Editor: Peiya Liu
Siemens Corporate Research

MPEG-4: A Multimedia Standard for the Third Millennium, Part 1

Stefano Battista
bSoft

Franco Casalino
Ernst & Young Consultants

Claudio Lande
CSELT

MPEG-4 (formally ISO/IEC international standard 14496) defines a multimedia system for interoperable communication of complex scenes containing audio, video, synthetic audio, and graphics material. In part 1 of this two-part article we provide a comprehensive overview of the technical elements of the Moving Pictures Expert Group's MPEG-4 multimedia system specification. In part 2 (in the next issue) we describe an application scenario based on digital satellite television broadcasting, discuss the standard's envisaged evolution, and compare it to other activities in forums addressing multimedia specifications.

Evolving standard

MPEG-4 started in July 1993, reached Committee Draft level in November 1997, and achieved International Standard level in April 1999. MPEG-4 combines some typical features of other MPEG standards, but aims to provide a set of technologies to satisfy the needs of authors, service providers, and end users.

For authors, MPEG-4 will enable the production of content with greater reusability and flexibility than possible today with individual technologies such as digital television, animated graphics, World Wide Web (WWW) pages, and their extensions. Also, it permits better management and protection of content owner rights.

For network service providers, MPEG-4 will offer transparent information, interpreted and translated into the appropriate native signaling messages of each network with the help of relevant standards bodies. However, the foregoing excludes quality-of-service (QoS) considerations, for which MPEG-4 will provide a generic QoS descriptor for different MPEG-4 media. The exact translations from the QoS parameters set for each media to the network QoS exceed the scope of MPEG-4 and remain for network providers to define.

For end users, MPEG-4 will enable many functionalities potentially accessable on a single compact terminal and higher levels of interaction with content, within the limits set by the author.

MPEG-4 achieves these goals by providing standardized ways to support

- *Coding*—representing units of audio, visual, or audiovisual content, called media objects. These media objects are natural or synthetic in origin, meaning they could be recorded with a camera or microphone, or generated with a computer.
- *Composition*—describing the composition of these objects to create compound media objects that form audiovisual scenes.
- *Multiplex*—multiplexing and synchronizing the data associated with media objects for transport over network channels providing a QoS appropriate for the nature of the specific media objects.
- *Interaction*—interacting with the audiovisual scene at the receiver's end or, via a back channel, at the transmitter's end.

The structure of the MPEG-4 standard consists of six parts: Systems,[1] Visual,[2] Audio,[3] Conformance Testing, Reference Software, and Delivery Multimedia Integration Framework (DMIF).[4]

Systems

The Systems subgroup[1] defined the framework for integrating the natural and synthetic components of complex multimedia scenes. The Systems level shall integrate the elementary decoders for media components specified by other MPEG-4 subgroups—Audio, Video, Synthetic and Natural Hybrid Coding (SNHC), and Intellectual Property Management and Protection (IPMP)—providing

1070-986X/99/$10.00 © 1999 IEEE

the specification for the parts of the system related to composition and multiplex.

Composition information consists of the representation of the hierarchical structure of the scene. A graph describes the relationship among elementary media objects comprising the scene.

The MPEG-4 Systems subgroup adopted an approach for composition of elementary media objects inspired by the existing Virtual Reality Modeling Language (VRML)[5] standard. VRML provides the specification of a language to describe the composition of complex scenes containing 3D material, plus audio and video.

The resulting specification addresses issues specific to an MPEG-4 system:

- description of objects representing natural audio and video with streams attached, and
- description of objects representing synthetic audio and video (2D and 3D material) with streams attached (such as streaming text or streaming parameters for animation of a facial model).

The techniques adopted for multiplexing the elementary streams borrow from the experience of MPEG-1 and MPEG-2 Systems for timing and synchronization of continuous media. A specific three-layer multiplex strategy defined for MPEG-4 fits the requirements of a wide range of networks and of very different application scenarios.

Figure 1 shows a high-level diagram of an MPEG-4 system's components. It serves as a reference for the terminology used in the system's design and specification: the demultiplexer, the elementary media decoders (natural audio, natural video, synthetic audio, and synthetic video), the specialized decoders for the composition information, and the specialized decoders for the protection information.

The following subsections present the composition and multiplex aspects of the MPEG-4 Systems in more detail.

Composition

The MPEG-4 standard deals with frames of audio and video (vectors of samples and matrices of pixels). Further, it deals with the objects that make up the audiovisual scene. Thus, a given scene has a number of video objects, of possibly differing shapes, plus a number of audio objects, possibly associated to video objects, to be combined before presentation to the user. Composition encompasses the task of combining all of the separate entities that make up the scene.

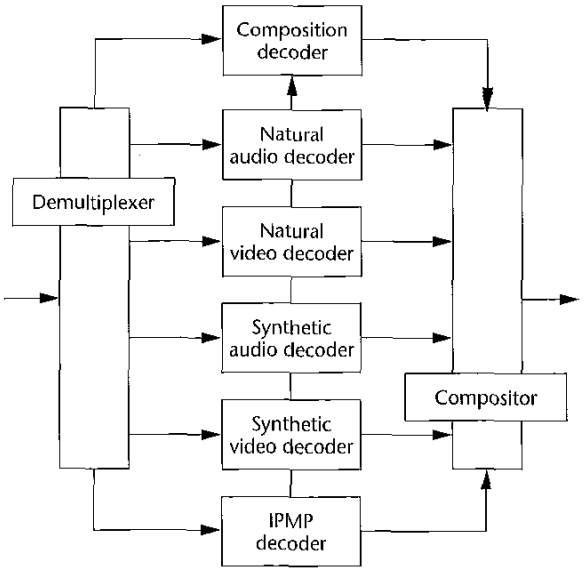


Figure 1. MPEG-4 high-level system architecture (receiver terminal).

The model adopted by MPEG-4 to describe the composition of a complex multimedia scene relies on the concepts VRML uses. Basically, the Systems group decided to reuse as much of VRML as possible, extending and modifying it only when strictly necessary.

The main areas featuring new concepts according to specific application requirements are

- dealing with 2D-only content, for a simplified scenario where 3D graphics is not required;
- interfacing with streaming media (video, audio, streaming text, streaming parameters for synthetic objects); and
- adding synchronization capabilities.

The outcome is the specification of a VRML-based composition format with extensions tuned to match MPEG-4 requirements. The scene description represents complex scenes populated by synthetic and natural audiovisual objects with their associated spatiotemporal transformations.

The author can generate this description in textual format, possibly through an authoring tool. The scene's description then conforms to the VRML syntax with extensions. For efficiency, the standard defines a way to encode the scene description in a binary representation—Binary Format for Scene Description (BIFS).

Multimedia scenes are conceived as hierarchical structures represented as a graph. Each leaf of

the graph represents a media object (audio; video; synthetic audio like a Musicial Instrument Digital Interface, or MIDI, stream; synthetic video like a face model). The graph structure isn't necessarily static, as the relationships can evolve over time as nodes or subgraphs are added or deleted. All the parameters describing these relationships are part of the scene description sent to the decoder.

The initial snapshot of the scene is sent or retrieved on a dedicated stream. It is then parsed, and the whole scene structure is reconstructed (in an internal representation) at the receiver terminal. All the nodes and graph leaves that require streaming support to retrieve media contents or ancillary data (video stream, audio stream, facial animation parameters) are logically connected to the decoding pipelines.

An update of the scene structure may be sent at any time. These updates can access any field of any updatable node in the scene. An updatable node is one that received a unique node identifier in the scene structure. The user can also interact locally with the scenes, which may change the scene structure or the value of any field of any updatable node.

Composition information (information about the initial scene composition and the scene updates during the sequence evolution) is, like other streaming data, delivered in one elementary stream. The composition stream is treated differently from others because it provides the information required by the terminal to set up the scene structure and map all other elementary streams to the respective media objects.

Spatial relationships. The media objects may have 2D or 3D dimensionality. A typical video object (a moving picture with associated arbitrary shape) is 2D, while a wire-frame model of a person's face is 3D. Audio also may be spatialized in 3D, specifying the position and directional characteristics of the source.

Each elementary media object is represented by a leaf in the scene graph and has its own local coordinate system. The mechanism to combine the scene graph's nodes into a single global coordinate system uses spatial transformations associated to the intermediate nodes, which group their children together. Following the graph branches from bottom to top, the spatial transformations cascade to reach the unique coordinate system associated to the root of the graph.

Temporal relationships. The composition stream (BIFS) has its own associated time base. Even if the time bases for the composition and for the elementary data streams differ, they must be consistent except for translation and scaling of the time axis. Time stamps attached to the elementary media streams specify at what time the access unit for a media object should be ready at the decoder input (DTS, decoding time stamp), and at what time the composition unit should be ready at the compositor input (CTS, composition time stamp). Time stamps associated to the composition stream specify at what time the access units for composition must be ready at the input of the composition information decoder.

In addition to the time stamps mechanism (derived from MPEG-1 and MPEG-2), fields within the scene description also carry a time value. They indicate either a duration in time or an instant in time. To make the latter consistent with the time stamps scheme, MPEG-4 modified the semantics of these absolute time fields in seconds to represent a relative time with respect to the time stamp of the BIFS elementary stream. For example, the start of a video clip represents the relative offset between the composition time stamp of the scene and the start of the video display.

Multiplex

Because MPEG-4 is intended for use on a wide variety of networks with widely varying performance characteristics, it includes a three-layer multiplex standardized by the Digital Media Integration Framework (DMIF)[4] working group. The three layers separate the functionality of

- adding MPEG-4–specific information for timing and synchronization of the coded media (synchronization layer);
- multiplexing streams with very different characteristics, such as average bit rate and size of access units (flexible multiplex layer); and
- adapting the multiplexed stream to the particular network characteristics in order to facilitate the interface to different network environments (transport multiplex layer).

The goal is to exploit the characteristics of each network, while adding functionality that these environments lack and preserving a homogeneous interface toward the MPEG-4 system.

Elementary streams are packetized, adding headers with timing information (clock references) and synchronization data (time stamps).

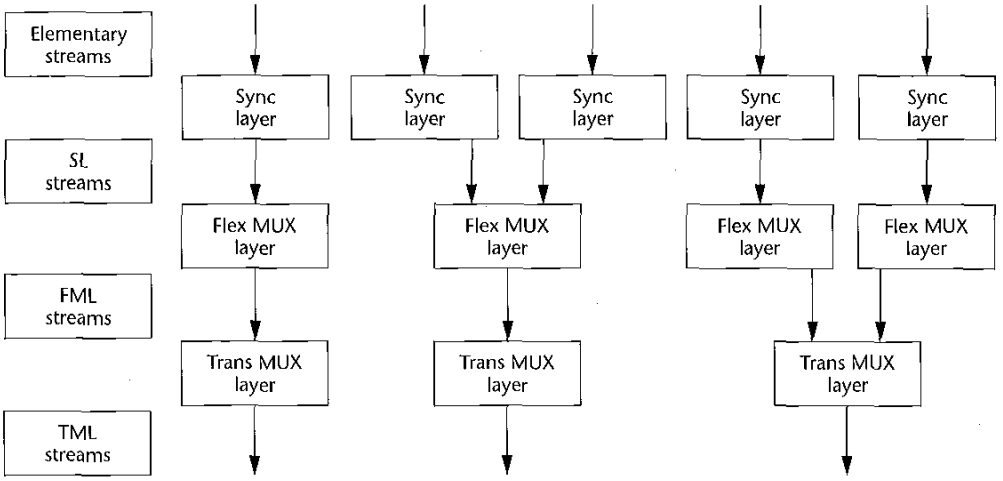


Figure 2. General structure of the MPEG-4 multiplex. Different cases have multiple SL streams multiplexed in one FML stream and multiple FML streams multiplexed in one TML stream.

They make up the synchronization layer (SL) of the multiplex.

Streams with similar QoS requirements are then multiplexed on a content multiplex layer, termed the flexible multiplex layer (FML). It efficiently interleaves data from a variable number of variable bit-rate streams.

A service multiplex layer, known as the transport multiplex layer (TML), can add a variety of levels of QoS and provide framing of its content and error detection. Since this layer is specific to the characteristics of the transport network, the specification of how data from SL or FML streams is packetized into TML streams refers to the definition of the network protocols— MPEG-4 doesn't specify it.

Figure 2 shows these three layers and the relationship among them.

Synchronization layer (timing and synchronization). Elementary streams consist of access units, which correspond to portions of the stream with a specific decoding time and composition time. As an example, an elementary stream for a natural video object consists of the coded video object instances at the refresh rate specific to the video sequence (for example, the video of a person captured at 25 pictures per second). Or, an elementary stream for a face model consists of the coded animation parameters instances at the refresh rate specific to the face model animation (for example, a model animated to refresh the facial animation parameters 30 times per second). Access units like a video object instance or a facial animation parameters instance are the self-contained semantic units in the respective streams, which have to be decoded and used for composition synchronously with a common system time base.

Elementary streams are first framed in SL packets, not necessarily matching the size of the access units in the streams. The header attached by this first layer contains fields specifying

- *Sequence number*—a continuous number for the packets, to perform packet loss checks
- *Instantaneous bit rate*—the bit rate at which the elementary stream is coded
- *OCR (object clock reference)*—a time stamp used to reconstruct the time base for the single object
- *DTS (decoding time stamp)*—a time stamp to identify the correct time to decode an access unit
- *CTS (composition time stamp)*—a time stamp to identify the correct time to render a decoded access unit

The information contained in the SL headers maintains the correct time base for the elementary decoders and for the receiver terminal, plus the correct synchronization in the presentation of the elementary media objects in the scene. The clock references mechanism supports timing of the system, and the mechanism of time stamps supports synchronization of the different media.

Flexible multiplex layer (content). Given the wide range of possible bit rates associated to the elementary streams—ranging, for example, from 1 Kbps for facial animation parameters to 1 Mbps

for good-quality video objects—an intermediate multiplex layer provides more flexibility. The SL serves as a tool to associate timing and synchronization data to the coded material. The transport multiplex layer adapts the multiplexed stream to the specific transport or storage media. The intermediate (optional) flexible multiplex layer provides a way to group together several low-bit-rate streams for which the overhead associated to a further level of packetization is not necessary or introduces too much redundancy. With conventional scenes, like the usual audio plus video of a motion picture, this optional multiplex layer can be skipped; the single audio stream and the single video stream can be mapped each to a single transport multiplex stream.

Transport multiplex layer (service). The multiplex layer closest to the transport level depends on the specific transmission or storage system on which the coded information is delivered. The Systems part of MPEG-4 doesn't specify the way SL packets (when no FML is used) or FML packets are mapped on TML packets. The specification simply references several different transport packetization schemes. The "content" packets (the coded media data wrapped by SL headers and FML headers) may be transported directly using an Asynchronous Transfer Mode (ATM) Adaptation Layer 2 (AAL2) scheme for applications over ATM, MPEG-2 transport stream packetization over networks providing that support, or transport control protocol/Internet protocol (TCP/IP) for applications over the Internet.

Intellectual property protection

The MPEG-4 standard specifies a multimedia-bit-stream syntax, a set of tools, and interfaces for designers and builders of a wide variety of multimedia applications. Each of these applications has a set of requirements regarding protection of the information it manages. These applications can produce conflicting content management and protection requirements. By implication, the Intellectual Property Management and Protection (IPMP) framework design needs to consider the MPEG-4 standard's complexity and the diversity of its applications.

The IPMP framework consists of a normative interface that permits an MPEG-4 terminal to host one or more IPMP systems. An IPMP system is a non-normative component that provides intellectual property management and protection functions for the terminal.

Video

The most important goal of both the MPEG-1 and MPEG-2 standards was to make the storage and transmission of digital audiovisual material more efficient through compression techniques. To achieve this, both deal with frame-based video and audio. Interaction with the content is limited to the video frame level, with its associated audio.

MPEG-4 Video functionalities

MPEG-4 Video[2] supports different functionalities that divide into three nonorthogonal classes based on the requirements they support:

- *Content-based interactivity.* This class includes four functionalities focused on requirements for applications involving some form of interactivity between the user and the data: content-based multimedia data access tools, content-based manipulation and bit-stream editing, hybrid natural and synthetic data coding, and improved temporal random access.

- *Compression.* This class consists of two functionalities: improved coding efficiency and coding of multiple concurrent data streams. These essentially target applications requiring efficient storage or transmission of audiovisual information and their effective synchronization.

- *Universal access.* The remaining two functionalities are robustness in error-prone environments and content-based scalability. These functionalities make MPEG-4 encoded data accessible over a wide range of media, with various qualities in terms of temporal and spatial resolutions for specific objects, decodable by a range of decoders with different complexities.

The error resilience tools developed for video divide into synchronization, data recovery, and error concealment. The basic scalability tools offered are temporal scalability and spatial scalability. MPEG-4 Video also supports combinations of these basic scalability tools, referred to as hybrid scalability. Basic scalability allows two layers of video, referred to as the lower layer and the enhancement layer, whereas hybrid scalability supports up to four layers.

MPEG-4 Video provides tools and algorithms for

- efficient compression of images and video

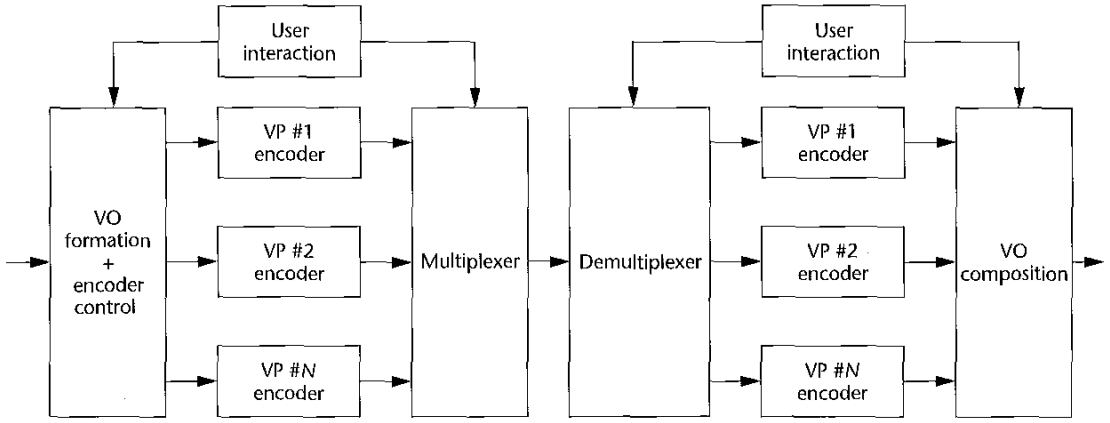


Figure 3. General structure of the MPEG-4 video encoder/decoder. (Note, VO represents video object.)

- efficient compression of textures for texture mapping on 2D and 3D meshes
- efficient compression of implicit 2D meshes
- efficient compression of time-varying geometry streams that animate meshes
- efficient random access to all types of visual objects
- extended manipulation functionality for images and video sequences
- content-based coding of images and video
- content-based scalability of textures, images, and video
- spatial, temporal, and quality scalability
- error robustness and resilience in error-prone environments

MPEG-4 video encoder and decoder structure

MPEG-4 includes the concepts of *video object* and *video object plane*. A video object in a scene is an entity that a user may access and manipulate. The instances of video objects at a given time are called video object planes (VOPs). The encoding process generates a coded representation of a VOP plus composition information necessary for display. Further, at the decoder a user may interact with and modify the composition process as needed.

The full syntax allows coding of rectangular as well as arbitrarily shaped video objects in a scene. Further, the syntax supports both nonscalable and scalable coding. The scalability syntax enables reconstructing useful video from pieces of a bit stream by structuring the total bit stream in two or more layers, starting from a stand-alone base layer and adding a number of enhancement layers. The base layer can be coded using a nonscalable syntax or, in the case of picture-based coding, even using the syntax of a different video coding standard.

The ability to access individual objects requires achieving a coded representation of their shape. A natural video object consists of a sequence of 2D representations (at different points in time) referred to here as VOPs. Efficient coding of VOPs exploits both temporal and spatial redundancies. Thus a coded representation of a VOP includes representation of its shape, its motion, and its texture.

Figure 3 shows the block diagram of the MPEG-4 video encoder/decoder. The most important feature is the intrinsic representation based on video objects when defining a visual scene. In fact, a user—or an intelligent algorithm—may choose to encode the different video objects composing source data with different parameters or different coding methods, or may even choose not to code some of them at all.

In most applications, each video object represents a semantically meaningful object in the scene. To maintain a certain compatibility with available video materials, each uncompressed video object is represented as a set of *Y*, *U*, and *V* components, plus information about its shape, stored frame after frame at predefined temporal intervals.

Another important feature of the video standard is that this approach doesn't explicitly define a temporal frame rate. This means that the encoder and decoder can function in different frame rates, which don't even need to stay constant throughout the video sequence (or the same for the various video objects).

Interactivity between the user and the encoder or the decoder takes place in different ways. The user may decide to interact at the encoding level, either in coding control to distribute the available bit rate between different video objects, for instance, or to influence the multiplexing to change parameters such as the composition script at the encoder. In cases where no back channel is available, or when the compressed bit stream already exists, the user may interact with the decoder by acting on the compositor to change either the position of a video object or its display depth order. The user can also influence the decoding at the receiving terminal by requesting the processing of a portion of the bit stream only, such as the shape.

The decoder's structure resembles that of the encoder—except in reverse—apart from the composition block at the end. The exact method of composition (blending) of different video objects depends on the application and the method of multiplexing used at the system level.

Audio

To achieve the highest audio quality within the full range of bit rates and at the same time provide extra functionalities, the MPEG-4 Audio[3] standard includes six types of coding techniques:

- Parametric coding modules
- Linear predictive coding (LPC) modules
- Time/frequency (T/F) coding modules
- Synthetic/natural hybrid coding (SNHC) integration modules
- Text-to-speech (TTS) integration modules
- Main integration modules, which combine the first three modules to a scaleable encoder

While the first three parts describe real coding schemes for the low-bit-rate representation of natural audio sources, the SNHC and TTS parts only standardize the interfaces to general SNHC and TTS systems. Because of this, already established synthetic coding standards, such as MIDI, can be integrated into the MPEG-4 Audio system. The TTS interfaces permit plugging TTS modules optimized for a special language into the general framework.

The main module contains global modules, such as the speed change functionality of MPEG-4 and the scalability add-ons necessary to implement large-step scalability by combining different coding schemes. Such scalability modules also allow the use of the International Telecommunications Union-Telecommunications (ITU-T) codecs within the scalable schemes.

Each of the natural coding schemes should cover a specific range of bit rates and applications. The following focuses on the tools most important in the current standard.

Parametric coding

The HVXC (harmonic vector excitation coding) decoder tools allow decoding of speech signals at 2 Kbps (and higher, up to 6 Kbps), while the individual line decoder tools allow decoding of nonspeech signals like music at bit rates of 4 Kbps and higher. Both sets of decoder tools allow independent change of speed and pitch during the decoding and can be combined to handle a wider range of signals and bit rates.

The HVXC decoder's basic decoding process consists of four steps: inverse quantization of parameters, generation of excitation signals for voiced frames by sinusoidal synthesis (harmonic synthesis), generation of excitation signals for unvoiced frames by codebook look-up, and linear predictive coding synthesis. Spectral postfilters enhance the synthesized speech quality.

CELP coding

While the parametric schemes currently allow for the lowest bit rates in MPEG-4 Audio, both narrow-band (4 kHz audio gross bandwidth) and wideband (8 kHz audio gross bandwidth) code-excited linear prediction (CELP) encoders cover the next higher range of bit rates. In general they offer the following advantages over the parametric encoders in the bit-rate range from about 6 to 24 Kbps:

- Lower delay (15- to 40-ms algorithmic delay, compared to about 90 ms for the parametric speech coder)
- At higher rates, better performance for signals not easily described by parametric models

In MPEG-4, linear predictive coding (LPC) is realized by means of CELP coding techniques. CELP is a general analysis-by-synthesis model, based on the combination of LPC and codebook excitation. In this model, linear prediction deals with the relevant speech parameters of spectral envelope (short-term prediction) and pitch (long-

term prediction), and codebook excitation takes into account the nonpredictive part of the signal.

The new feature of CELP coding in MPEG-4 is the scalability in audio bandwidth, bit rate, and delay. Different coding schemes, using the same set of basic functions, can be combined, including a bit-rate-adjustable narrow-band speech coder operating at bit rates from 5 to 12 Kbps with an algorithmic delay of 25 ms, a coder offering better performance at a higher delay, or a wide-band speech coder operating with bit rates from 16 to 24 Kbps. Just recently, the addition of a true scalable coding scheme has been proposed for narrow-band CELP coding.

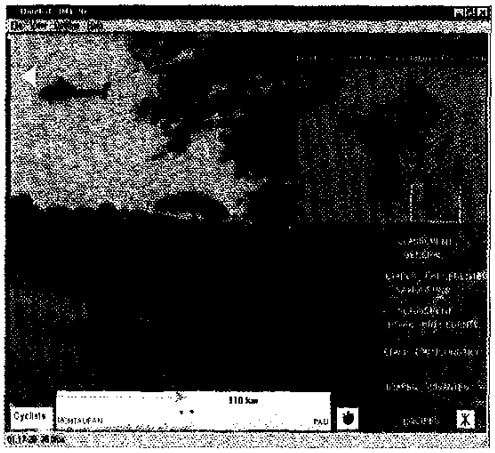


Figure 4. An MPEG-4 application called "Le tour de France" featuring many different A/V objects.

Time/frequency coding

This coding scheme is characterized best by coding the input signal's spectrum. The input signal is first transformed into a different representation that gives access to its spectral components. Since these codecs don't rely on a special model of the input signal, they suit encoding any type of input signal. The usable bit rate ranges from 16 Kbps for a 7-kHz audio bandwidth to up to more than 64 Kbps per audio channel for CD-like quality coding of mono, stereo, or multichannel audio.

The most important tools included in the time/frequency part derive from the MPEG-2 Advanced Audio Coding (AAC) standard. Many optional tools modify one or more of the spectra to provide more efficient coding. For those operating in the spectral domain, the option to "pass through" is retained, and in all cases where a spectral operation is omitted, the spectra at its input pass directly through the tool without modification.

Synthetic and natural hybrid coding

SNHC deals with the representation and coding of synthetic (2D and 3D graphics) and natural (still images and natural video) audiovisual information. SNHC represents an important aspect of MPEG-4 for combining mixed media types including streaming and downloaded A/V objects.

SNHC fields of work include 2D and 3D graphics, human face and body description and animation, integration of text and graphics, scalable textures encoding, 2D/3D mesh coding, hybrid text-to-speech coding, and synthetic audio coding (structured audio).

Media integration of text and graphics

MITG provides a way to encode, synchronize, and describe the layout of 2D scenes composed of animated text, audio, video, synthetic graphic shapes, pointers, and annotations. The 2D BIFS graphics objects derive from and are a restriction of the corresponding VRML 2.0 3D nodes. Many different types of textures can be mapped on plane objects: still images, moving pictures, complete MPEG-4 scenes, or even user-defined patterns. Alternatively, many material characteristics (color, transparency, border type) can be applied on 2D objects.

Other VRML-derived nodes are the interpolators and the sensors. Interpolators allow predefined object animations like rotations, translations, and morphing. Sensors generate events that can be redirected to other scene nodes to trigger actions and animations. The user can generate events, or events can be associated to particular time instants.

MITG provides a Layout node to specify the placement, spacing, alignment, scrolling, and wrapping of objects in the MPEG-4 scene. Still images or video objects can be placed in a scene graph in many ways, and they can be texture-mapped on any 2D object. The most common way, though, is to use the Bitmap node to insert a rectangular area in the scene in which pixels coming from a video or still image can be copied.

The 2D scene graphs can contain audio sources by means of the Sound2D nodes. Like visual objects, they must be positioned in space and time. They are subject to the same spatial transformations of their parents' nodes hierarchically above them in the scene tree.

Text can be inserted in a scene graph through the Text node. Text characteristics (font, size, style, spacing, and so on) can be customized by means of the FontStyle node.

Figure 4 shows a rather complicated MPEG-4 scene from "Le tour de France" with many differ-

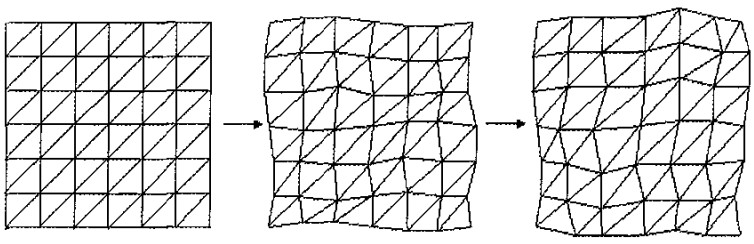


Figure 5. Mesh object with uniform triangular geometry.

ent object types like video, icons, text, still images for the map of France and the trail map, and a semitransparent pop-up menu with clickable items. These items, if selected, provide information about the race, the cyclists, the general placing, and so on.

3D graphics

The advent of 3D graphics triggered the extension of MPEG-4 to the third dimension. BIFS 3D nodes—an extension of the ones defined in VRML specifications—allow the creation of virtual worlds.

Like in VRML, it's possible to add behavior to objects through Script nodes. Script nodes contain functions and procedures (the terminal must support the Javascript programming language) that can define arbitrary complex behaviors like performing object animations, changing the values of nodes' fields, modifying the scene tree, and so on.

MPEG-4 allows the creation of much more complex scenes than VRML, of 2D/3D hybrid worlds where contents are not downloaded once but can be streamed to update the scene continuously.

Face animation

Face animation focuses on delineating parameters for face animation and definition. It has a very tight relationship with hybrid scalable text-to-speech synthesis for creating interesting applications based on speech-driven avatars. Despite previous research on avatars, the face animation work is the first attempt to define in a standard way the sets of parameters for synthetic anthropomorphic models.

Face animation is based on the development of two sets of parameters: facial animation parameters (FAPs) and facial definition parameters (FDPs).

FAPs allow having a single set of parameters regardless of the face model used by the terminal or application. Most FAPs describe atomic movements of the facial features; others (expressions and visemes) define much more complex deformations. Visemes are the visual counterparts of phonemes and hence define the position of the mouth (lips, jaw, tongue) associated with phonemes. In the context of MPEG-4, the expressions mimic the facial expressions associated with human primary emotions like joy, anger, fear, surprise, sadness, and disgust.

Animated avatars' animation streams fit very low bit-rate channels (about 4 Kbps).

FAPs can be encoded either with arithmetic encoding or with discrete cosine transform (DCT).

FDPs are used to calibrate (that is, modify or adapt the shape of) the receiver terminal default face models or to transmit completely new face model geometry and texture.

2D mesh encoding

A 2D mesh object in MPEG-4 represents the geometry and motion of a 2D triangular mesh, that is, tessellation of a 2D visual object plane into triangular patches. A dynamic 2D mesh is a temporal sequence of 2D triangular meshes. The initial mesh can be either uniform (described by a small set of parameters) or Delaunay (described by listing the coordinates of the vertices or nodes and the edges connecting the nodes). Either way, it must be simple—it cannot contain holes.

Once the mesh has been defined, it can be animated by moving its vertices and warping its triangles. To achieve smooth animations, motion vectors are represented and coded with half-pixel accuracy. When the mesh deforms, its topology remains unchanged. Updating the mesh shape requires only the motion vectors that express how to move the vertices in the new mesh.

An example of a rectangular mesh object borrowed from the MPEG-4 specification appears in Figure 5.

Dynamic 2D meshes inserted in an MPEG-4 scene create 2D animations. This results from mapping textures (video object planes, still images, 2D scenes) onto 2D meshes.

Texture coding

MPEG-4 supports an ad-hoc tool for encoding textures and still images based on a wavelet algorithm that provides spatial and quality scalability, content-based (arbitrarily shaped) object coding, and very efficient data compression over a large range of bit rates. Texture scalability comes through many (up to 11) different levels of spatial resolutions, allowing progressive texture transmission and many alternative resolutions (the analog of mipmapping in 3D graphics). In other words, the wavelet technique provides for scalable bit-stream coding in the form of an image-resolution pyramid for progressive transmission and

temporal enhancement of still images. For animation, arbitrarily shaped textures mapped onto 2D dynamic meshes yield animated video objects with a very limited data transmission.

Texture scalability can adapt texture resolution to the receiving terminal's graphics capabilities and the transmission rate to the channel bandwidth. For instance, the encoder may first transmit a coarse texture and then refine it with more texture data (levels of the resolution pyramid).

Structured Audio

Structured Audio allows creating synthetic sounds starting from coded input data. A special synthesis language called Structured Audio Orchestra Language (SAOL) permits defining a synthetic orchestra whose instruments can generate sounds like real musical instruments or process prestored sounds. MPEG-4 doesn't standardize SAOL's methods of generating sounds; it standardizes the method of describing synthesis.

Downloading scores in the bit stream controls the synthesis. Scores resemble scripts in a special language called Structured Audio Score Language (SASL). They consist of a set of commands for the various instruments. These commands can affect different instruments at different times to generate a large range of sound effects. If fine control over the final synthesized sound isn't needed, it's easier to control the orchestra through the MIDI format. Supporting MIDI adds to MPEG-4 scenes the ability to reuse and import a huge quantity of existing audio contents.

The bit rate needed by Synthetic Audio applications ranges from few bits per second to 2 or 3 Kbps when controlling many instruments and performing very fine coding. For terminals with less functionality, and for applications that don't need sophisticated synthesis, MPEG-4 also standardizes a wavetable bank format. This format permits downloading sound samples for use in wavetable synthesis, as well as simple processing tools (reverb, chorus, and so on.)

MPEG-4 text-to-speech

MPEG-4 doesn't define a specific text-to-speech technique but rather the binary representation of a TTS stream and the interfaces of an MPEG-4 text-to-speech (M-TTS) with the other parts of an MPEG-4 decoder. An M-TTS stream may contain many different information types about the synthetic voice apart from text: gender, age, speech rate, language code, prosody, and lip shape information. It may contain fields that allow trick mode (fastforwarding, pausing, playing, or rewinding the synthetic speech).

An M-TTS can also carry the International Phonetic Alphabet (IPA) coded phonemes with their time duration. Handed to the face animation engine in the MPEG-4 player, they can produce speech-driven face animation. In this case the face animation system doesn't receive a FAP stream from the MPEG-4 demultiplexer; instead it converts phonemes into visemes and uses them to perform the face model deformations. The phoneme duration synchronizes model animation and speech.

Interestingly, applications require a tiny channel bandwidth—from 200 bps to 1.2 Kbps.

This concludes part 1. We'll look at applications and what comes next for MPEG-4 in part 2. **MM**

For More Information

ISO official site: http://www.iso.ch/
MPEG official site: http://www.cselt.it/mpeg/
MPEG-4 Systems site: http://garuda.imag.fr/MPEG4/
MPEG-4 Visual site: http://wwwam.hhi.de/mpeg-video/
MPEG-4 Audio site: http://www.tnt.uni-hannover.de/project/mpeg/audio/
MPEG-4 SNHC site: http://www.es.com/mpeg4-snhc/
MPEG-4 Synthetic Audio site: http://sound.media.mit.edu/mpeg4/
Web3D (formerly VRML) official site: http://www.web3d.org/
IPA (International Phonetic Alphabet) site: http://www.arts.gla.ac.uk/IPA/ipa.html

References

Available to MPEG members or from ISO (http://www.iso.ch) or the National Standards bodies (for example, American National Standards Institute (ANSI) in the US:

1. MPEG-4 Part 1: Systems (IS 14496-1), doc. N2501, Atlantic City, N.J., USA, Oct. 1998.
2. MPEG-4 Part 2: Visual (IS 14496-2), doc. N2502, Atlantic City, N.J., USA, Oct. 1998.
3. MPEG-4 Part 3: Audio (IS 14496-3), doc. N2503, Atlantic City, NJ, USA, Oct. 1998.
4. MPEG-4 Part 6: DMIF (IS 14496-6), doc. N2506, Atlantic City, N.J., USA, Oct. 1998.
5. VRML (IS 14772-1), "Virtual Reality Modeling Language," April 1997.

Readers may contact Casalino at Ernst & Young Consultants, Corso Vittorio Emanuele II, n. 83, 10128 Torino, Italy, e-mail Franco.Casalino@it.eyi.com.

Contact Standards editor Peiya Liu, Siemens Corporate Research, 755 College Road East, Princeton, NJ 08540, e-mail pliu@scr.siemens.com.