content, but only gives the tools to represent such a description. In this sense, MPEG-7 follows the policy adopted for previous MPEG standards: The analysis engine and the encoder will not be standardized. An analogy exists with MPEG-4 that can represent arbitrary-shaped VOs. MPEG-4 only specifies the syntax and semantics to represent a shape and defines how that representation should be decoded, regardless of how it was obtained. The same applies to the systems that make use of the MPEG-7 descriptions, such as search engines and filters, which reside on the other end of a possible MPEG-7 processing chain. The scope of the MPEG-7 standard using a simplified processing chain is shown in Figure 5.86. MPEG-7 will have the possibility to denote spatiotemporal entities in non-MPEG-4 content.

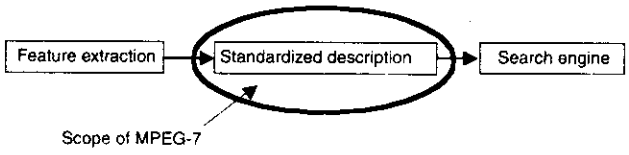


Figure 5.86 A possible processing chain and scope of the MPEG-7 standard [5.156]. ©2000 ISO/IEC.

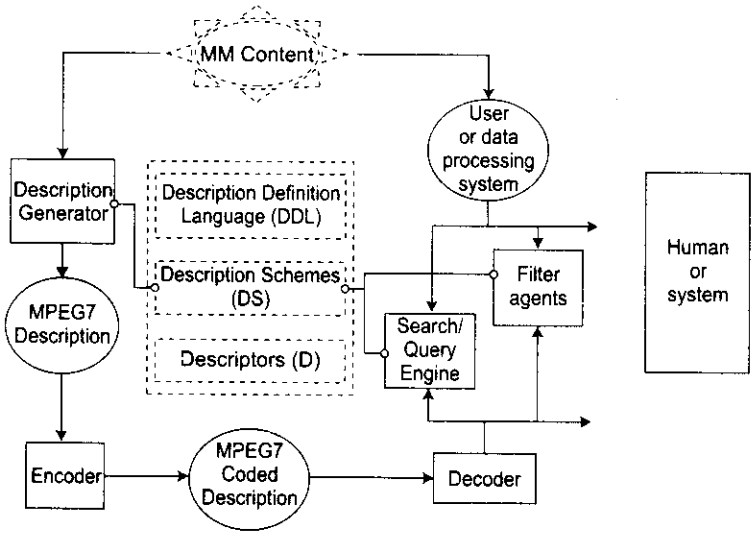


Figure 5.87 Representation of possible applications using MPEG-7 [5.156]. ©2000 ISO/IEC.

In comparison with other available or emerging solutions for multimedia description, MPEG-7 can be characterized by the following:

- *Its generality*—The capability to describe content from many application environments
- *Its object-based data model*—The capability of independently describing individual objects within a scene
- *Its integration of low- and high-level features/descriptors into a single architecture*—The capability to combine the power of both types of descriptors
- *Its extensibility provided by the DDL*—The capability to keep growing, to be extended to new application areas, to answer newly emerging needs and to integrate novel description tools

Figure 5.87 explains a hypothetical MPEG-7 chain in practice. The circular boxes depict tools that are implementing functions, such as encoding or decoding, and the square boxes represent static elements, such as a description. The dotted boxes in the figure encompass the normative elements of the MPEG-7 standard. Besides the descriptors themselves, the database structure plays a crucial role in the final retrieval's performance. To allow the desired fast judgment about whether the material is of interest, the indexing information will have to be structured, for example, in a hierarchical or associative way.

5.7.2 Status of the MPEG-7 Standard

Currently, MPEG-7 concentrates on the specification of description tools together with the development of the MPEG-7 reference software, known as XM (eXperimentation Model).

The MPEG-7 Audio group develops a range of description tools from generic audio descriptors to more sophisticated description tools like spoken content. Generic audio description tools will allow the search for similar voices, by searching similar envelopes and fundamental frequencies of a voice sample against a database of voices. The spoken content description scheme is designed to represent the output of a great number of state-of-the-art automatic speech-recognition systems, containing both words and phoneme representations and transition likelihoods. This alleviates the problem of out-of-vocabulary words, allowing retrieval even when the original words were wrongly decoded.

The MPEG-7 Visual group is developing four groups of description tools: color, texture, shape and motion. Color and texture description tools will allow the searching and filtering of visual content (images, graphics and video) by dominant color or textures in the same (arbitrarily shaped) regions or the whole image. Shape description tools will facilitate query by sketch or by contour similarity in an image database or, for example, searching trademarks in registration databases. Motion description tools will allow searching of videos with similar motion patterns that can be applicable to news or to surveillance applications.

The Multimedia Description Schemes group is developing the description tools dealing with generic and audiovisual and archival features. Its central tools deal with content management and content description. Content management description tools cover the viewpoints of

media creation and production and usage. Media description tools allow searching for preferred storage formats, compression qualities and aspect ratios, among others. Creation and prediction description tools cover the typical archival and credit information (for example, title, creators and classification). Usage description tools deal with description related to the use of the described content (for example, rights, broadcasting, dates and places, availability, audience and financial data). The content description covers both structural and conceptual viewpoints. Structural description tools provide segmentation, both spatial and temporal, of the content. Among other functionalities, this allows assigning descriptions to different regions and segments and providing importance rating of temporal segments and regions. Conceptual description tools allow providing of a semantic-based description besides the content description and content management description tools. Other tools target content organization, navigation, access and user preferences.

5.7.3 Major Functionalities in MPEG-7

The MPEG-7 standard consists of the following parts [5.157, 5.158]:

- ***Part 1: Systems***—The tools that are needed to prepare MPEG-7 descriptions for efficient transport and storage and to allow synchronization between content descriptions. There are also tools related to managing and protecting intellectual property.
- ***Part 2: DDL***—The language for defining new description schemes and perhaps also new descriptors.
- ***Part 3: Audio***—The descriptors and description schemes dealing with only audio descriptors.
- ***Part 4: Visual***—The descriptors and description schemes dealing with visual descriptors.
- ***Part 5: Multimedia Description Schemes***—The descriptors and description schemes dealing with generic features and multimedia descriptions.
- ***Part 6: Reference Software***—A software implementation of relevant parts of the MPEG-7 standard.
- ***Part 7: Conformance***—Guidelines and procedures for testing conformance of MPEG-7 implementations.

In what follows, the major functionalities that the different parts of the MPEG-7 standard offer are described.

MPEG-7 Systems

This part includes the tools that are needed to prepare MPEG-7 descriptions for efficient transport and storage and to allow synchronization between content and tools related to managing and protecting intellectual property. It defines the terminal architecture and the normative interfaces. The information representation specified in the MPEG-7 standard provides the means to

represent coded multimedia content description information. The entity that makes use of such coded representation of the multimedia content is referred to as a terminal. The architecture of a terminal making use of MPEG-7 representations is depicted in Figure 5.88. A transmission/storage medium refers to the lower layers of the delivery infrastructure. These layers deliver multiplexed streams to the delivery layer. The transport of the MPEG-7 data can occur on a variety of delivery systems. For example, this includes MPEG-2 transport streams, IP or MPEG-4 (MP4) files or streams. The Delivery Layer encompasses mechanisms allowing synchronization, framing and multiplexing of MPEG-7 content. The MPEG-7 architecture allows conveying data such as queries or request back from the terminal to the transmitter or server. The Delivery Layer provides MPEG-7 ES to the Compression Layers. They consist of consecutive individually accessible portions of data named access units. An access unit is the smallest data entity to which timing information can be attributed. Description information is either a complete description of the multimedia content or a fragment of the description. MPEG-7 data can be represented either in textual format, in binary format or in a mixture of the two formats, depending on application usage. The syntax of the textual format is defined in Part 2 (DDL) of the standard [5.157]. The syntax of the binary format for MPEG-7 data (BiM) is defined in Part 1 (Systems) of the standard. At the Compression Layer, the flow of access units (either textual or binary encoded) is parsed, and the content description is reconstructed. The MPEG-7 binary stream can be either parsed by the BiM parser and transformed in textual format and then transmitted in textual format to further reconstruction processing, or the binary stream can be parsed by the BiM parser and then transmitted in proprietary format to further processing. MPEG-7 access units are further structured as commands encapsulating the scheme or the description information. Commands provide the dynamic aspects of the MPEG-7 content. They allow a description to be delivered in a single chunk or to be fragmented in small pieces. They allow basic operations on the MPEG-7 content, such as updating a descriptor, deleting part of the description or adding new DDL structure.

MPEG-7 normative interfaces are presented in Figure 5.89. Content refers either to essence or to content description. An MPEG-7 binary/textual encoder transforms the content into a compliant format. A textual format interface describes the format of the textual units. The MPEG-7 textual decoder consumes a flow of such access units and reconstructs the content description in a normative way. A binary format interface describes the format of the binary access units. The MPEG-7 binary decoder consumes a flow of such access units and reconstructs the content description in a normative way. An MPEG-7 binary/textual decoder transforms data into a content description.

The question often arises as to how proof can be established that the binary representation and textual representation provide dual representations of the content. The process is described in Figure 5.90. In addition to the elements described in MPEG-7 normative interfaces, the validation process involves the definition of a canonical representation of a content description. The validation process works as follows. A content description is encoded in a lossless way in textual and in binary format, generating two different representations of the same entity. The two

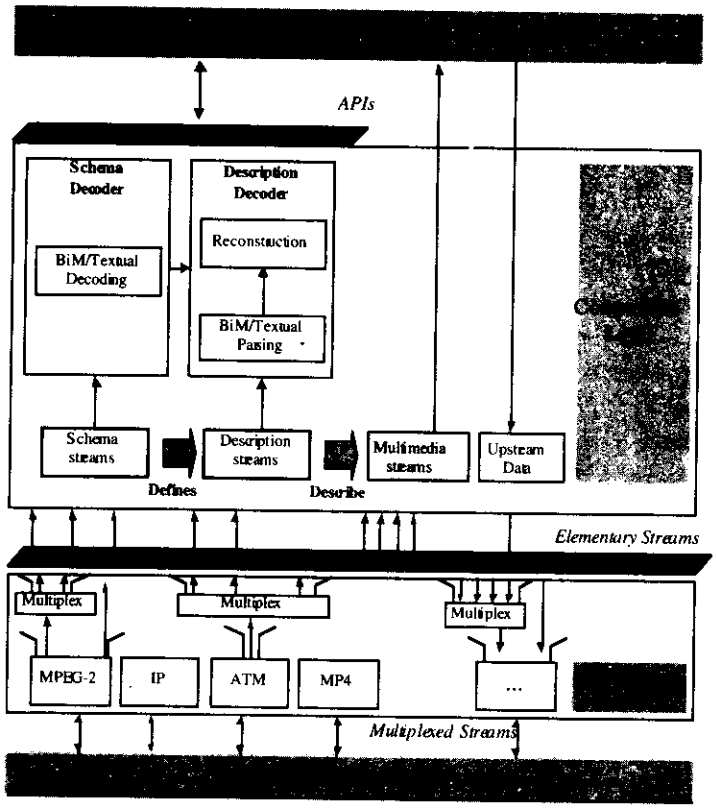


Figure 5.88 MPEG-7 architecture [5.156]. ©2000 ISO/IEC.

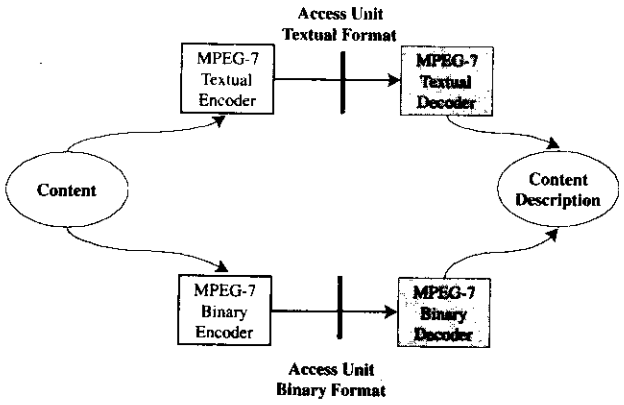


Figure 5.89 MPEG-7 normative interfaces [5.156]. ©2000 ISO/IEC.

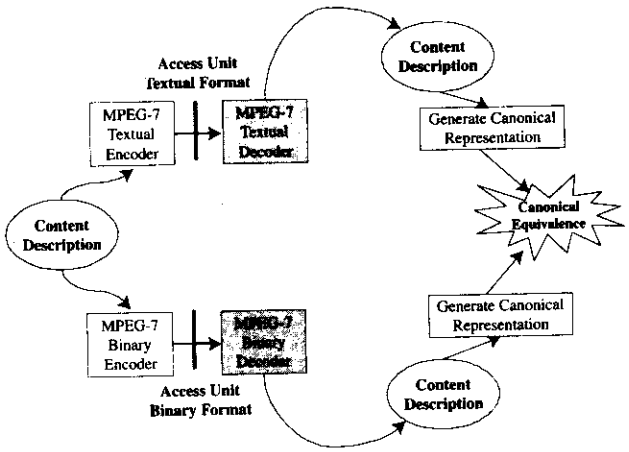


Figure 5.90 Validation process [5.156].©2000 ISO/IEC.

encoded descriptions are decoded with their respective binary and textual decoders. The two canonical descriptions will be equivalent.

MPEG-7 DDL

The main tools used to implement MPEG-7 descriptions are the DDL, DSs and Ds. Ds bind a feature to a set of values. DDs are models of the multimedia objects and of the data model of the description. They specify the types of the Ds that can be used in a given description and the relationships between these Ds or between other DSs. The DDL forms a core part of the MPEG-7 standard. It provides the solid descriptive foundation by which users can create their own DSs and D. The DDL defines the syntactic rules to express and combine DSs and Ds. The DDL is a schema language to represent the results of modeling audiovisual data, that is, DSs and Ds. The DDL must satisfy the MPEG-7 DDL requirements. It has to be able to express spatial, temporal, structural and conceptual relationships between the elements of Ds and DSs. It must provide a rich model for link and references between one or more Ds and the data that they describe. In addition, it must be platform and application independent as well as human and machine readable. The general consensus within MPEG-7 is that it should be based on Extensible Markup Language (XML) syntax. The XML schema language has been selected to provide the basis for the DDL [5.157]. As a consequence of this decision, the DDL can be broken down into the following logical normative components: the XML scheme structural language components and XML scheme data type language components.

MPEG-7 Audio

MPEG-7 Audio CD comprises the following technologies: the audio description framework, sound effect description tools, instrument timbre description tools, spoken content description, uniform silence segment and melodic Ds to facilitate query-by-humming. Four sets of audio

description tools that roughly represent application areas are integrated in the CD sound effects, musical instrument timbre, spoken content and melodic contour [5.157].

The sound effects Ds and DSs are a collection of tools for indexing and categorization of general sound effects. Support for automatic sound-effect identification and indexing is included, as well as tools for specifying taxonomy of sound classes and tools for specifying an ontology of sound recognizers. Such recognizers may be used to index and segment sound tracks automatically.

Timbre Ds aims to describe perceptual features of instrument sounds. Timbre is currently defined in the literature as the perceptual features that make two sounds having the same pitch and loudness sound different. The aim of the timbre DS is to describe these perceptual features with a reduced set of Ds. The Ds relate to notions such as *attack, brightness* or *richness* of sound.

The Spoken Content DS consists of combined word and phone lattices for each speaker in an audio stream. By combining the lattices, the problem of out-of-vocabulary words is greatly alleviated, and retrieval may still be carried out when the original decoding was in error. The DS can be used for two broad classes of retrieval scenarios: indexing into and retrieval of an audio stream and indexing of multimedia objects annotated with speech.

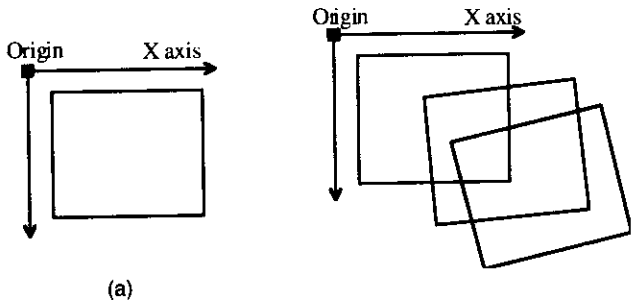
The Melody Contour DS is a compact representation for melodic information that allows for efficient and robust melodic similarity matching, for example, in query by humming. The Melody Contour DS uses a five-step contour (representing the interval difference between adjacent notes) in which intervals are quantized. The Melody Contour DS also represents basic rhythmic information by storing the number of the nearest whole beats of each note, which can dramatically increase the accuracy of matches to a query.

Several technologies combine to form the low-level audio D framework. One of the foundations is the scale tree, which allows (generally temporal) series of Ds to be represented in a scalable way. The low-level Ds to fit into this foundation include temporal envelope, spectral envelope, harmonicity, spectral centroid and fundamental frequency.

MPEG-7 Visual

MPEG-7 Visual description tools included in CD/XM consist of basic structures and Ds that cover color, texture, shape, motion, localization and other basic visual features. Each category consists of elementary and sophisticated Ds [5.157]. There are five visual-related basic structures: grid layout, time series, multiview, spatial 2D coordinates and temporal interpolation. The grid layout is a splitting of the image into a set of equally sized rectangular regions so that each region can be described separately. Each region of the grid can be described in terms of other Ds such as color texture. Furthermore, the D allows assignment of the subdescriptors to all rectangular areas, as well as to an arbitrary subset of rectangular regions.

The 2D/3D D specifies a structure that combines 2D Ds representing a visual feature of a 3D object seen from different view angles. The D forms a complete 3D view-based representation of the object. Any 2D visual D, such as, for example, contour shape, region shape, color or texture can be used. The 2D/3D D allows the matching of 3D objects by comparing their views, as well as comparing pure 2D views with 3D objects.



(a)

Figure 5.91 Local (a) and integrated coordinate (b) system [5.156]. ©2000 ISO/IEC.

A time series D defines a temporal series of Ds in a video segment and provides image-to-video frame matching and video-frames-to-video-frames matching functionalities. Two types of time series are available: regular time series and irregular time series. In the former, Ds locate regularly (with constant intervals) within a given time span. This enables a simple representation for the application that requires low complexity. On the other hand, Ds locate irregularly (with various intervals) within a given time span in the latter. This enables an efficient representation for the application that has the requirement of narrow transmission bandwidth or low storage capability. These are useful in particular to build Ds that contain the time series of Ds. The spatial 2D coordinates description defines a 2D spatial coordinate system to be used in other Ds/DSs when relevant. It supports two kinds of coordinate systems: local and integrated, as shown in Figure 5.91. In a local coordinate system, all images are mapped to the same position. In an integrated coordinate system, each image (frame) may be mapped to different areas. The integrated coordinate system can be used to represent coordinates as a mosaic of a video shot.

The temporal interoperation D describes a temporal interpolation using connected polynomials. This can be used to approximate multidimensional variable values that change with time, such as an object position in a video. The description size of the temporal interpolation is usually much smaller than describing all values. As an example, 25 real values are represented by 5 linear interpolation functions and 2 quadratic interpolation functions. Real data and interpolation functions are shown in Figure 5.92. The beginning of the temporal interpolation is always aligned to time 0.

The color Ds are color space, color quantization, dominant colors, scalable color, color-structure, color layout and group of frames/group of pictures color descriptor.

The feature is the color space that is to be used in other color-based Ds. In the current descriptions, the following color spaces are supported: RGB, YCrCb, Hue Saturation Value (HSV) and linear transformation matrix with reference to RGB and monochrome.

The color quantization D defines the quantization of a color space, and it supports uniform and nonuniform quantizers as well as lookup tables. Great flexibility is provided for a wide range of applications. For a meaningful application in the context of MPEG-7, this descriptor has to be combined with others to express the meaning of the values of a color histogram.

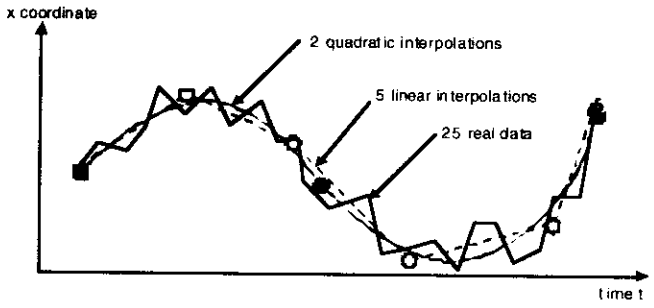


Figure 5.92 Real data and interpolation functions [5.157]. ©2000 ISO/IEC.

The dominant color D is most suitable for representing local (object or image region) features where a small number of colors are enough to characterize the color information in the region of interest. Whole images are also applicable (for example, flag images or color trademark images). Color quantization is used to extract a small number of representing colors in each region/image.

The scalable color D is a color histogram in HSV color space, which is encoded by a Haar transform. Its binary representation is scalable in terms of bit-representation accuracy across a broad range of data rates. The scalable color description is useful for image-to-image matching and retrieval based on a color feature. Retrieval accuracy increases with the number of bits used in the representation.

The color structure D is a color feature D that captures both color content (similar to a color histogram) and information about the structure of this content. Its main functionality is image-to-image matching, and its extended use is for still-image retrieval. The extraction methods embed color structure information into the descriptor by taking into account the colors in a local neighborhood of pixels instead of considering each pixel separately. The color structure D provides additional functionality and improved similarity-based image-retrieval performance for natural images compared to the ordinary color histogram.

The color layout D specifies the spatial distribution of colors for high-speed retrieval and browsing. It targets not only image-to-image matching, and video-clip-to-video-clip matching, but also layout-based retrieval for color, such as sketch-to-image matching which is not supported by other color Ds. This D can be applied either to a whole image or to any part of an image. This D can also be applied to arbitrarily shaped regions [5.160].

The group of frames/group of pictures color D extends the scalable color D that is defined for a still image to the color description of a video segment or a collection of still images.

There are three texture Ds: homogeneous texture, texture browsing and edge histogram. Homogeneous texture has emerged as an important visual primitive for searching and browsing through large collections of similar-looking patterns. An image can be considered as a mosaic of homogeneous textures so that these texture features are associated with the looking patterns. An image can be considered as a mosaic of homogeneous textures so that these texture features associated with the regions can be used to index the image data. The homogeneous texture D

provides a precise quantitative description of a texture that can be used for accurate search and retrieval in this respect.

The texture browsing D is useful for representing homogeneous texture for browsing type applications, and requires only 12 bits (maximum). It provides a perceptual characterization of texture, similar to a human characterization, in terms of regularity, coarseness and directionality. The computation of this D proceeds similarly to the homogeneous texture D. First, the image is filtered with a bank of orientation and scale-tuned filters (modeled using Gabor functions); from the filtered outputs, two dominant texture orientations are identified. Three bits are used to represent each of the dominant orientations. This is followed by analyzing the filtered image projections along the dominant orientations to determine the regularity (quantized to 2 bits) and coarseness (92 bits x 2). The second dominant orientation and second scale feature are optional. This D, combined with the homogeneous texture Descriptor, provides a scalable solution to representing homogeneous texture regions in images.

The edge histogram D represents the spatial distribution of five types of edges, namely, four directional edges and one nondirectional edge. Because edges play an important role for image perception, they can retrieve images with similar semantic meaning. Thus, it primarily targets image-to-image matching (by example or by sketch), especially for natural images with nonuniform edge distribution. In this context, the image-retrieval performance can be significantly improved if the edge histogram D is combined with other Ds such as the color histogram D. Besides having the best retrieval, performances considering this D alone are obtained by using the semiglobal and the global histograms generated directly from the edge histogram D as well as the local ones for the matching process.

There are four shape Ds: object region-based shape, contour-based shape, 3D shape and 2D-3D multiple view. The shape of an object may consist of either a single region or a set of regions as well as some holes in the object, as illustrated in Figure 5.93. Because the region-based shape D makes use of all pixels constituting the shape within a frame, it can describe any shape. This includes not only a simple shape with a single connected region as in Figure 5.93 (a) and (b), but also a complex shape that consists of holes in the object or several disjoint regions as illustrated in Figure 5.93 (c), (d) and (e), respectively. The region-based shape D not only can describe such diverse shapes efficiently in a single D, but can also describe robust to minor deformations along the boundary of the object. Figure 5.93 (g), (h) and (i) are very similar shaped images for a cup. The differences are at the handle. Shape (g) has a crack at the lower handle, and the handle in (i) is filled. The region-based shape D considers (g) and (h) similar, but different from (i) because the handle is filled. Figures 5.93 (j through l) show the part of video sequence where two disks are being separated. With the region-based D, they are considered similar. The descriptor is also characterized by its small size, fast extraction time and matching.

The contour-based shape D captures characteristic shape features of an object or region based on its contour. It uses so-called curvature scale-space representation, which captures perceptually meaningful features of the shape.

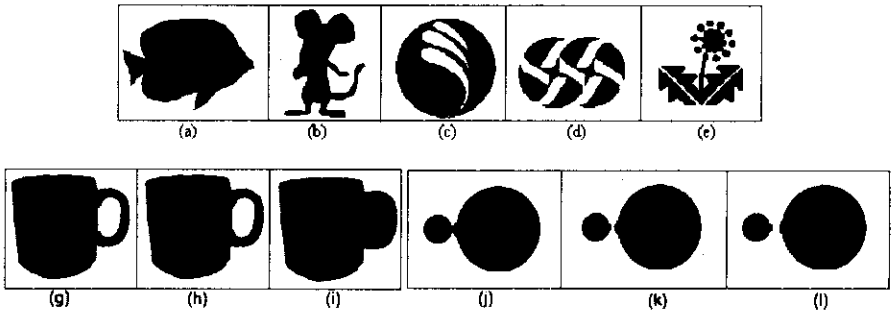


Figure 5.93 Examples of various shapes. The black pixel within the object corresponds to 1 in an image, and the white background corresponds to 0 [5.156]. ©2000 ISO/IEC.

The object-based shape D is based on the curvature scale-space representation of the contour. This representation has a number of important properties:

- It captures characteristic features of the shape very well, enabling similarity-based retrieval.
- It reflects properties of the perception of the human visual system and offers good generalization.
- It is robust to nonrigid motion.
- It is robust to partial occlusion of the shape.
- It is robust to perspective transformations, which result from the changes of the camera parameters and are common in images and video.

Considering the continuous development of multimedia technologies, virtual worlds and augmented reality, 3D contents have become a common feature of today's information systems. Most of the time, 3D information is represented as polygonal meshes. MPEG-4, within the SNHC subgroup, considered this issue and developed technologies for efficient 3D mesh-model coding. Within the framework of the MPEG-7 standard, tools for intelligent content-based access to 3D information are needed. The main MPEG-7 applications targeted here are search and retrieval and browsing of 3D model databases.

The proposed 3D shape D described in detail aims at providing an intrinsic shape description of 3D mesh models. It exploits some local attributes of the 3D surface.

There are four motion Ds: camera motion, object motion trajectory, parametric object motion and motion activity [5.161].

The camera motion D characterizes 3D camera motion parameters. It is based on 3D camera motion parameter information, which can be automatically extracted or generated by capture devices. The camera motion D supports the following well-known basic camera operations: fixed, panning (horizontal rotation), tracking (horizontal transverse movement, also called traveling in the film industry), tilting (vertical rotation), booming (vertical transverse movements),

zooming (change of the focal length), dollying (translation along the optical axis) and rolling (rotation around the optical axis).

The motion trajectory of an object is a simple, high-level feature defined as the localization, in time and space, of one representative point of this object. This D shows usefulness for content-based retrieval in object-oriented visual databases. It is also of help in more specific applications. In a given context with a prior knowledge, trajectory can enable many functionalities. In surveillance, alarms can be triggered if some object has a trajectory identified as dangerous (for example, passing through a forbidden area, being unusually quick, and so forth). In sports, specific actions (for example, tennis rallies taking place at the net) can be recognized. Such a description also allows enhancing data interactions/manipulations. Semiautomatic multimedia editing can be performed, or a trajectory can be stretched or shifted to adopt the object motion to any given sequence global context.

The camera motion D is essentially a list of keypoints (x,y,z,t) along with a set of optional interpolating functions that describe the path of the object between keypoints in terms of acceleration. The speed is implicitly known by the keypoints' specification. The keypoints are specified by their time instant and either their 2D or 3D Cartesian coordinates, depending on the intended application. The interpolating functions are defined for each component x(t), y(t) and z(t) independently.

Parametric motion models have been extensively used within various related image-processing and analysis areas, including motion-based segmentation and estimation, global motion estimation, and mosaic and object tracking. Parametric motion models have already been used in MPEG-4 for global motion estimation and compensation and sprite generation. Within the MPEG-7 framework, motion is a highly relevant feature related to the spatiotemporal structure of a video and concerning several MPEG-7 specific applications, such as storage and retrieval of video databases and hyperlinking purposes. Motion is also a crucial feature for some domain-specific applications that have already been considered within the MPEG-7 framework, such as language indexing. The basic underlying principle consists of describing the motion of objects in video sequences as a 2D parametric model. Specifically, affine models include translations, rotations, scaling and combinations of them, planar perspective models make it possible to take into account global deformations associated with perspective projections and quadratic models make it possible to describe more complex movements [5.162].

The parametric model is associated with arbitrary (foreground or background) objects, defined as regions (group of pixels) in the image over a specified time interval. In this way, the object motion is captured in a compact manner as a set of a few parameters. Such an approach leads to a very efficient description of several types of motions, including simple translations, rotation and zooming, or more complex motions, such as combinations of these elementary motions [5.163, 5.164].

Defining appropriate similarity measures between motion models is mandatory for effective motion-based retrieval. It is also necessary for supporting both level queries, useful in query-by-example scenarios, and high-level queries such as "search for object approaching the

camera," for "search for object describing a rotational motion," "search for object translating left," and so forth.

A human watching a video or animation sequence perceives it as being a slow sequence, fast-paced sequence, action sequence, and so forth. The activity D captures this intuitive notion of intensity of action or pace of action in a video segment. Examples of high activity include scenes such as goal scoring in a soccer match, scoring in basketball games, a high-speed car chase, and so forth. On the other hand, scenes such as news reader shot, an interview scene, a still shot, and so forth, are perceived as low action shots. Video content in general spans the gamut from high to low activity, so we need a D that enables us to express accurately the activity of a given video sequence/shot and to cover comprehensively the aforementioned gamut. The activity D is useful for applications such as video repurposing, surveillance, fast browsing, dynamic video summarization, content-based query, and so forth. For example, we could slow down the presentation frame rate if the activity D indicates high activity to make the high activity viewable. Another example of an application is finding all the high-action shots in a news video program example, which can be viewed both as browsing and abstraction.

As for localization, there exists the region locator and spatiotemporal locator. The region locator D enables localization of regions within images or frames by specifying them with a brief and scalable representation of a box or a polygon. On the other hand, the spatiotemporal locator describes spatiotemporal regions in a video sequence, such as moving-object regions, and provides localization functionality. An example of spatiotemporal regions is shown in Figure 5.94.

The main application for these locaters is hypermedia, which displays the related information when the designed point is inside the object. Another main application is object retrieval by checking whether the object has passed particular points. This can be used for surveillance. The spatiotemporal locator can describe both spatially connected and nonconnected regions.

Among others Ds, we will mention here the face recognition D [5.165]. It can be used to retrieve face images that match a query face image. The D represents the projection of a face vector onto a set of basis vectors that span the space of possible face vectors. The face recognition feature set is extracted from a normalized face image. This normalized face image contains 56 lines with 46 intensity values in each line. The center of two eyes in each face image are located on the 24^{th} row and the 16^{th} and 31^{st} column for the right and left eye, respectively. This normalized image is then used to extract the 1D face vector that consists of the luminance pixel values from the normalized face image arranged into a 1D vector, using a raster scan starting at the top-left corner of the image and finishing at the bottom-right corner of the image. The face

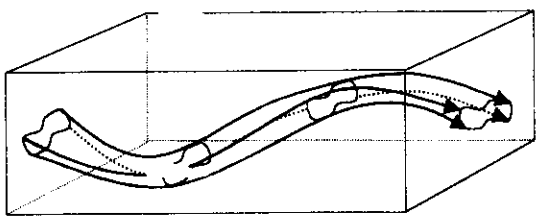


Figure 5.94 An example of a spatiotemporal region.

recognition feature set is then calculated by projecting the 1D face vector onto the space defined by a set of basis vectors.

MPEG-7 MMDSs

The main focus of the MMDS is to standardize a set of description tools (Ds and DSs) dealing with generic and multimedia entities. Generic entities are features that are used in audio, visual and text descriptions and are therefore generic to all media. These are, for example, vector, histogram, time and so forth. Apart from this set of generic description tools, more complex description tools are standardized. They are used whenever more than one medium needs to be described (audio and video). These description tools can be grouped into five different classes according to their functionalities:

- *Content description*—Representation of perceivable information
- *Content management*—Information about media features and the creation of the use of the audiovisual content
- *Content organization*—Representation of the analysis and classification of several audiovisual contents
- *Navigation and access*—Specification of summaries and variations of the audiovisual content
- *User interaction*—Description of user preferences pertaining to the consumption of the multimedia material

MPEG-7 provides DSs for content descriptions. These elements describe the structure (regions, video frames and audio segments) and semantics (objects, events and abstract notions). Structural aspects describe the audiovisual content from the viewpoint of its structure. The structure DSs are organized around segment DSs that represent the spatial, temporal or spatiotemporal structure of the audiovisual content. The segment DS can be organized into a hierarchical structure to produce a table of contents for accessing or an index for searching the audiovisual content. The segments can be further described on the basis of perceptual features using MPEG-7 Ds for color, texture, shape, motion, audio features and semantic information using textual annotations. Conceptual aspects describe the audiovisual content from the viewpoint of real-world semantics and conceptual notions. The semantic DSs involve entities, such as objects, events, abstract concepts and relationships. The structure DSs and semantic DSs are related by a set of links, which allows the audiovisual content to be described on the basis of both content structure and semantics together [5.157].

MPEG-7 also provides DSs for content management. Together, these elements describe different aspects of creation and production, media coding, storage and file formats and content usage. The functionality of each of these classes of DSs is given as creation information, usage information and media description. Creation information describes the creation and production of audiovisual content. The creation information describes the creation and classification of the audiovisual content and other material that is related to the audiovisual content. The creation

information provides a title, textual annotation and creation information, such as creators, creation locations and dates. The classification information describes how the audiovisual material is classified into categories, such as gear, subject, purpose, language and so forth. It also provides review and guidance information, such as age classification, subjective review, parental guidance and so forth. Finally, the related material information describes whether other audiovisual material exists that is related to the content being described. Usage information describes the usage information related to the audiovisual content, such as user rights, availability, usage record and financial information. The usage information is typically dynamic in that it is subject to change during the lifetime of the audiovisual content. A media description describes the storage media, such as the compression, coding and storage format of the audiovisual data. The media information DSs identify the master media, which is the original source from which different instances of the audiovisual content are produced. The instances of the audiovisual content are referred to as media profiles. Each media profile is described individually in terms of the encoding parameters, storage media information and location [5.157].

MPEG-7 provides encryption schemes for organizing and modeling collections of audiovisual content, segments, events and/or objects and for describing their common properties. The collections can be further described using different models and statistics in order to characterize the attributes of the collection members [5.157]. The collection structure DS describes collections of audiovisual content or pieces of audiovisual material such as temporal segments of video. The collection structure DS groups the audiovisual content, segments, events or objects into collection clusters and specifies properties that are common to the elements.

MPEG-7 provides DSs that facilitate navigation and access of audiovisual content by specifying summaries, views, partitions and variations of multimedia data. The MPEG-7 summary DSs provide summaries and abstracts of audiovisual content to enable efficient browsing and navigating of audiovisual data. The MPEG-7 space and frequency views provide views of the audiovisual data in the space or frequency domain, which allows multiresolution and progressive access. The MPEG-7 variation DSs specify the relation between different variations of audiovisual material, which allow adaptive selection of the different variations of the content under different terminal and delivery conditions.

Finally, the best set of MPEG-7 DSs deals with user interaction. The user preference information describes user frequencies pertaining to the consumption of the multimedia material. This allows, for example, matching between user preferences and MPEG-7 content descriptions in order to facilitate personalization of audiovisual content access, presentation and consumption. The user preference DS allows the specification of preferences for different types of content and modes of browsing, including context dependencies in terms of time and place. The user preference DS also allows weighting of the relative importance of different preferences. The user preference DS allows the specification of the privacy characteristics of the preferences and whether preferences are subject to update, such as by an agent who automatically learns through interaction with the user.

MPEG-7 Reference Software (XM)

The XM software is the simulation platform for the MPEG-7 Ds, DSs, Coding Schemes (CSs), and DDL. Besides the normative components, the simulation platform also needs some nonnormative components, essentially to execute some procedural code to be executed on the data structure. The data structures and the procedural code together form the applications. The XM *applications* are divided into two types: the server applications and the client applications. The server applications are used to extract the D data from the media data. The extracted D data is coded and written to an MPEG-7 bit stream. To create a server application, the media data has to be specified. This is done by a database file containing all the names of media files for which the DS and DSs should be extracted and stored. The processing of the database file, containing the input file information is performed by the general XM-component/class Image IO. The Image IO class also includes the loading of media data. The loaded media data is stored in an object of a media class. From the media data, the D data is extracted using an extraction class. First, the application has to give the references of the media and D data (addresses of the VOs) to the extraction object. Then, the extraction is performed. The next step is the coding of the D data into its binary representation. Analogous to the extraction, a coding scheme class is used. The output of the coding is put in a bit stream, which is implemented with the class Encoder File IO. The Encoder File IO class gets the bits from the coding scheme and writes them to the MPEG-7 database file [5.157].

The client application performs the search in the MPEG-7 coded database by computing the distance between the query D and all reference Ds of the database. Therefore, one D, the query D, is extracted in the same way as in the server application except that the coding is not performed. The reference Ds are all extracted from the MPEG-7 bit stream. The Decoder FileIO class is used to read the data from the bit-stream file. Each decoded D is stored in a D class object of the array containing all Ds of the database. Before decoding, the references of the Decoder FileIO object and of the D object have to be given to the coding scheme (which includes the encoder and the decoder). Now the query D and the reference Ds are available. The matching is performed by the search object of the search class with one reference descriptor after the other. Prior to the matching, the addresses of both Ds have to be given to the search object. The matching returns the distance between both Ds. For easier processing of the results, the number of the reference D and its distance to the query D are given to an object of the MatchList class. This class stores the numbers and the distances of the best matches. At the end of the client application, the n best matches are printed at the output. The filename of a specific reference D can be computed by its number and the database file containing all the media filenames, which was used for the extraction of the database in the server application.

MPEG-7 Conformance

MPEG-7 conformance includes the guidelines and procedures for testing conformance of MPEG-7 implementations.

5.7.4 Applications Enabled by MPEG-7

Many applications, services and domains will benefit from the MPEG-7 standard. One of the first efforts in the development of the standard was collecting requirements from an application point of view. MPEG-7 has helped the requirements process to make the distinction between push and pull applications. Pull roughly refers to interactive queries, and push denotes the situation in which information is available in streamed form, such as in broadcast [5.166].

Currently, the MPEG-7 Applications document distinguishes between these push and pull applications and another category, the so-called specialized professional and control applications, such as biomedical and remote sensing. Some examples of applications are as follows [5.166, 5.167]:

- *Digital libraries*—Many examples fall under this category. They include video libraries, image catalogs, musical dictionaries, future home multimedia databases, and so forth.
- *Multimedia directory services*—An example is The Yellow Pages.
- *Broadcast media selection*—This includes radio channel, TV channel and Internet broadcast search and selection.
- *Multimedia editing*—Personalized electronic news services and media authoring are examples.
- *Universal access to multimedia content*—This is allowing content to scale to access conditions and devices in an intelligent way.
- *Automated processing of multimedia information*—This is an automated analysis of the output from a surveillance camera, where this output has already been segmented and where MPEG-7 descriptions of the objects have been generated.
- The potential applications are spread across the following application domains:
- Film, video and radio archives
- Professional editing and journalism (for example, searching speeches of a certain politician using his name, his voice or his face; real-time markup of incoming programs)
- Education and cultural services (for example, history museums, art galleries, and so forth)
- Tourist information
- Entertainment (for example, searching for a game or karaoke)
- Investigation services (for example, human characteristics recognition and forensics)
- Computer vision and information systems
- Geographical information systems, remote sensing (for example, cartography, ecology, natural resource management, and so forth) and surveillance (for example, traffic control, surface transportation, nondestructive testing in hostile environments, and so forth)
- Biomedical applications

- Shopping (for example, searching for clothing and fashions)
- Architecture, real estate and interior design
- Social (for example, dating services)

These lists are not exhaustive, but they are considered to provide a set of representative requirements for the standard so that other applications are also enabled. New applications are being added to the list as the work continues. An example of an application that has recently been given much attention is that of universal access to multimedia content.

5.8 MPEG-21 Multimedia Framework

The aims of starting MPEG-21 are the following:

- To understand if and how various components fit together
- To discuss which new standards may be required, if gaps in the infrastructure exist and when the above two points have been reached.
- To accomplish the integration of different standards

The digital marketplace, which is founded upon ubiquitous international communication networks such as the Internet, rewrites existing business models for trading physical goods with new models for distributing and trading digital content electronically. In this new marketplace, it is becoming increasingly difficult to separate the different IPRs that are associated with multimedia content.

The latest MPEG project, MPEG-21 Multimedia Framework, has been started with the goal to enable transparent and augmented use of multimedia resources across a wide range of networks and devices.

The basic elements of the framework are the following:

- Digital Items, which are structured digital objects with a standard representation, identification and metadata within the MPEG-21 framework
- Users of all entities that interact in the MPEG-21 environment or make use of MPEG-21 digital items

A digital item is a very broad concept. Let us see how this applies to music compilation. This digital item may be composed of music files or streams, associated photos, videos, animation graphics, lyrics, scores and MIDI files, but could also contain interviews with singers, news related to the song, statements by opinion makers, ratings of agencies, positions in the hit list and so forth. Most important, it could contain navigational information driven by user preferences and, possibly, bargains related to each of these elements.

The meaning of a user in MPEG-21 is very broad and is by no means restricted to the end-user. Therefore, an MPEG-21 user can be anybody who creates content, provides content, archives content, rates content, enhances or delivers content, aggregates content, syndicates con-

tent, sells content to end-users, consumes content, subscribes to content, regulates content or facilitates or regulates transactions that occur from any of the previous examples.

The work carried out so far has identified seven technologies that are needed to achieve the MPEG-21 goals. They include the following [5.168]:

- *Digital item declaration*—A uniform and flexible abstraction and interoperable schema for declaring digital items
- *Content representation*—How the data is represented as different media
- *Digital item identification and description*—A framework for identification and description of any entity regardless of its nature, type or granularity
- *Content management and usage*—The provision of interfaces and protocols that enable creation, manipulation, search, access, storage, delivery and (re)use of content across the content distribution and consumption value chain
- *Intellectual property management and protection*—The means to enable content to be persistently and reliably managed and protected across a wide range of networks and devices
- *Terminals and networks*—The ability to provide interoperable and transparent access to content across networks and terminal installations
- *Event reporting*—The metrics and interfaces that enable users to understand precisely the performance of all reportable events within the framework

The relationship of the seven technologies in the framework is depicted in Figure 5.95.

To carry out the necessary tasks, MPEG has identified the following method of work. First, it is necessary to define a framework supporting the vision. This is being done by drafting a TR that describes the complete scope of the multimedia framework and that identifies the critical technologies of the framework. The TR explains how the components of the framework are related and identifies the goals that are not currently filled by existing standardized technologies. The next step is the involvement of other relevant bodies in this effort. This is necessary because some technologies needed for MPEG-21 are not MPEG specific and are better dealt with by other bodies. For each of the technologies that is not yet available, MPEG will do one of the following:

- Develop them if MPEG has the necessary expertise.
- Otherwise, engage other bodies to achieve their development.

Last, the actual integration of the technologies has been performed. The TR was approved in December 2001.

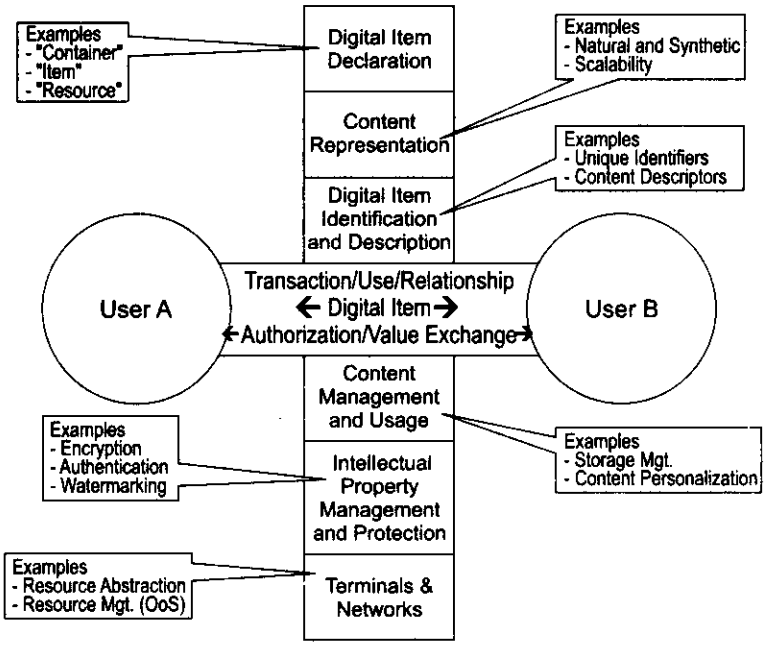


Figure 5.95 MPEG-21 multimedia framework [5.168]. ©2001 ISO/IEC.

5.8.1 Audiovisual Content Representation Issues

MPEG-21 intends to address the traditional audiovisual content representation issues, but with one major difference. Content can no longer be seen as essence (what the user has traditionally consumed) or as metadata (the description of the essence), but as an integrated whole.

A second area of the MPEG-21 standard is explained by the consideration that the way the user interacts with this complex world is difficult to separate from the way the user acquires the right to access the content. MPEG-21 will therefore also identify the interfaces with content access in such a way that content protection—a necessity for the holders of the rights in order to retain control of their assets—is transparent to the end-user.

A third area of the MPEG-21 standard is caused by the pace of advances in content digitization that makes more urgent than ever the need to define a more comprehensive way to identify and describe content as compared to the simple solutions provided by MPEG-2 and MPEG-4. The problem is twofold. There is a need to define content identification and description at the semantic level (what the identification and description code means), but also the way this information is carried by the content itself needs to be defined. The obvious requirement is that content identification and description should be carried out in such a way that it cannot be tampered with. Here we are talking about such technologies as watermarking or fingerprinting techniques.

MPEG-21 presents itself as the enabler of the world of multimedia that has been in the making for some years and to which several bodies, MPEG not being the last, have provided some basic technology elements.

This richness of solutions is part of the problem. In fact, there is no big picture to describe how these elements, either in existence or under development, relate to each other. The first goal of MPEG-21 is then to understand if and how these various components fit together and to see which new standards may be required. This material is in a TR.

MPEG-21 will contain other parts that will be normative. These will address the technologies that are needed in an electronic-trading environment founded upon ubiquitous networks that are encouraging new business models for trading digital content. The digital nature of content will make the distinction between content types less clear as their integration in new products and services makes the traditional boundaries between the different industries less distinct. What will remain unchanged is the value, both commercial and intrinsic, of the digital asset resources, and what will be enhanced are the new possibilities presented by the tools that enable players on the delivery chain to create, collect, package, distribute, store, access, consume, use and reuse content. The underpinning vision is one of a multimedia framework that supports transactions that are interoperable and highly automated to support these new types of commerce. A slogan for this could be "six billion creators, value addresses, publishers, retailers, consumers and resellers all seamlessly connected by networks and all operating through interoperable standards."

5.8.2 Description of a Multimedia Framework Architecture

As previously stated, the functionalities of a multimedia framework architecture have been grouped into architectural elements, even though some overlap exists between these and the purposes of standardization. At first, user requirements are formulated that are relevant for all seven architectural elements:

- To satisfy the experience of all types of users in the multimedia framework through the extension of all existing members of the value chain (creators, rights holders, distributors and consumers of digital items).
- To ensure that the increasing sophistication of technological solutions does not undermine the user experience.
- To achieve interoperability of systems through the integration of the components of the multimedia framework. A component is the binding of a resource to all of its relevant descriptors.
- To provide the means to protect the intellectual properties of all categories of users.
- To ensure that the privacy of users will be respected.

MPEG-21 Digital Item Declaration

The goal is to establish a uniform and flexible abstraction and interoperable schema for defining digital items. A digital item is a structured digital object with a standard representation, identifi-

cation and description within the MPEG-21 framework. This entity is also the fundamental unit of distribution and transaction within the MPEG-21 framework.

Content Representation

MPEG-21 provides content representation technology to represent efficiently any content of all the relevant data types of natural and synthetic origin, or any combination thereof, in a scalable and error-resilient way. The various elements in a multimedia scene will be independently accessible, synchronizable and multiplexed and will allow various types of interaction.

Digital Item Identification and Description

By digital item identification, we mean a token that is uniquely designated and enables the recognition of a digital item, its organization and attributes. The information used to describe a digital item is called a digital item description. The framework for the identification and description is interoperable and integrated to provide the following:

- Accuracy, reliability and uniqueness of identification
- Seamless identification of any entity regardless of its nature or type of granularity
- Persistent and efficient methods for the association of identifiers with digital items
- Security and integrity of identification and description that survive all kinds of manipulations and alterations
- Automated processing of rights transactions and content location, retrieval and acquisition

Content Management and Usage

The MPEG-21 multimedia framework should provide interfaces and protocols that enable creation, manipulation, search, access, storage, delivery and (re)use of content (which can be any media data and descriptive data) across the content distribution and consumption value chain, with emphasis on improving the interaction model for users with personalization and content management. The previous should be supported both when the user is performing these functions and when the functions are delegated to nonhuman entities (such as agents). In this context, content management should not be understood as managing the rights of the content.

Intellectual Property Management and Protection

The MPEG-21 multimedia framework should provide a multimedia digital rights management framework that does the following:

- Enables all users to express their rights and interests in, and agreements related to, digital items and to have assurance that those rights, interests and agreements will be persistently and reliably managed and protected across a wide range of networks and devices.
- Enables, to the extent possible, the capture, codification, dissemination and reflection of updates of relevant legislation, regulations, agreements and cultural norms that

together create the setting and generally accepted societal platform for commerce involving digital items.

- Provides, to the extent possible, a uniform technical and organizational foundation for domain governance organizations that govern (on behalf of all users of digital items) the behavior of devices, systems and applications involved in interacting with digital items and services that provide transactional support within the MPEG-21 framework.

Terminals and Networks

With terminals and networks, we achieve interoperable transport access to distributed advanced multimedia content by scheduling users from network and terminal installation, management and implementation issues. This will enable the provision of network and terminal resources on demand to form user communities where multimedia content can be created and shared. This is always with the agreed/contracted quality, reliability and flexibility, allowing the multimedia applications to connect arbitrary sets of users such that the quality of the user experience will be guaranteed.

This implies the following at a minimum:

- Networks should provide content-transport functions according to a QoS contract established between the user and the network.
- Terminals and networks should provide scalable execution functions as requested by content.
- Access to network and terminal resources will happen through interfaces.

Event Reporting

MPEG-21 should provide metrics and interfaces that enable users to understand precisely the performance of all reportable events within the framework. Such event reporting then provides users a means of acting on specific interactions, as well as enabling a vast set of out-of-scope processes, frameworks and models to interoperate with MPEG-21. Event reporting creates a standardized set of metrics and interfaces with which to describe the temporally unique events and interactions within MPEG-21.

5.8.3 Requirements for Digital Item Declaration

Within any system (such as MPEG-21) that proposes to facilitate a wide range of actions involving digital items, there is a strong need for a concrete representation of any individual digital item. Clearly, there are many kinds of content, and probably just as many possible ways of representing it. This presents a strong challenge to design a powerful and flexible model for digital items that accommodate the many types of content, as well as any and all new forms that the content may assume in the future. Such a model is truly useful only if it yields a schema that can be used to represent unambiguously and to communicate interoperably about any digital items defined within the model.

The framework must support the following global requirements for a digital item declaration [5.169]:

- Hierarchies of containers and digital items must be capable of being efficiently searched and traversed. A container represents a potentially hierarchical structure that allows items to be grouped.
- Media resources and the digital item declaration are fully separable, meaning that the representation of an item can be communicated and processed in the absence of local copies of the associated media resources.
- The framework must enable the robust processing, validation and manipulation of digital items and containers.
- Identification and revision management of elements in a digital item or container must be supportable in an open and extensible manner.
- Digital items and containers may contain individual elements that are associated with multiple locations within the definition/hierarchy.

The framework must support a definition of containers that does the following:

- Support the unrestricted construction and description of hierarchical groupings of digital items.
- Support the metaphor of shelves for the organization and management of collections of digital items.
- Support the metaphor of packages for the transfer and delivery of digital items.

The framework must support a definition of digital items that supports the construction of compilations of items that fully preserves the structure and properties of the subitems.

The framework must support definitions of components that do the following:

- Associate a media resource of any type or format with an item.
- Provide an externally identifiable target for links from a media resource.

The framework must support a definition of descriptors that does the following:

- Associate a component or statement of any description scheme with any element. A statement is a literal textual value that contains information, but not an asset.
- Is capable of being described by other descriptors.

Also, the framework must support the configuration and atomization of digital items that enable a flexible mechanism for defining decision trees.

5.9 ITU-T Standardization of Audiovisual Communication Systems

The ITU is involved in the preparation of standards for the Global Information Infrastructure (GII). It is a focal point of the converging technologies and industries and will provide various multimedia applications. When considering the transition from national to GII, it is natural that the attention of those involved with telecommunications, such as entrepreneurs, service providers and regulators, should be focused on questions of technology, market opportunity, industry structure, investment risks and potential returns. Global interconnectivity and interoperability are some of the key issues of GII and require global standards. There have been many activities on GII-related standardization since the GII was first advocated in 1994 [5.170]. The standardization activity on GII in ITU-T was initiated by ITU-T SG13 in July 1995. Focusing on standardization, the ITU can be seen as a GII facilitator by providing global interconnectivity and interoperability through its standards. It is generally accepted that the essential global standards must address market needs, must not impair or restrict the creativity of equipment manufacturers, information providers and service providers and must provide a realistic and stable base for the envisioned information infrastructure. Global specifications are universally seen as necessary for a timely, successful GII. Such standards can achieve applicability and interoperability and can meet the market requirements for cost effectiveness, QoS, and support for cultural diversity. For the development of GII, the ITU is ideally placed to be an integrator, linking nonindustrial and developed countries.

5.9.1 ITU-T Standardization Process

Without a common language that both the transmitter and the receiver understand, communication is impossible. For multimedia communication that involves transmission of video data, standards play an even more important role. A video coding standard not only has to specify a common language, formally known as the bit-stream syntax, but the language also has to be efficient. Efficiency has two aspects. One is that the standard has to support a good compression algorithm that brings down the bandwidth requirement for transmitting the video data. The other is that the standard has to allow efficient implementation of the encoder and the decoder, that is, the complexity of the compression algorithm has to be as low as possible. Suppose some multimedia content needs to be transmitted from a source to a destination. The success of the communication is mainly determined by whether the source and the destination understand the same language. Adoption of standards by equipment manufacturers and service providers results in higher volume and hence lowers the cost. In addition, it offers consumers more freedom of choice among manufacturers and therefore is highly welcomed by the consumers.

There are two major types of standards. Industrial or commercial standards are mainly defined by mutual agreement among a number of companies. Sometimes these standards can become very popular in the market, become the de facto standards and are widely accepted by the other companies. The other type is called a voluntary standard that is defined by volunteers in open committees. The agreement on these standards has to be based on the consensus of all

committee members. These standards are usually driven by the market needs. At the same time, they need to stay ahead of the development of technologies.

For multimedia communication, there are two major standard organizations: ITU-T, and the ISO. For example, recent video-coding standards defined by these two organizations are summarized in Table 5.23. These standards differ mainly in the operating bit rates due to the applications that they were originally designed for, but all standards can essentially be used for all applications at a wide range of bit rates. In terms of coding algorithms, all standards follow a similar framework.

Table 5.23 Video-coding standards.

Standards organization	Video-coding standard	Typical range of bit rates	Typical applications
ITU-T	H.261	px64 Kb/s, p=1,2,...,30	ISDN video phone
ISO	IS 11172-2 MPEG-1 Video	1.2 Mb/s	CD-ROM
ISO	IS 13818-2 MPEG-2 Video	4-80 Mb/s	SDTV, HDTV
ITU-T	H.263	64 Kb/s or below	PSTN video phone
ISO	CD 14496-2 MPEG-4 Video	24-1024 Kb/s	Interactive audio/video
ITU-T	H.263 Version 2	< 64 Kb/s	PSTN video phone
ITU-T	H.26L	< 64 Kb/s	Network-friendly packet-based video

In the past, most video compression and coding standards were developed with a specific application and networking infrastructure in mind. For example, ITU-T recommendation H.261 was optimized for use with interactive audiovisual communication equipment, for example, a videophone [5.171], and in conjunction with the H.320 series of recommendations as multiplex and control protocols on top of ISDN [5.173]. Consequently, the H.261 designers made various design choices that limit the applicability of H.261 to this particular environment. The original H.263 was developed for video-compression rates below 64 Kb/s. This was the first international standard for video compression that would permit video communications at such low rates [5.174, 5.175]. After H.263 was completed, it become apparent that incremental changes could be made to H.263 that would visibly improve its compression performance. It was thus decided in 1996 that a revision to H.263 would be created that incorporated these incremental improvements. ITU-T recommendation H.263 Version 2 (abbreviated as H.263+) is the very first interna-

tional standard in the area of video coding that is specifically designed to support the full range of both circuit-switched and packet-switched networks [5.176, 5.177, 5.178]. H.263+ contains functionalities that improve the quality of video transmission in error-prone environments and nonguaranteed QoS networks. H.26L is an ongoing standard activity that is searching for advanced coding techniques that can be fundamentally different from H.263.

A number of trends and directions have been identified that need to be taken into account by both the ITU-T and other standards development organizations when developing their programs, program priorities and liaison or partnership arrangements. Telecommunication networks are currently providing voice and data services worldwide with a high level of reliability and defined QoS and are based on different network technologies with interworking among them. Extension of the networks to include broadband capabilities is based on ATM technology. ATM is also being enhanced to provide not only for connection-oriented services, but also to meet requirements of connectionless network capabilities and services supported by these capabilities. Networks based on IPs provide a platform that allows users connected to different network infrastructures to have a common set of applications and to exchange data with an undefined QoS. The IP suite is evolving to include voice, data and video applications with defined QoS. Additionally, terrestrial radio, cable and satellite networks are providing local broadcast entertainment services and are also evolving to provide interactive voice, data and video services.

5.9.2 Audiovisual Systems (H.310, H.320, H.321, H.322, H.323, and H.324)

Audiovisual services provide real-time communication of speech together with visual information between two or more end-users. The visual information is typically moving pictures, but may be still pictures, graphics or any other form. The ITU-T Study Group 15 has been standardizing the audiovisual communication systems in various network environments. The first set of such standards, called *Recommendations* in the ITU-T, was formally established in December 1990 for narrow-band ISDN (N-ISDN), which provides digital channels of 64 Kb/s (B channel), 384 Kb/s (H0 channel), and 1,536/1,920 Kb/s (H11/H12 channel) [5.171]. Recommendation H.320 describes a total system stipulating several other recommendations to which respective constituent elements, such as audio coding, video coding, multimedia multiplexing and system control should conform. Its major target applications are video conferencing and videotelephony, but other applications are not excluded [5.173]. Since then, multipoint and security enhancements of N-ISDN systems have been developed [5.179]. In parallel with this, a new standardization activity was initiated in July 1990 towards broadband and high-quality audiovisual systems by forming an experts group. The first videoconferencing standard to emerge was H.320, which was defined for a circuit-switched narrow-band ISDN environment at bandwidths ranging from 64 Kb/s to more than 2 Mb/s. The ITU-T standards are also addressing videoconferencing in different network environments, such as H.324 for POTS [5.180], the H.323 LANs [5.181] and H.310 suite for ATM [5.182].

Table 5.24 shows the network environments for which audiovisual communication systems have been defined and gives the numbers of the ITU-T recommendations that specify these

Table 5.24 Audiovisual communication systems in various network environments [5.171].

Network	GSTN	N-ISDN	Guaranteed QoS LANs	Nonguaranteed QoS LANs	ATM (B-ISDN, ATM LANs)
Channel capacity	Up to 28.8 Kb/s	Up to 1,536 or 1,920 Kb/s	Up to 6/16 Mb/s	Up to 10/100 Mb/s	Up to 600 Mb/s
Characteristics	Ubiquitous	Circuit based Existing	Similar to N-ISDN	Packet-loss prone	Future basic network
Total system (date of first approval)	H.324 (03/96)	H.320 (12/90)	H.322 (03/96)	H.323 (11/96)	H.310 (11/96) H.321 (03/96)
Audio coding	G.723.1	G.711 G.722 G.728	G.711 G.722 G.728	G.711 G.722 G.723.1 G.728	G.711 G.722 G.728 ISO/IEC 11172-3
Video coding	H.261 H.263	H.261	H.261	H.261 H.263	H.261 H.262
Data	T.120, etc.	T.120, etc.	T.120, etc.	T.120, etc.	T.120, etc.
System control	H.245	H.242	H.242	H.245	H.242 (for H.321) H.245 (for native H.310)
Multimedia multiplex and synchronization	H.223	H.221	H.221	H.225.0 TCP/IP etc.	H.222.0 H.222.1
Call setup signaling	National standards	Q.931	Q.931	Q.931 H.225.0	Q.931

©1997 IEEE.

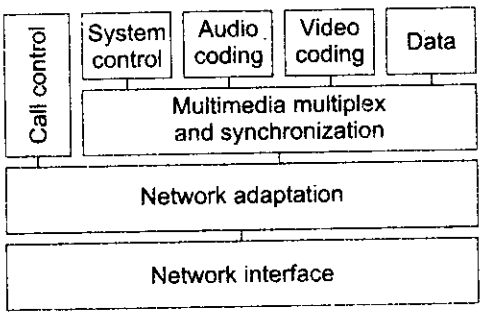


Figure 5.96 General protocol stack of H-series audiovisual communication terminal [5.171]. ©1997 IEEE.

systems, as well as their constituent elements. A general protocol stack model is shown in Figure 5.96. Note that system control and data may be directly on the top of network adaptation.

The General Switched Telephone Network (GSTN) system has been studied by a separate experts group for low bit-rate coding. It has produced not only the total system recommendation H.324 [5.180], but also the improved video-coding recommendation H.263 [5.175], the improved audio coding recommendation G.723.1 [5.183] and the multimedia multiplexing scheme defined in recommendation H.223 [5.184].

H.320 Standard

For all of the ITU standards, interoperability with the H.320 standard is mandatory. However, this interoperability is achieved through a gateway that, in some cases, must perform translations between different signaling protocols, different compression standards and different multiplexing schemes. The variations of signaling, compression and multiplexing for the various standards are due to the differing characteristics of the underlying networks to which each standard applies. Various aspects of the H.320 standard are clearly reusable in these network environments, but other aspects are not. For example, the audio- and video-compression algorithms or variations of them are being adopted by most of the new standards. However, different multiplexing schemes are being developed to better suit each network. One aspect of the H.320 standard is the centralized approach to multiparty conferencing. The H.320 standard defines a central conference server called a Multipoint Control Unit (MCU) to enable multiparty calls [5.185]. Each participant in the call contacts the MCU directly, which then controls the conference. This paradigm is clearly suited to the point-to-point connectivity nature of ISDN networks. Although not available everywhere, the ISDN is today's most widely used circuit-switched network for interactive multimedia communication. Video-telephone and videoconferencing systems based on the H.320 family of ITU-T recommendations are still the only affordable, medium- to high-quality videoconferencing solutions for most business users.

Standards for Audiovisual Services across ATM H.310 and H.321

The next generation public network is envisaged as, BISDN which is based on ATM [5.186]. Customer premises networks are also moving to ATM so that seamless network services can be provided [5.187]. ATM networks provide many opportunities for new and improved services, but they also pose new problems that must be solved before these services can be offered [5.6]. ATM network characteristics are summarized in Table 5.25.

The objective of standardizing audiovisual communication systems in ATM environments was to allow interoperability among different systems and interoperability with terminals connected to other networks while taking advantage of the opportunities and alleviating the limitations of ATM. In particular, it was an essential requirement that the new generation systems should interwork with the existing ones, that is, the ATM audiovisual systems should be able to interwork with H.320 systems situated in the N-ISDN. This could be achieved in many ways, including switching the elementary media coding to a common coding (H.261, G.711, and so forth) and using an intermediate gateway.

Table 5.25 ATM network characteristics.

Properties	Characteristics
Opportunities	Availability of high bandwidths Flexibility in bandwidth usage Variable bit-rate capability Service integration Use of Cell Loss Priority (CLP) Multipoint distribution in the network Flexibility in multimedia multiplexing or multiple connections
Limitations	Cell loss Cell delay variation (jitter) Packetization delay Usage parameter control (peak and/or average rates)

One of the greatest opportunities of BISDN is the high bandwidth available, which may be up to several hundred Mb/s compared to only 1.5 or 2 Mb/s for N-ISDN. Generally, higher bandwidth brings higher quality.

A video-coding standard, ITU-T recommendation H.262 has been established that gives pictures of broadcast television quality at around 5 to 10 Mb/s. High-quality stereo sound with subjective quality equal or close to that of a CD is obained at 384 Kb/s or lower bit rates by using MPEG-1 Audio [5.13]. Its extension to multichannel and lower sampling frequencies is also available as MPEG-2 Audio [5.16]. Hence, the ATM audiovisual systems should be able to realize high quality.

Another outstanding feature of the ATM network is its capability to achieve service integration. Cells and virtual channels can transport any type of information media after they are digitized and packetized. Different types of services can share the same network. This is seen by the user as an opportunity to access a number of different services through a single terminal. Hence, the ATM audiovisual communication systems should be able to cover as many applications as possible in a harmonized way. Possible applications are as follows:

- Conversational services (videoconferencing, videotelephony and distance learning)
- Distributive services (TV broadcasting and intracompany TV)
- Retrieval services (video on demand and network database)
- Messaging services (video mail)
- Video transmission (point-to-point transport of video programs)
- Video surveillance (road traffic monitoring)

The high bandwidth of the ATM network also provides a capability of low delay for conversational services. N-ISDN audiovisual systems use H.261 video coding, which incurs a buffering delay of at least four times the frame period (133 ms) plus any display delay due to picture skipping. The ATM audiovisual systems should significantly improve the end-to-end delay, so a

target of less than about 150 ms has been set. This value corresponds to the acceptable-for-most user-applications level of specification in ITU-T recommendation G.114 for one-way transmission time [5.188].

To meet the previous requirements, ITU-T SG15 has developed the following two recommendations for audiovisual communication systems in ATM environments:

- H.321—Adaptation of H.320 visual telephone terminals to BISDN environments [5.189]
- H.310—Broadband audiovisual communication systems and terminals [5.182]

Recommendation H.321 specifies the adaptation of H.320 visual telephone terminals to B-ISDN environments, thus satisfying the requirement that ATM terminals should interwork with those connected to N-ISDN. Recommendation H.310 includes the H.320/H.321 interoperation mode, but also defines a native mode, which takes advantage of the opportunities provided by ATM, to provide higher quality audiovisual communication systems. Although N-ISDN allows only a small number of transfer rates, quantized to being multiples of 64, 384, 1,536 and 1,920 Kb/s, BISDN allows a wide, almost infinite, range of transfer rates. This provides an obvious benefit of flexibility, but also causes a potential interoperability problem. It may happen that one terminal supports a group of transfer rates and another supports a different group of transfer rates with no value in common. Recommendation H.310 solves this problem by first defining the transfer rate to be a multiple of 64 Kb/s and then by mandating the two rates: 96x64=6,114 Kb/s and 144x64=9,216 Kb/s. Other optional transfer rates can be negotiated through the H.245 capability exchange procedures [5.190].

The two mandatory rates correspond to the H.262 Main profile at Main level medium-quality services and high-quality services, respectively [5.171].

Audiovisual communication requires the following phases to be completed before audiovisual communication can take place:

- Initial VC setup
- Capability exchange to identify the available common modes of operation
- Additional VC setup or bandwidth modification of the initial VC
- Logical channel establishment for audiovisual and data communication

Due to the wide flexibility of the ATM networks, the following two solutions have been considered for the basic model of H.310 (Figure 5.97):

- *Solution One*—A VC is initially used for only H.245 signaling, and audio is subsequently adjusted to accommodate audiovisual signals as well.
- *Solution Two*—The initial VC is used for H.245 signaling, after which another VC is set up for audiovisual signals.

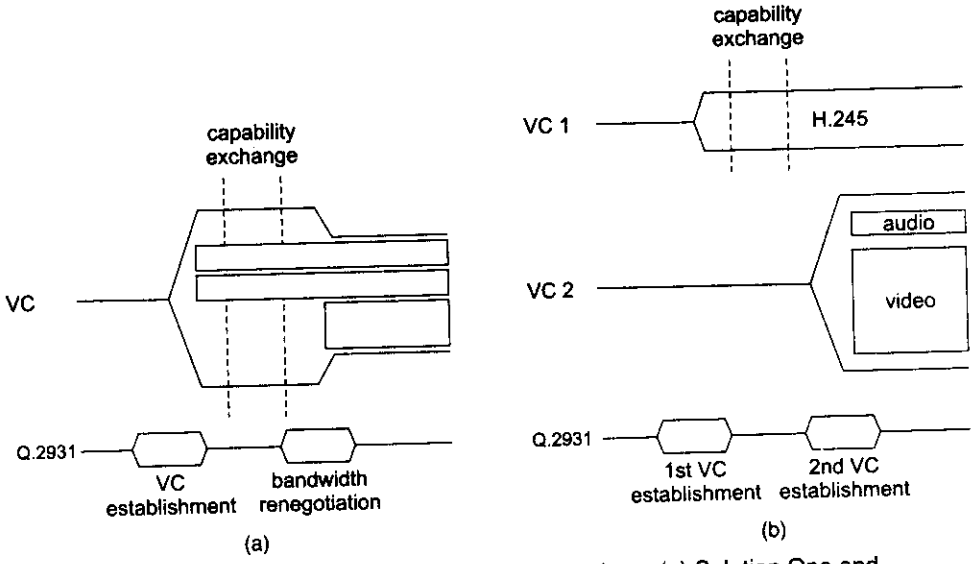


Figure 5.97 Two alternatives for H.310 start-up procedures (a) Solution One and (b) Solution Two [5.171]. ©1997 IEEE.

Solution One has two advantages. It provides audio immediately when the connection is made and may offer cheaper connection charges. However, it has many disadvantages. Two compatible terminals may not always connect at the first attempt if the calling terminal selects the wrong AAL, and the bandwidth may have to be renegotiated if the initial bandwidth is not equal to that actually required, which is a likely problem in asymmetric audiovisual channel configurations.

Solution Two, which avoids the disadvantages of Solution One by negotiating the capabilities of the second VC by running H.245 on the first VC, was adopted as the basic mode of operation of H.310. The initial VC is symmetrical with 64 Kb/s bandwidth and AAL5. Solution One may be added in the future if it is considered beneficial.

All H.310 terminals, when operating in the natural ATM mode, use the H.245 control protocol. Other audiovisual devices that use ATM transport may not use H.245.

Standard H.322–Guaranteed QoS LAN Systems

The proposal that the ITU should have a recommendation covering the provision for LANs and fc videotelephony and video conferencing facilities, equivalent to those specified by recommendation H.320 for N-ISDN, was made at the September 1993 meeting of the SG15 Experts Group for ATM Video Coding. At a subsequent meeting the Working Party mandated the Experts Group to begin studies and to produce a draft recommendation under the working number H.322. Even before the ITU-T had commenced its work on H.327, the Institute for Electrical and Electronic Engineers (IEEE) had been developing this new LAN standard. Originally, it was known as Iso-

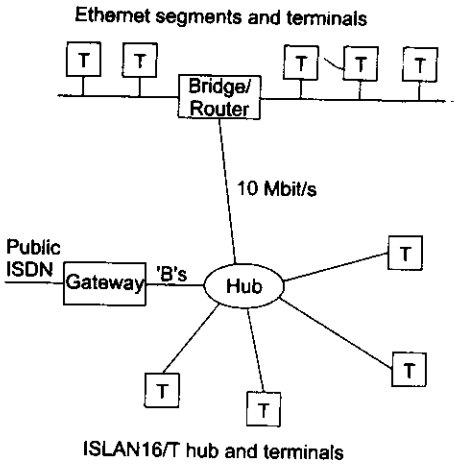


Figure 5.98 Typical configuration of H.322 using ISLAN 16-T [5.171]. ©1997 IEEE.

chronous Ethernet, frequently abbreviated to ISO-Enet, but was later renamed ISLAN 16-T [5.191]. Typical configuration of H.322 using ISLAN 16-T is shown in Figure 5.98.

ISLAN 16-T can be considered as an upgrade to the conventional 10 Mb/s Ethernet. It does require that terminals are star-wired to a central hub. However, the majority of recent Ethernet installations has this physical configuration even though the logical one is the traditional linear segment. As the 16 in its name infers, the bit rate is 16 Mb/s and, because each user has an individual link to the hub, all of this 16 Mb/s is available to that user and is not shared with others. It can be configured in two modes.

In the first mode, the 16 Mb/s is divided into a 10 Mb/s portion and a 6 Mb/s portion. The former is used with conventional Ethernet protocols and therefore gives complete compatibility with existing Ethernet software. The user can continue to access the file servers and to communicate with other Ethernet terminals as before. The remaining 6 Mb/s appears as 96 B channels, each using 64 Kb/s, plus one signaling channel.

In the second mode, the entire 16 Mb/s appears as B channels plus a signaling channel. Initially, this mode is unlikely to be used often because it offers no compatibility with Ethernet, and few applications need more than 6 Mb/s. However, this mode is well matched to H.262, which has a maximum rate of 15 Mb/s, and this may be important in the future.

A connection between the ISLAN 16-T hub and the existing Ethernet carries the 10 Mb/s traffic. To access a wider population of H.320 terminals, some B channels are connected between the new hub and a gateway to the ISDN.

The H.322 gateway unit need not carry out any function that is specific to the H.320 signals passing through it. Consequently, it is not restricted to serving only H.322 terminals on its LAN side, but can equally handle generic ISDN terminals. It is expected that the first H.322 gateways will have N-ISDN interfaces, but, as BISDN becomes widespread, interfaces to that will become more prevalent. Such a configuration will then permit good- to high-quality video at 2 Mb/s to be sent to and received from remote locations across the public BISDN. However,

H.322 does not mandate any minimum number of simultaneous calls that it can support or any minimum number of simultaneous calls to or from the ISDN.

To provide multipoint calls, an MCU can be added. Placing this at the gateway gives much flexibility over the balance between the number of participants on-LAN and off-LAN. Such an MCU can be of the switched type as exemplified by recommendation H.231 or the continuous presence type [5.192].

It is also possible to provide unidirectional audiovisual services to multiple recipients as in recommendation H.331 [5.193]. Such MCU and broadcast facilities may be provided by additional units separate from the gateway, or they may be combined into the gateway itself. H.322 mandates neither their presence nor the method of implementation.

ITU-T H.323 Standard

This recommendation specifies equipment and systems for visual telephony on nonguaranteed QoS LANs. It covers those situations where the transmission path includes one or more LANs, which may not provide a guaranteed QoS equivalent to that of N-ISDN. Examples of this type of LAN include Ethernet, Fast Ethernet, Token Ring [5.194] and Fiber Distributed Data Service (FDDS) [5.195].

The primary design considerations in the development of H.323 were the following:

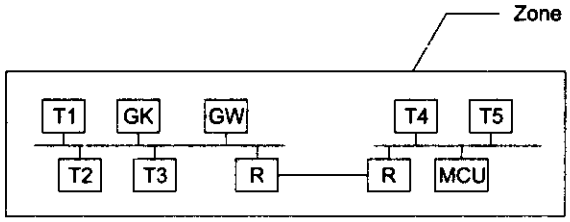
- Interoperability, especially with N-ISDN and H.320
- Control of access to the LAN to avoid congestion
- Multipoint call models
- Scalability from small- to medium-sized networks.

H.323 terminals may be used in multipoint configurations and may interwork with H.310 terminals on BISDN, H.320 terminals on N-ISDN, H.321 terminals on BISDN, H.322 terminals on guaranteed QoS LANs, H.324 terminals on GSTN and wireless networks and V.70 terminals on GSTN [5.196].

Provisions have been made to provide a gatekeeper that performs admission control for H.323 terminals within its zone that are attempting to gain access to the LAN. The criteria used by the gatekeeper to allow such access are nonstandardized. In addition, the gatekeeper can limit the bandwidth that a terminal uses and can control the call model used, which also affects bandwidth usage.

The zone of the H.323 system is presented in Figure 5.99. The scope of H.323 does not include the LAN itself or the transport layer that may be used to connect various LANs

Only elements needed for interaction with the Switched Circuit Network (SCN) are within the scope of H.323. The combination of the H.323 Gateway, the H.323 terminal and the out-of-scope LAN appears on the SCN as an H.320, H.310 or H.324 terminal. Recommendation H.323 describes the total system and its components, including terminals, gateways, gatekeepers, multipoint controllers, multipoint processors and MCUs. Recommendation H.225.0 describes the underlying protocols used for media packetization and control in the H.323 system [5.197].



GK: Gate Keeper
GW: Gate Way
MCU: Multipoint Control Unit
R: Router
T: Terminal

Figure 5.99 Zone of the H.323 system [5.171]. ©1997 IEEE.

A zone is the collection of all terminals, gateways, and multipoint control units (MCUs) managed by a single gatekeeper. A zone includes at least one terminal and may or may not include gateways or MCUs. A zone has one and only one gatekeeper. Multiple LAN segments that are connected using routers or other devices may be in the same zone.

Example 5.17 Figure 5.100 is an example of the H.323 terminal. It shows the user equipment interfaces, video codec, audio codec, telematic equipment, H.255.0 layer, system control functions and the interface to the LAN. All H.323 terminals have a system control unit, H.225.0 layer, network interface and an audio codec unit. The video codec unit and user data applications are optional.

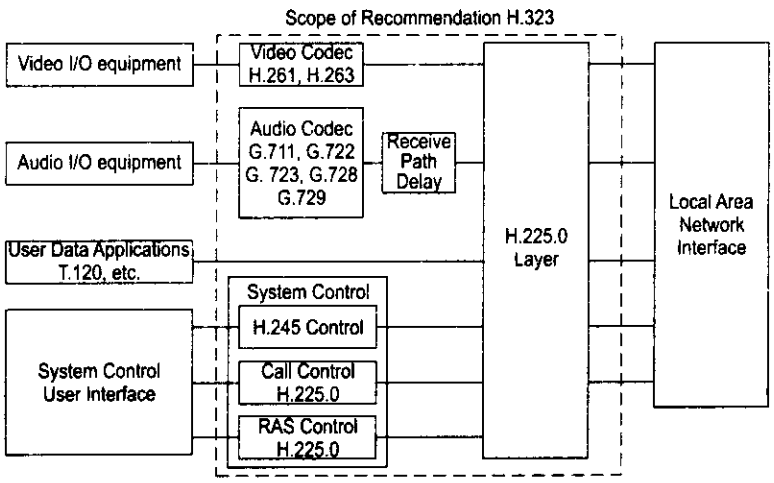


Figure 5.100 H.323 terminal equipment [5.171]. ©1997 IEEE.

H.324 Standard

If the telephony network is to be used for multimedia conferencing, most commercial products rely on the H.324 protocol hierarchy to ensure interoperability. The ITU-T recommendation H.324 entitled "Terminals for low bit rate multimedia communications" [5.180] provides an overview of PSTN multimedia terminals and references and all other ITU-T recommendations that are necessary to build such a terminal in a standard conformant way. For this reason, the standardization community refers to an H.324 family of recommendations.

H.324-based systems are free to negotiate protocol parameter values, such as packet size and allows round-trip delay. Such values, and especially the observed round-trip delay, substantially impact the performance of the error-resilience modes. To minimize packetization delay, packets with small sizes are usually negotiated. A typical payload size for Adaptation Layer (AL3) video packets in H.324 systems is 128 bytes. This is a number often used in various interoperability tests of commercial H.324 systems [5.198].

Unfortunately, many of today's H.324 systems have a significant end-to-end delay, due mainly to the integration of H.324 protocol mechanisms within a PC operating system environment, which is usually not optimized for real-time applications. The typical end-to-end delay for video in such a system can be well above 0.5 s. Therefore, AL3 retransmission should be avoided if interactive use, and thus a reasonable delay, is to be achieved. Because the special modem protocols used for H.324 do not perform any of their own error control or correction, significantly high bit-error rates can occur. Modems allow us, however, to optimize the trade-off between error rates and bit rates. System designers use this mechanism to gain optimal performance based on their design considerations.

5.9.3 Video-Coding Standards (H.261, H.263 and H.26L)

There are two approaches to understanding video-coding standards. One approach is to focus on the bit-stream syntax and to try to understand what each layer of the syntax represents and what each bit in the bit stream indicates. This approach is very important for manufacturers who need to build equipment that is compliant to the standard. The other approach is to focus on coding algorithms that can be used to generate standard-compliant bit streams and to try to understand that each component does not specify any encoding algorithms. The latter approach provides a better understanding of video-coding techniques as a whole, not just the standard bit-stream syntax.

H.261 Standard

This standard is defined by the ITU-T Study Group (SG) 15 for video telephony and video-conferencing applications [5.199]. After a reorganization within ITU-T in early 1997, SG 16 is the new group for video-coding standards. H.261 emphasizes low bit rates and the low coding delay. It was originated in 1984 and intended to be used for audiovisual services at bit rates around px384 Kb/s, where p is between 1 and 5. In 1988, the focus shifted, and it was decided to aim at bit rates around px64 Kb/s, where p is from 1 to 30. Therefore, H.261 also has an informal name

called px64. H.261 was approved in December 1990. The coding algorithm used in H.261 is basically a hybrid of motion compensation to remove temporal redundancy and transform coding to reduce spatial redundancy. Such a framework forms the basis of all video-coding standards that were developed later. Therefore, H.261 has very significant influence on many other existing and evolving video-encoding standards.

Digital video is composed of a sequence of pictures, or frames, that occur at a certain rate. For H.261, the frame rate is specified to be 30,000/1,001 (approximately 29.97) pictures per second. Each picture is composed of a number of samples. These samples are often referred to as pixels (picture elements) or pels. For a video-coding standard, it is important to understand the picture sizes that the standard applies to and the position of the samples. H.261 is designed to deal with two picture formats: the CIF and QCIF. In the still-image mode as defined in H.261, four times the currently transmitted video is used. For example, if the video format is CIF, the corresponding still-image format is 4 CIF. Table 5.26 summarizes a variety of picture formats supported by H.261 and H.263.

Table 5.26 A variety of picture formats supported by H.261 and H.263 [5.176].

Parameter	Sub-QCIF	QCIF	CIF	4CIF	16CIF
Number of pixels per line	128	176	352	704	1,408
Number of lines	96	144	288	576	1,152
Uncompressed bit rates	4.4 Mb/s	9.1 Mb/s	37 Mb/s	146 Mb/s	584 Mb/s

©1997 ITU-T.

H.261 is designed for videotelephony and videoconferencing, in which typical source material is composed of scenes of talking persons, so-called head and shoulder sequences, rather than general TV programs that contain a lot of motion and scene changes.

In H.261, each sample contains a luminance component called Y and two chrominance components called Cb and Cr. In particular, black is represented by Y=16, white is represented by Y=235, and the range of Cb and Cr is between 16 and 240, with 128 representing color difference (that is, gray). A particular format, as shown in Table 5.26, defines the size of the image, hence the resolution of the Y pels. The chrominance pels, however, typically have a lower resolution than the luminance pels in order to take advantage of the fact that human eyes are less sensitive to chrominance than luminance. In H.261, the Cb and Cr pels are specified to have half the resolution, both horizontally and vertically, of that of the Y pels. This is commonly referred to as 4:2:0 format. Each Cb or Cr pel lies in the center of four neighboring Y pels, as shown in Figure 5.101. Note that block edges lie in-between rows or columns of Y pels [5.199].

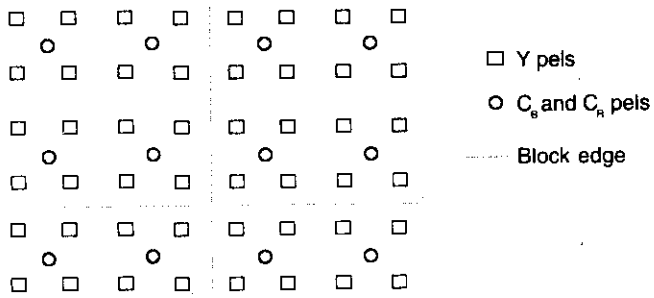


Figure 5.101 Positions of samples for H.261 [5.199]. ©1993 ITU-T.

Typically, we do not code an entire picture all at once. Instead, it is divided into blocks that are processed one by one both by the encoder and the decoder in a scan order as shown in Figure 5.102. This approach is often referred to as block-based coding.

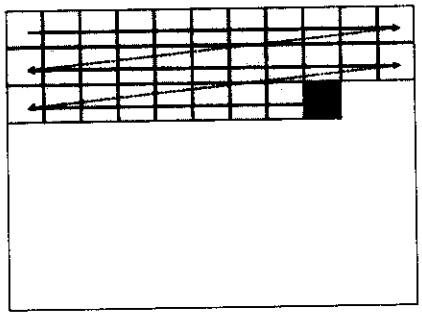


Figure 5.102 Illustration of block-based coding.

In H.261, a block is defined as a group of 8x8 pels. Because of the downsampling in the chrominance components as mentioned earlier, one block of Cb pels and one block of Cr pels correspond to four blocks of Y pels. The collection of these six blocks is called a Macroblock (MB), as shown in Figure 5.103 with the order of blocks marked as 1 to 6. An MB is treated as one unit in the coding process.

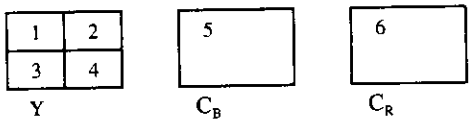


Figure 5.103 A macroblock [5.199]. ©1993 ITU-T.

A number of MBs are grouped together and called a Group of Blocks (GOB). For H.261, a GOB contains 33 MBs, as shown in Figure 5.104. The resulting structures for a picture, in the CIF case and the QCIF case, are shown in Figure 5.105 [5.199].

1	2	3	4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30	31	32	33

Figure 5.104 GOB for H.261 [5.199]. ©1993 ITU-T.

GOB 1	GOB 2
GOB 3	GOB 4
GOB 5	GOB 6
GOB 7	GOB 8
GOB 9	GOB 10
GOB 11	GOB 12

CIF

GOB 1
GOB 3
GOB 5

QCIF

Figure 5.105 GOB structures in CIF and QCIF case for H.261 [5.199]. ©1993 ITU-T.

Compression of video data typically is based on two principles: the reduction of spatial redundancy and the reduction of temporal redundancy. H.261 uses the DCT to remove spatial redundancy [5.200] and motion compensation to remove temporal redundancy [5.201].

The coding algorithm used in H.261 can be summarized into block diagrams in Figure 5.106 and Figure 5.107 [5.199]. At the encoder, the input picture is compared with the previously decoded frame with motion compensation. The difference signal is DCT transformed and quantized and then entropy-coded and transmitted. At the decoder, the decoded DCT coefficients

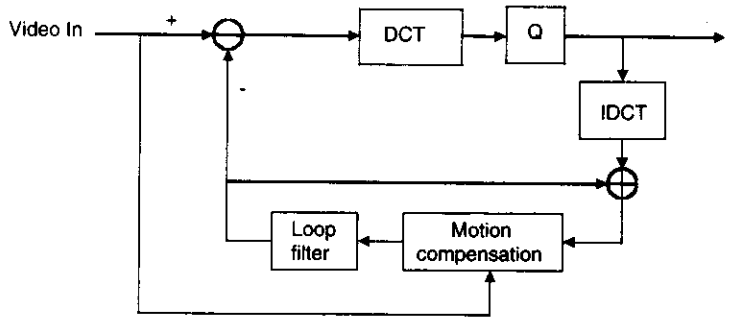


Figure 5.106 Block diagram of a video encoder used in H.261 [5.199]. ©1993 ITU-T.

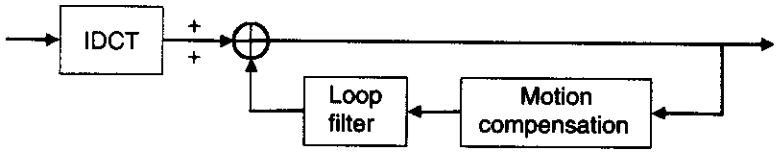


Figure 5.107 Block diagram of a video decoder used in H.261 [5.199]. ©1993 ITU-T.

are IDCT-transformed and then added to the previously decoded picture with motion compensation. Because the prediction of the current frame is composed of blocks at various locations in the reference frame, the prediction itself (or simply called the predicted frame) may contain coding noise and blocking artifacts. These artifacts may cause higher prediction errors. It is possible to reduce the prediction errors by passing the predicted frame through a low-pass filter before it is used as the prediction for the current frame. This filter is referred to as a loop filter (optional) because it operates inside the motion compensation loop.

As in all video-coding standards, H.261 specifies only the bit-stream syntax and how a decoder should interpret the bit stream to decode the image. Therefore, it specifies only the design of the decoder, not how the encoding should be done. For example, an encoder can simply decode to use only zero motion vectors and let transform coding take all of the burden of coding the residual. This may not be an efficient encoding algorithm, but it does generate a standard compliant bit stream.

H.261 has been included in several ITU-T H-series terminal standards for various network environments. One example is H.320 that is mainly designed for narrow-band ISDN terminals [5.173]. H.320 defines the systems and terminal equipment that use H.261 for video coding; H.221 for frame multiplexing; H.242 for signaling protocol [5.202] and G.711, G.722 and G.728 for audio coding. H.261 can also be used in other terminal standards including H.321, H.322 and H.324.

H.263 Standard

The activities of H.263 started around November 1993, and the standard was adopted in March 1996. The main goal of this endeavor was to design a video-coding standard suitable for applications with bit rates lower than 64 Kb/s (the so-called very-low bit-rate applications). For example, when sending video data across the PSTN and the mobile network, the video bit rates typically range from 10 to 24 Kb/s. During the development of H.263, it was identified that the near-term goal would be to enhance H.261 using the same general framework, and the long-term goal would be to design a video-coding standard that may be fundamentally different from H.261 in order to achieve further improvement in coding efficiency. As the standardization activities moved along, the near-term effort became H.263 and H.263 Version 2, and the long-term effort is now referred to as H.26L [5.174], previously called H.263L.

In essence, H.263 combines the features of H.261 together with MPEG and is optimized for very low bit rates. In terms of SNR, H.263 can provide a 3 to 4 dB gain over H.261 at bit rates below 64 Kb/s. In fact, H.263 provides superior coding efficiency to that of H.261 at all bit rates. When compared with MPEG-1, H.263 can give a 30% bit-rate saving.

Because H.263 was built on top of H.261, the main structures of the two standards are essentially the same. Therefore, we focus only on the differences between the two standards. The major differences include the following:

- H.263 supports more picture formats and uses a different GOB structure.

- H.263 uses half-pel motion compensation, but does not use loop filtering (optional) as in H.261.
- H.263 uses 3D VLC for coding of DCT coefficients.
- In addition to the basic coding algorithm, four options in H.263 that are negotiable between the encoder and the decoder provide improved performance.
- H.263 allows the quantization step size to change at each MB with less overhead.

In addition to CIF and QCIF as supported by H.261, H.263 also supports sub-QCIF, 4CIF and 16CIF (Table 5.26). Chrominance subsampling and the relative positions of chrominance pels are the same as those defined in H.261. However, H.263 uses different GOB structures (Figure 5.108). Unlike H.261, a GOB in H.263 always contains at least one full row of MBs.

A major difference between H.261 and H.263 is the half-pel prediction in the motion compensation. This concept is also used in MPEG. The motion vectors in H.261 can have only integer values, but H.263 allows the precision of motion vectors to be at a half of a pel. For example, it is possible to have a motion vector with values (4.5, -2.5). When a motion vector has noninteger values, bilinear interpolation is used to find the corresponding pel values for prediction.

The coding of motion vectors in H.263 is more sophisticated than that in H.261. The motion vectors of three neighboring MBs (the left, the above and the above right) as shown in

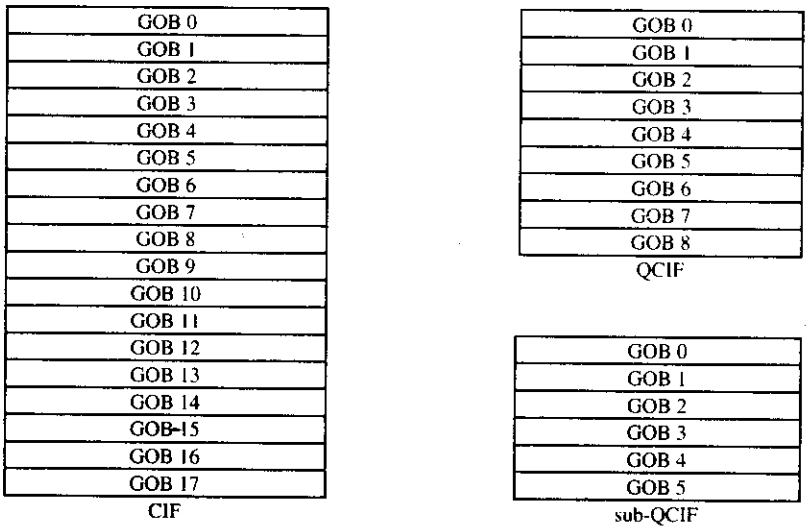
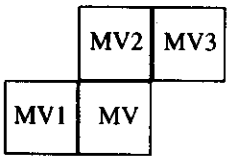


Figure 5.108 GOB structures for H.263 [5.174]. ©1996 ITU-T.



MV: Current motion vector
MV1, MV2, MV3: predictors
Prediction = median (MV1,MV2,MV3)

Figure 5.109 Prediction of motion vectors in H.263 [5.174]. ©1996 ITU-T.

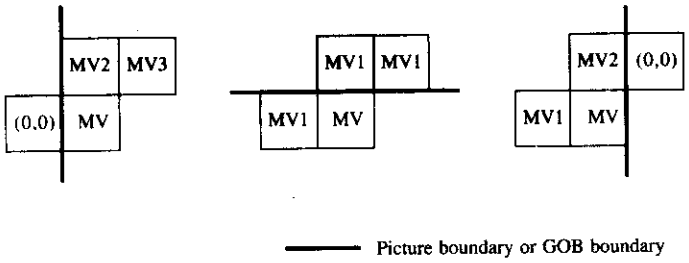


Figure 5.110 Motion vector prediction at picture/GOB boundaries in H.263 [5.174]. ©1996 ITU-T.

Figure 5.109 are used as predictors. The median of the three predictors is used as the prediction for the motion vector of the current block, and the prediction error is coded and transmitted. However, around a picture boundary or GOB boundary, special cases are needed (Figure 5.110). When only one neighboring MB is outside the picture boundary or GOB boundary, a zero motion vector is used to replace the motion vector of that MB as the predictor. When two neighboring MBs are outside, the motion vector of the only neighboring MB that is inside is used as the prediction.

H.263 specifies four options that are negotiable between the encoder and the decoder. At the beginning of each communication session, the decoder signals the encoder as to which of these options the decoder has the capability to decode. If the encoder also supports some of these options, it may enable those options. However, the encoder does not have to enable all the options that are supported by both the encoder and decoder. The four options in H.263 are the unrestricted motion vector mode, the syntax-based arithmetic-coding mode, the advanced prediction mode and the PB-frame mode.

In the unrestricted motion vector mode option, motion vectors are allowed to point outside of the picture boundary. In this case, edge pels are repeated to extend to the pels outside so that prediction can be done. A significant coding gain can be achieved with unrestricted motion vectors if there is movement around picture edges, especially for smaller picture formats like QCIF and sub-QCIF. In addition, this mode allows a wider range of motion vectors than H.261. Large motion vectors can be very effective when the motion in the scene is caused by heavy motion, for example, motion due to camera movement.

In syntax-based arithmetic coding, arithmetic coding is used, instead of VLC tables, for entropy coding. Under the same coding condition, using arithmetic coding will result in a bit stream different from the bit stream generated by using a VLC table, but the reconstructed frames and the SNR will be the same. Experiments show that the average bit-rate savings is about 3 to 4% for interframes and about 10% for intrablocks and frames [5.203].

In the advanced prediction mode, Overlapped Block Motion Compensation (OBMC) [5.204] is used to code the luminance of P-pictures, which typically results in less blocking artifacts. This mode also allows the encoder to assign four independent motion vectors to each MB, that is, each block in one MB can have an independent motion vector. In general, using four motion vectors gives better prediction since one motion vector is used to represent the move-

ment of an 8x8 block instead of a 16x16 MB. Of course, this implies more motion vectors and hence requires more bits to code the motion vectors. Therefore, the encoder has to decide when to use four motion vectors and when to use only one. Finally, in the advanced prediction mode, motion vectors are allowed to cross picture boundaries as is the case in the unrestricted motion vector mode.

In the PB-frame mode, a PB-frame consists of two pictures coded as one unit, as shown in Figure 5.111. The first picture, called the P-picture, is a picture predicted from the last decoded picture. The last decoded picture can be either an I-picture, a P-picture or the P-picture of a PB-frame. The second picture, called the B-picture (B for bidirectional), is a picture predicted from both the last decoded picture and the P-picture that is currently being decoded. As opposed to the B-frames used in MPEG, PB frames do not need separate bidirectional motion vectors. Instead, forward vectors for the P-picture are scaled and added to a small delta vector to obtain vectors for the B-picture. This results in less bit-rate overhead for the B-picture. For relatively simple sequences at low bit rates, the picture rate can be doubled with this mode with minimal increase in the bit rate. However, for sequences with heavy motion, PB-frames do not work as well as B-pictures. Also, note that the use of PB-frame mode increases the end-to-end delay, so it may not be suitable for two-way interactive communication.

As in H.261, H.263 can be used in several terminal standards for different network environments. One example is H.324 [5.180] that defines audiovisual terminals for the traditional PSTN. In H.324, a telephone terminal uses H.263 as the video codec, H.223 as the multiplexing protocol, H.245 as the control protocol [5.190], G.723.1 for speech coding at 5.3/6.3 Kb/s and V.34 for the modem interface. H.324 is sometimes used to refer to the whole set of standards. H.263 can also be used in other terminal standards, such as H.323, which is designed for LANs without guaranteed QoS.

The ITU-T H.261/H.263 video-compression standards were designed for real-time coding and decoding for videoconferencing across constant bit-rate connections. In addition to low-coding delay, these standards extend the notion of I and P frames to I and P blocks within a frame. A block is typically an 8x8 pixel region. These algorithms achieve smoother processing

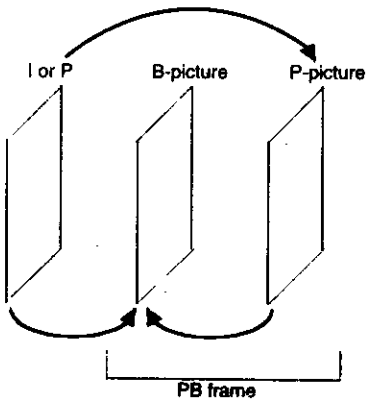


Figure 5.111
The PB-frame mode.

requirements and bandwidth profiles by essentially distributing an I-frame across several frames. I-block insertion is required to recover from the accumulation of DCT mismatch errors and for more efficient encoding when acute scene changes eliminate the advantage of difference encoding. For example, H.263 defines a maximum forced update period of 132 transmitted MB, within which each block must be updated at least once.

H.261/H.263 error resilience is achieved by inserting an I-block asynchronously and incrementally to refresh the image. This scheme, however, does not have regular synchronization points. I-block insertions are scattered across different frames. A block refreshed by an I-block insertion in the current frame might get corrupted again in the next frame if it is decoded using a corrupted motion vector reference. In fact, certain motion patterns could cause an error to propagate indefinitely [5.205].

As for the distribution interval, it can be reduced by shortering the forced updating period at the expense of a higher data rate or lower quality. This problem could be solved by retransmitting the lost data at the expense of increased delays.

H.263+ (H.263 Version 2) Standard

H.263+ is a revision of the original 1999 version of the H.263 ITU-T recommendation [5.176]. The original H.263 was developed for video compression at rates lower than 64 Kb/s and more specifically at rates lower than 33.4 Kb/s (V.34 modem). This was the first international standard for video compression that would permit video communications at such a low rate [5.174, 5.175]. H.263+ (often called H.263 Version 2) contains approximately 12 new features that do not exist in H.263. These include new coding modes that improve compression efficiency, support for scalable bit streams, several new features to support packet networks and error-prone environments, added functionality and support for a variety of video formats. Among the new features of H.263+, one of several that correct design inefficiencies of the original H.263 recommendation, is modified quantization mode. This mode has four key elements:

- Indication for larger quantizer changes from macroblock to macroblock to better react to rate-control requirements
- The ability to use a finer chrominance quantizer to better preserve chrominance fidelity The capability to support the entire range of quantized coefficient values rather than having to clip values greater than 128
- Explicitly restricting the representation of quantized transform coefficients to those that can reasonably occur

The second modification of the original H.263 is motion vector range. When H.263+ mode is invoked, the range is generally larger and depends on the frame size. Motion vector ranges in H.263+ are shown in Table 5.27.

Another modification to the original H.263 recommendation is the addition of a rounding term to the equation for half-pel interpolation. The rounding term toggles from frame to frame, thus eliminating this rounding bias and thereby reducing the artifact noticeably. Finally, H.263+

Table 5.27 Motion vector ranges in H.263+ [5.176].

Frame sizes up to	Motion vector range
352 x 288	[-32, 31.5]
704 x 576	[-64, 63.5]
1,408 x 1,152	[-128, 127.5]
Widths up to 2,048	Horizontal range [-256, 255.5]

©1997 ITU-T.

supports a wider variety of input video formats than H.263. In addition to five standard sizes, arbitrary frame sizes, in multiples of four from (32x32) to (2,048x1,152) can be supported.

One feature of H.263 Version 2 is that it extends the possible source formats specified in H.263. These extensions include the following:

Higher Picture Clock Frequency (PCF)—This allows picture clock rates higher than 30 frames per second. This feature helps to support additional camera and display technologies.

Custom picture formats—It is possible for the encoder and the decoder to negotiate a custom picture format, which is not limited by a number of fixed formats anymore. The number of lines can be from 4 to 1,152 as long as it is divisible by 4, and the number of pels per line can be from 4 to 2,048 as long as it is divisible by 4.

Custom Pixel Aspect Ratios (PARs)—This allows the use of additional PARs other than those used in CIF (11:12), SIF (10:11) and the square (1:1) aspect ratio. All custom PARs are shown in Table 5.28.

Table 5.28 Custom PARs [5.176].

PAR	Pixel width: pixel height
Square	1:1
CIF	12:11
525 type for 4:3 picture	10:11
CIF for 16:9 picture	16:11
525 type for 16:9 picture	40:33
Extended PAR	m:n, m and n are relatively prime

©1997 ITU-T.

Among the new negotiable coding options specified by H.263 Version 2, five of them are intended to improve coding efficiency:

Advanced intracoding mode—This is an optional mode for intracoding. In this mode, intrablocks are coded using a predictive method. A block is predicted from the block to the left or the block above, as shown in Figure 5.112. For isolated intrablocks for which no prediction can be found, the prediction is simply turned off.

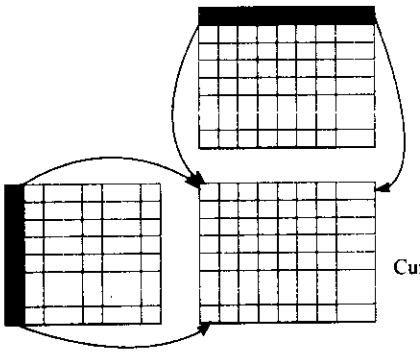


Figure 5.112
Advanced intracoding mode.

Alternate inter-VLC mode—This mode provides the ability to apply a VLC table originally designed for intra coding to intercoding, where there are often many large coefficients, by simply using a different interpretation of the level and the run.

Modified quantization mode—This mode improves the flexibility of controlling the quantizer step size. It also reduces the quantizer step size for chrominance quantization in order to reduce the chrominance artifacts. An extension of the range of values of the DCT coefficient is also provided. In addition, by prohibiting certain unreasonable coefficient representations, this mode increases error detection performance and reduces decoding complexity.

Deblocking filter mode—In this mode, an adaptive filter is applied across the 8x8 block edge boundaries of decoded I- and P-pictures to reduce blocking artifacts. The filter affects the picture that is used for the prediction of subsequent pictures and thus lies within the motion-prediction loop, similar to the loop filtering in H.261.

Improved PB-frame mode—This mode deals with the problem that the PB-frame mode in H.263 cannot represent large motion very well. It provides a mode with more robust performance under complex motion conditions. Instead of constraining a forward motion vector and a backward motion vector to come from a single motion vector as in H.263, the improved PB-frame mode allows them to be totally independent as in the B-frames of MPEG.

The following optional modes are designed to address the needs of mobile video and other unreliable transport environments:

Slice structured mode—In this mode, a slice structure replaces the GOB structure. Slices have more flexible shapes and may appear in any order within the bit stream for a picture. Each slice has a specified width. The use of slices allows a flexible partitioning of the picture in contrast to the fixed partitioning and fixed transmission order required by the GOB structure. This can provide enhanced error resilience and can minimize the video delay.

Reference picture selection mode—In this mode, the reference picture does not have to be the most recently encoded picture. Instead, any temporally previous picture can be referenced. This mode can provide better error resilience in unreliable channels, such as mobile and packet networks, because the codec can avoid using an erroneous picture for future reference.

Independent segment decoding mode—This mode improves error resilience by ensuring that any error in a certain region of the picture does not propagate to other regions.

The temporal, SNR, and spatial scalability modes support layered-bit-stream scalability in three forms, similar to MPEG-2. Bidirectionally predicted frames, which are the same as those used in MPEG, are used for temporal scalability by adding enhancement frames between other coded frames. This is shown in Figure 5.113. A similar syntactical structure is used to provide an enhancement layer of video data to support spatial scalability by adding enhancement information for construction of a higher-resolution picture, as shown in Figure 5.114. Finally, SNR scalability is provided by adding enhancement information for reconstruction of a higher-fidelity picture with the same picture resolution, as in Figure 5.115. Furthermore, different scalabilities can be combined together in a very flexible way (Figure 5.116).

Two other enhancement modes are described in H.263 Version 2:

Reference picture resampling mode—This allows a prior-coded picture to be resampled, or warped, before it is used as a reference picture (Figure 5.117). The warping is defined by four

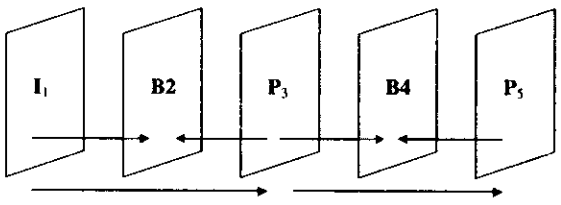


Figure 5.113 Temporal scalability [5.176]. ©1997 ITU-T.

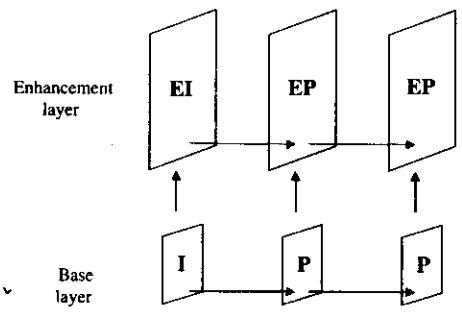


Figure 5.114 Spatial scalability [5.176]. ©1997 ITU-T.