# MPEG STANDARDS EVOLUTION AND IMPACT ON CABLE

Dr. Paul Moroney, Dr. Ajay Luthra
Motorola Broadband Communications Sector

## *Abstract*

*The MPEG standards process has evolved from its beginnings in 1988 through today to cover a wide range of multimedia delivery technology. The entire phenomenon of digital television and the standards involved have affected our culture through increased content choices for the consumer, increased competition, and new types of products. Cable delivery has seen a huge impact, in particular from MPEG-2 based digital television. If we can extrapolate from the evolution of the MPEG process, more change is coming!*

## INTRODUCTION

No one would dispute that the cable industry has been dramatically impacted by the transition to digital television and related services, whether one considers the advent of effective DBS services, or the distribution of digital television to cable headends, or the distribution of digital services through the cable plant to consumers. Although many factors have played a role in the process, certainly the development of true standards was and continues to be a true driver. The ISO/IEC sponsored Motion Picture Experts Group [MPEG] has been the primary organization for the formation of the basic multimedia source coding standards. This paper examines where MPEG has been, and where it is going, to build a vision of the future of digital services over cable. Particular attention is paid to video compression.

The MPEG process began in October of 1988, under the very effective leadership of its convener, Leonardo Chiariglione, who continues in that role to this day. MPEG-1 [1] targeted stored content up to a 1.5 Mbps rate, matching the parameters necessary to store on CDs. MPEG-1 addressed audio compression, video compression, and a storage-oriented systems layer to combine them with other higher-level information, and the resulting standard earned an Emmy award. MPEG-2 [2] shifted the focus to entertainment television, added more video compression tools, multi-channel audio, and a new systems layer more tuned to the need for a transport definition for broadcast. This standard has been deployed very successfully worldwide, and earned an unprecedented second Emmy for MPEG. The designation “MPEG-3” had been set aside for HDTV extensions, but MPEG-2 worked so well for this application that MPEG-3 was never needed.

MPEG-4 [3] began slowly, while MPEG-2 based systems were deployed, and has developed into a true next generation multimedia system. MPEG-4 covers a much broader scope than its predecessors, and achieves dramatic improvements in the underlying source coding technology as well.

MPEG has also begun efforts in two related areas, not focused per se on multimedia compression. MPEG-7 [4] is targeted as multimedia search and retrieval, enabling the creation of search engines for such content. MPEG-21 [5] (the 21 suffix representing the 21st century) addresses the goal of true universal multimedia access. This group focuses on defining a multimedia framework so that broad classes of content can be created, delivered, and consumed in an interoperable manner.

## MPEG-1 AND MPEG-2

MPEG-1 and MPEG-2 are defined in three main parts, addressing video source coding, audio source coding, and a systems layer.

### Video

MPEG video source coding is designed to remove redundancy in video content both frame to frame, and within a frame. Algorithms that achieve these goals without degradation in the reconstructed video images are "lossless," that is, reversible. Algorithms that sacrifice quality in the interests of bit rate reduction are termed "lossy." Lossy algorithms attempt to hide errors, or artifacts, based upon models of human video perception. For example, it is more difficult to detect artifacts under high motion. It is also true that human vision chroma resolution is lower than human vision luminance resolution. Compression systems, such as MPEG, that seek to address a wide range of applications typically employ both lossless and lossy algorithms.

MPEG video compression at its base employs transformation, quantization, and variable length coding (VLC), in that order. Pixel domain representations of video frames are transformed through an 8 by 8 Discrete Cosine Transform (DCT) to a frequency domain representation. The resulting spatial frequency coefficients are scanned in a zigzag manner, and quantized. The resulting sequence of amplitudes and runs of zero values are then Huffman coded. The bits generated are grouped under a syntax organized roughly at the *sequence* level (a number of consecutive frames), the *picture* (typically frame) level, the *slice* level (rows of blocks), the *macroblock* level (4 luminance and 2 chrominance blocks) and the DCT block level, including header information appropriate to each.

The last key aspect is the inclusion of motion estimation and compensation, where the above approach applies to only the difference between a current frame (macroblock) region, and its best possible match in the prior frame. The offsets representing the best match, to the nearest x and y half-pixel, become motion vectors for the region, and are included in as header information.

Figure 1 shows the most basic block diagram of this compression process. Note the presence of the decode (decompress) loop, so that the compressor will be able to transform the pixel differences required for motion compensation.

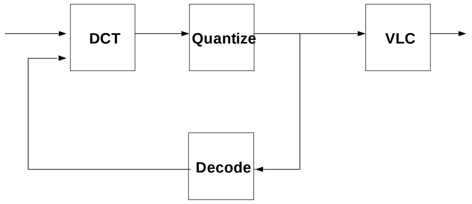

**Figure 1: Basic MPEG Compressor**

Although not shown in the Figure above, MPEG also allows motion estimation from a *future* frame, or from both the previous and future frames, in combination (See Figure 2). In MPEG parlance, a frame without motion compensation is an *I* frame, one with compensation from only the previous frame is a *P* frame, and one with bi-directional motion is a *B* frame. Note that in order to decode a B frame, coded frames must be sent out of (display) order. A decoder must have decoded the future frame before that frame can be used to "anchor" any frame decoded with bi-directional motion.

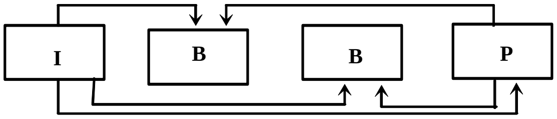

**Figure 2: B Frame Prediction**

The MPEG-2 standard introduced various enhancements, but two major changes were the addition of frame/field mode and direct support for film content. In order to compress interlaced video signals intended to be ultimately displayed on interlaced televisions, (see Figure 3) efficiency is enhanced by allowing the 8 by 8 pixels to be assembled from alternate (interlaced) fields. In fact, throughout the interlaced content, some regions compress better in this fashion [frame mode] and some compress better by taking the 8 by 8 pixel block from each field separately! Thus the standard allows either approach, signaled by header bits.

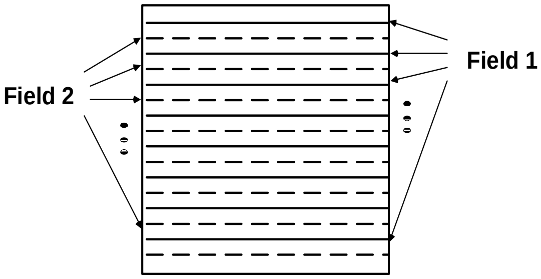

**Figure 3: Interlaced Block Pair**

For film content carried in NTSC television signals, common practice has been to convert the 24 frames per second progressive film to the NTSC 29.97 frames per second interlace (59.94 fields per second) through a *telecine* process known as 3:2 pull down. The first film frame is divided into two fields, and the first field is repeated in the NTSC signal as a "field 3". Frame 2 of the film becomes "field 4" and "field 5," without any repeat. Film frame 3 becomes fields 6 and 7, and field 6 is repeated as a field 8. To complete the cycle, film frame 4 becomes NTSC field 9 and 10. MPEG allows compressors to drop the duplicate fields, and code the film frames directly. An overhead bit called "repeat field" informs the decompressor, or decoder, how to restore NTSC for display.

Given the descriptions above, one can deduce the philosophy adopted by the MPEG planners. The maximal benefit to equipment suppliers, and consumers, taken together, was determined to be achieved through a standard that defines decoders to the extent required to be able to receive all signals that are "MPEG compliant" according to the standard syntax. Thus all aspects of the decoder are defined, with the exception of error recovery and the specifics of the display generation.

The compression process, however, provides substantial room for creativity and invention, within the context of the standard. Any compressor can be built, so long as an MPEG compliant decoder can decode the signals produced. Compression is, thus, far more of an art form, than decompression. Consider the various degrees of freedom. Which frame regions are to be predicted, and how? For predicted regions, what are the best motion vectors? Which frames do you select as I frames, or which slices are not predicted at all, so that decompression equipment can acquire a signal after a channel change, or after an error? How much should any given block or region coefficients be quantized to minimize perceptual loss yet achieve some desired net output bit rate (*rate control*)? How does the equipment estimate such loss? Which regions are best coded as fields, rather than frames? What is the best way to scan the coefficients (there is more than one option provided in the standard)?

MPEG-2 introduced the concept of *profiles* and *levels*, to allow the broad range of compression tools and bit rates to be targeted at classes of applications. This allowed products to be *compliant* at only a specific profile and level; thus low cost consumer devices were possible. No one device was burdened with implementing all the tools of MPEG, at all bit rates.

## Audio

MPEG audio compression bears several similarities to the video compression described above. The overall compression approach involves lossless techniques, as well as lossy quantization. As with video, the approach is designed based upon a well-understood model of human auditory processing, so that errors are hidden as much as possible. Specifically, the ear tends to act as a multi-band analyzer. Within any *critical band*, distortion can be masked if there is enough signal content present. Thus quantization should be applied to these bands, according to this masking phenomenon.

MPEG-1 defined three layers of audio compression algorithms, with increasing complexity and performance. Layer 2 is a sub-band compression approach known as *MUSICAM*, and is broadly deployed in many existing MPEG-2 based systems. Layer 3 adds more compression tools, and performs better at a given bit rate. This layer carries the abbreviated name *MP3*, and has seen extensive recent use over the Internet for audio track distribution.

The MPEG-2 audio standard introduced a backward compatible multi-channel surround version of MPEG-1 defined audio. Thus advanced receivers could decode surround, while earlier equipment could process the stereo audio defined for MPEG-1. As this approach is not completely optimal for multi-channel surround, the MPEG audio sub-group also defined a new non-backward compatible algorithm called *Advanced Audio Coding (AAC)*. AAC provides the best quality overall audio coding approach, at added complexity, both for multi-channel surround and for regular stereo audio. With AAC, 64 kbps stereo audio is generally perceived as excellent.

## Systems

MPEG defined a systems layer to combine the various audio and video streams into a higher-level *program*, and to provide additional information, such as time references. In both MPEG-1 and MPEG-2, the compressed audio and video are termed *elementary streams*. These streams reflect the syntax defined for each media type. The systems layer produces a *Packetized Elementary Stream* (*PES*) from each elementary stream, adding header information such as time markers (*timestamps*), and a type/label description. The resulting PES packets are then combined in one of two structures to build a higher-level construct.

In both MPEG-1 and MPEG-2, PES packets can be combined into a *program stream*, which is more targeted at storage applications. Program streams are characterized by large packet sizes, such as might be appropriate for a sector of a storage medium. The format has little or no error resilience, again, such as would be well matched to a file system. Program streams include a directory structure that defines various general information about the program, and describes all the component packetized elementary streams within the program stream.

*Transport streams* are MPEG-2 specific, intended for the transport of multiplexes of multimedia programs. MPEG-2 transport divides the component PES into 184 byte segments, prepended with a 4 byte header. Each resulting 188-byte transport packet thus includes data from a single PES, and a multiplex of such packets would mix together a number of programs. Content within a multiplex is grouped by a set of identifiers and tables within the multiplex.

Each packet header contains a 13-bit *Packet ID* (*PID*) for reference. The *Program Map Table* (*PMT*) lists the PIDs associated with that program, along with a type value (audio, video, etc.), and optional descriptive information. The PMT is carried in transport packets with its own PID. A single *Program Association Table (PAT)* lists all the programs contained in the multiplex, and includes a pointer to each PMT PID. The PAT is defined to be carried in packets of PID 0.

MPEG-2 transport also includes mechanisms to reference the conditional access data that applies to each program, and the 4 byte packet header includes encryption related bits, sequence numbers, and a provision to include an adaptation field within the body of the packet. Among other functions, the adaptation field must be present often enough to carry a *Program Clock Reference* (*PCR*) for the program. PCR reception allows an MPEG decompressor to rebuild system time, and manage buffers for proper decode and display. Features of this type allow MPEG-2 to support broadcast quality reconstruction of video, which was not an emphasis of MPEG-1.

## MPEG-4

The MPEG-4 process initially focused on developing a new video coding standard for low bit rate coding. During the course of development of the standard, it was realized that a much broader scope was appropriate to support evolving new classes of multimedia applications, including those suited to the Internet. Thus MPEG-4 not only describes new, improved video coding tools, but a much broader multimedia support for these new applications.

First, MPEG-4 was targeted at a broad range of compression resolutions and bit rates. MPEG-4 was designed to be as efficient, or better, than MPEG-1 and the H.263 family of standards at very low bit rates, as efficient or better than MPEG-2 at mid to high bit rates, and so on. MPEG-4 syntax overhead had to be flexible enough to allow efficient operation for all these rates.

Second, to satisfy the needs of such broad applications it was considered important to have independent representation of each media type, e.g. video, graphics, images, speech, animation, synthetic audio, textures and text. This provides much better coding efficiency, since converting media types like text and graphics to video and compressing them as video causes a great loss in quality. (Just consider the rolling text credits at the end of an MPEG-2 heavily compressed movie!) Therefore, in addition to developing *natural* video coding tools, MPEG-4 also developed *Synthetic-Natural Hybrid Coding (SNHC)*, *Texture Coding* and *Sprite Coding* tools.

Third, these broader applications and rich media types drove MPEG-4 to the concept of objects. Rather than considering video as a progression of rectangular frames, scenes are now built from a composition of arbitrarily shaped objects. Certainly this shifts complexity to the decompressor/receiver device in a system, since this device must now compose scenes from these objects, but the potential for new service offerings that this enables is incredible. Not overstated, MPEG-4 allows the content creator to shift a (controlled) portion of the creative process, a portion of the studio, to the user's device.

Hyperlinking of these objects to other objects, such as parameters and text descriptions, offers the cable industry for the first time a true standards based approach to interactivity, with multimedia features only seen today in movie theaters.

## MPEG-4 Natural Video Coding

At a very high level, MPEG-4 video coding, like MPEG-2, is motion compensated DCT based. However, MPEG-4 video coding in general includes more compression tools, and more complex tools, than its predecessors, as such tools are now cost effective to implement.

Coded frames can be of four types – I, P, B and *S*. As MPEG-4 went beyond the concept of coding rectangular video frames to arbitrarily shaped video objects, I, P and B frames are called I, P and B *Video Object Planes (VOPs)*. For video in a rectangular window, VOPs are the same as frames. Similar to MPEG-2, I-VOPs are compressed without any motion compensation, P-VOPs are compressed with motion compensation based on the past I or P-VOPs, and B-VOPs are compressed with motion compensation based on both past and future I or P-VOPs. In addition to I, P and B-VOPs, where the motion compensation is block based, MPEG-4 also allows S-VOPs with *Global Motion Compensation (GMC*), where a single global motion vector is estimated between two VOPs. It is helpful mainly for video sequences with large similar motion across the picture, such as when a camera is panning or zooming.

As mentioned above, MPEG-4 also developed tools to represent and compress arbitrary shaped video objects. Once pictures are broken into multiple objects, each object is compressed independently. Objects can thus have different quality, "frame" rate, and bit rate. In addition, two types of arbitrarily shaped video coding capabilities are defined: *binary* and *gray scale*. In binary shaped coding, objects can be composed to be either in the foreground or in the background. In gray scale shape coding, objects can be composed with 256 levels of transparency, or blending. With object representation, an encoder (now much more than a compressor) needs to have the capability to describe how a particular scene is composed, based on the multiple objects within it, and a receiver device needs to have the capability to recompose the scene in addition to simply decoding the objects and presenting them. Furthermore, the term "scene" is generalized to include all the media types in the content, not only video objects. To facilitate the capability of efficiently describing and sending the dynamically changing scene, a *BInary Format for Scene (BIFS)* description was also developed.

In addition to MPEG-2 type temporal and spatial scalability, a new type of scalability, *Fine Granularity Scalability (FGS)*, is also defined in MPEG-4. Scalability tools allow video to be compressed in multiple layers. To decode a picture, one does not need to receive all the layers; however, the picture quality of the decoded picture can be incrementally improved by decoding more layers. In FGS, as the name suggests, many multiple layers with small incremental numbers of bits can be sent. This technique is very helpful for adapting the compression rate and picture rate to the time varying available bandwidth of a network like the Internet.

MPEG-4 also defined new error resilience tools, which allow bit streams to withstand relatively large bit loss. As an example, the VLC tables can be decoded in both a forward and a reverse direction! All these tools are contained in Part 2 of MPEG-4. As this part describes the standard for coding more than natural video, it is called MPEG-4 *Visual* instead of MPEG-4 Video.

## Profiles and Levels

MPEG-4 contains a plethora of compression tools that are useful for many different applications. As with MPEG-2, all the tools are not necessary or required for all the applications. Thus MPEG-4 has defined several profiles and levels to define interoperability points where a good compromise is made between implementation complexity and the needs of certain classes of applications. To achieve interoperability, decoders are required to implement all the tools in a given profile. Each profile then has multiple levels. Levels are defined by keeping in mind how much processing power is needed for real time decoding. They are mainly defined according to the sizes of the pictures, such as *QCIF* (176x144), *CIF* (352x288), *HHR* (360x576) and full resolution (720x576). Three main application sets, used for defining profiles related to the coding of natural video, were: (1) Wireless and Wireline Video Phone and Conferencing, (2) Streaming Video and (3) Interactive TV / Web enabled multimedia.

The most commonly known and used profile is the *Simple Profile (SP*). SP was defined for two-way video communication and very low complexity receivers, such as wireless videophones. The tools were selected by giving high priority to low delay, low complexity and error resilience. SP includes very basic coding tools including I-VOPs, P-VOPs, *AC/DC* prediction, *Unrestricted Motion Vectors (UMV*) and error resilience tools such as *Data Partitioning, Slice Resynchronization* and *Reversible VLC.* This profile is a subset of all other video profiles, that is, a decoder compliant to any other video profile is also capable of decoding SP. Due to its simplicity and the lack of any other profile for rectangular video, this also became the most commonly used profile for streaming video. However, the coding efficiency of this profile is low. In addition, levels are defined only up to CIF size pictures.

To make available a profile that is more suitable for streaming video applications over the Internet and has higher coding efficiency, *Advanced Simple Profile (ASP)* was defined [6, 7]. As a delay on the order of hundreds of milliseconds is not an issue for those applications, and targeted platforms have higher processing power, ASP coding tools include B-VOPs, GMC, *Quarter Pixel Interpolation*, and *Interlaced Video* tools (in addition to the SP tools). Streaming video applications generally use only the rectangular video window. Therefore, to control the complexity of implementation, shape-coding tools are not used in ASP. ASP thus provides the highest video coding efficiency (for rectangular video) among all the profiles in Part 2, significantly more than SP. Figure 4 shows a comparison of the performance of ASP with SP, for the case of the Stephan sequence and CIF resolution. The x-axis represents bit rate, and the y-axis represents the *Peak Signal to Noise Ratio (PSNR),* a measure of the distortion (error) in the decoded pictures (High PSNR means better quality.) Reference [8] provides additional

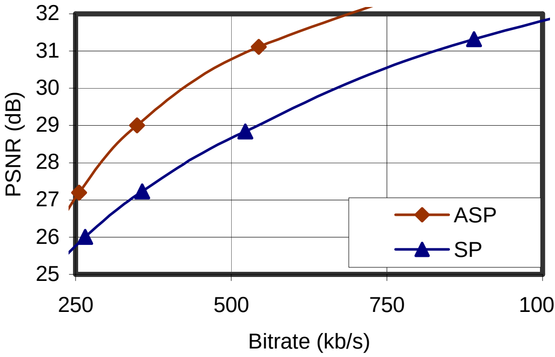

**Figure 4: ASP vs. SP Performance, Stefan sequence (CIF)**

performance data.  In general, ASP provides a good tradeoff between video quality and coding complexity.  In this profile, levels are defined that allow all the way up to full resolution size pictures.  Therefore, this profile is equally applicable to both the PC and TV environments.

The FGS profile supports scalability for applications where bandwidth and processing power vary, such as is typical for Internet distribution to PCs.  This profile allows the use of ASP as a base layer, with scalable FGS extension layers on top that allow improved quality for those receivers that can receive and/or process those extension layers. [7]

To promote the Interactive / Web enabled class of applications, the *Core* and *Main* Profiles are defined.  Core Profile is a superset of SP and Main Profile is a superset of Core.  The Core Profile adds B-VOP and Binary Shape coding tools on top of SP.  The Main Profile adds Gray Scale shape coding, Interlaced Video Coding and Sprite Coding tools. It should be noted that Core and Main Profile do not have Quarter Pixel and GMC coding tools, as they were not developed at that time.

MPEG-4 also defines many other profiles. Reference [3] provides a full description.

## Advanced Video Coding (AVC)

During 2001, the MPEG committee conducted some informal tests to see whether video coding technology had progressed since the development of the MPEG-4 Visual Part 2 standard.  It was concluded that, although there were not fundamentally new coding tools discovered since the completion of Part 2, many coding tools that were rejected during the development of MPEG-4, due to their high implementation complexity, should be reconsidered for inclusion to provide higher coding gain.  Basically, advancements in implementation technology, such as the availability of low cost high-performance microprocessors and high-density high speed integrated circuits and memory, justified inclusion of more complex tools.

In addition, the ITU-T had already started developing a next generation (H.26L) for low bit rate coding.  Tests also showed that the ITU-T's H.26L effort had done a good initial job of coherently assembling a basic structure of coding algorithms with new coding tools. Thus, a joint video team (JVT) was formed with ITU-T, starting from the defined H.26L structure, to develop jointly a new video coding standard to provide significantly higher coding gain, a standard that was not necessarily backward compatible with MPEG-4 Part 2.  The first phase of this standard will be completed by December 2002. Its targeted goal is to provide at least 2 times the coding gain over MPEG-2 for hard-to-compress video sequences.  Once the JVT completes its task, ISO will adopt it as MPEG-4 Part 10 and will call it Advanced Video Coding (AVC), and ITU-T will adopt it most probably as H.264.

At a high level, AVC is also motion compensated block transform based coding. Some of the  video coding tools that are likely to form a part of AVC are: I, P and B frames, *variable block size* motion compensation, a *4x4 integer* (DCT like) transform, *multi-frame* motion compensation, interlaced video coding tools, quarter-pixel interpolation (eighth-pixel in discussion), a *de-blocking (in-loop) filter*, an *adaptive block size transform* (also under discussion), global motion compensation and *global motion vector* coding (also under discussion), *switch-P frames* (SP) to allow switching from one bit stream to another at specific locations, *universal* variable length coding and *context based adaptive binary arithmetic coding*.  AVC has not yet defined any specific profiles and levels, but these are expected in May 2002.

It is expected that this new standard will be capable of sending high-quality movies in the 500 kbps to 750 kbps range, acceptable quality movies in the 250 kbps to 500 kbps range, and high-quality video below 2 Mbps. Examples of the coding efficiency of the MPEG-4 AVC (JVT) standard are provided in Figures 5 and 6. As in Figure 4, x-axes in these figures represent bit rates and y-axes represent PSNRs. Further examples can be found in [8]

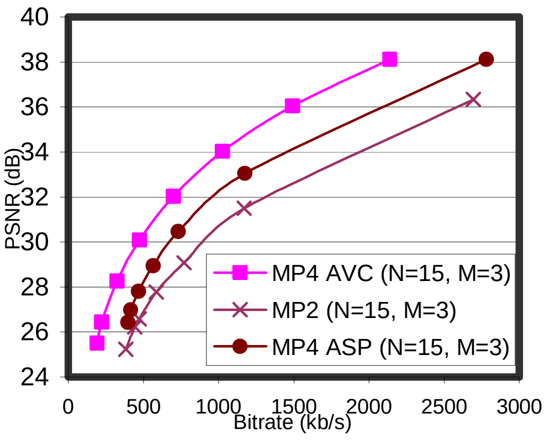

**Figure 5: MPEG-2, MPEG-4 ASP, and MPEG-4 AVC, Bus sequence (CIF)**

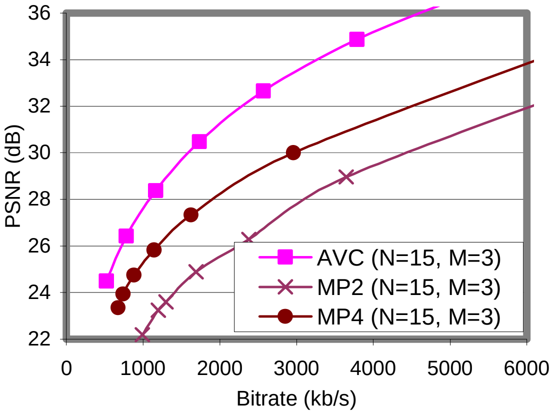

**Figure 6: MPEG-2, MPEG-4 ASP, and AVC, Mobile & Calendar sequence (HHR)**

The impact of a two-to-one efficiency improvement on cable and other forms of broadcast distribution is clearly one of capacity. More choices, more opportunities to narrowcast content, and better HDTV carriage will all be beneficial to the consumer. For IP carriage, where last mile bandwidth and quality of service guarantees are still major concerns, the efficiency gain may spell the difference between carriage of acceptable quality content at acceptable cost, and the more typical postage stamp sized Internet images. Furthermore, less video bandwidth consumption allows a set of new interactive services; such services can now supply additional streams of text, scene composition, graphics, etc., all under control of the overall application.

## Other MPEG-4 Media Types

MPEG-4 extends AAC audio with several new tools, broadening its range of application. Speech coding is supported through a *twin vector quantization* algorithm, and a *CELP* (Code Excited Linear Prediction) algorithm, and text-to-speech conversion is defined. Synthetic audio can be supported through the *FM wavetable* or *model-based* synthesis algorithms, also including MIDI formats, essentially providing an enhanced “score.”

MPEG-4 supports face and body animation through a set of parameters that can be updated over time. As an example, avatars can be represented with very low bandwidth streams describing the animation changes and the accompanying coded speech.

Still images can be coded in MPEG-4 with the *zerotree wavelet* algorithm, and text can be coded in Unicode, with a font and color, for example. Rolling movie credits would be best handled in this fashion. MPEG-4 graphics objects are compressed (*geometry compression*) as a two-dimensional or threedimensional mesh, with texture mapping. Figure 7 provides an example 3D mesh. Vertex positions, attributes, and connectivity (static or dynamic) would be coded as the geometry representation.

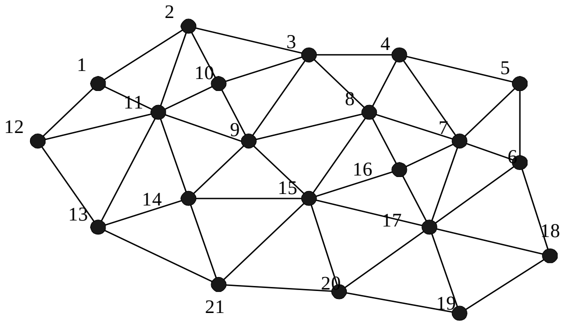

Figure 7: MPEG-4 Mesh

A full description of MPEG-4 media types can be found in [3].

MPEG-4 Systems

MPEG-4 systems provides various options for combining objects of different media types, or multiple objects of the same media type, into higher level structures, as well as techniques for combining and carrying multiple services. Figure 8 shows an example scene and Figure 9 its decomposition, which can be represented with BIFS. Timestamps can be supported through a *sync layer*, and services can be premultiplexed using a *flexmux*. MPEG-4 structures such as these can be carried in MPEG-2 transport, or directly in IP protocols [9]. See reference [3] for a complete description of MPEG-4 systems.

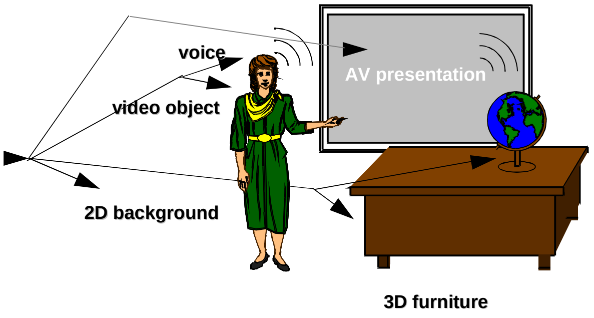

Figure 8: Multimedia Scene Example

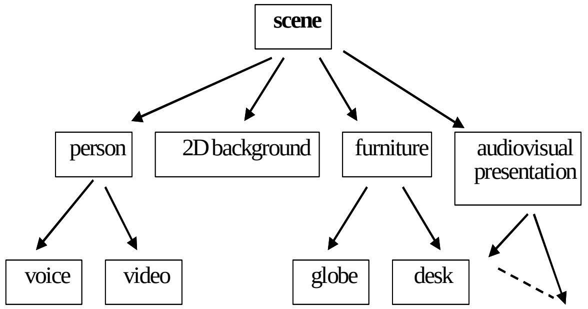

Figure 9: Scene Decomposition of Figure 8

MPEG-7 AND MPEG-21

MPEG-7 and MPEG-21 were not begun as efforts to find yet another new generation of coding techniques. Rather, when one surveys the broad landscape of multimedia applications, there are areas not covered by the earlier MPEG standards.

Search and retrieval of media objects is an important area for future growth of services. Images and sounds are proliferating every day on the Internet, and in the home, and the text-oriented search engines now available can only locate such content if an author has provided the right textual key words or titles. Longer term, users need to be able to find objects through their intrinsic multimedia attributes.

MPEG-7 standardizes such attributes to enable a new class of search engines, similar to the way earlier MPEG standards describe syntax decoding. Thus MPEG-7 does not define how one extracts such features from an object, nor does it define search engine technologies such as pattern recognition. This is the province of the creative process, such as in MPEG video *encoding*.

Examples of the over 100 features defined in MPEG-7 include dominant color or texture, color histograms, thumbnails, shape or edge description (well suited to trademark searches), and various motion oriented attributes (well suited to video surveillance). For audio, examples include waveform and spectral envelope, which can assist in finding similar voices; spoken word features, such as speech recognition employs; timbre description; and other features that might help locate content based upon a whistled or hummed portion. MPEG-7 also includes more basic aspects of content, such as content author, or owner, access rules, storage format, and so forth.

Structurally, features are defined by *descriptors* with defined syntax and semantics. *Description schemes*, expressed in XML schema, also provide relationships between these descriptors.

MPEG-21 began in 2000, and is still in its earlier stages. This work addresses the overall multimedia framework, filling in any other missing elements for providing a complete systems approach to creating, storing, sending, and using multimedia content. One key area being addressed by MPEG-21 currently is the definition of a digital rights description to support standardized access to content. This work will produce a *Rights Expression Language*, and a *Rights Data Dictionary*.

## SUMMARY

The nearly 14-year MPEG process has addressed a wide range of multimedia signal processing and systems issues. From its inception as an audio/video compression standard for storage of content, through its evolution to its current overarching support of multimedia content generation, transmission, storage, and consumption, MPEG has succeeded in producing useful, successful, standards. As the earlier MPEG-2 standard has been widely deployed in cable systems and other broadcast distribution networks, the newer MPEG standards can be expected to support new classes of (revenue generating) services and applications for those industries.

## REFERENCES

[1] MPEG-1, ISO/IEC 11172: Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1,5 Mbit/s.

[2] MPEG-2, ISO/IEC 13818: Information Technology - Generic Coding of Moving Pictures and Associated Audio Information.

[3] MPEG-4, ISO/IEC 14496: Information Technology - Coding of Audio-visual Objects.

[4] MPEG-7, ISO/IEC 15938: Information Technology - Multimedia Content Description Interface.

[5] MPEG-21, ISO/IEC TR 21000: Information Technology - Multimedia Framework.

[6] A. Luthra, “Need for Simple Streaming Video Profile,” ISO/IEC JTC1/SC29/WG11 M5800 Noordwijkerhout, March 2000.

[7] ISO/IEC 14496-2:1999 / Final Draft Amendment (FDAM4), WG-11 N3904, January 2001.

[8] A. Luthra, “MPEG-4 Video Coding Standard – An Overview,” SCTE 2002 Conference on Emerging Technologies, January 2002.

[9] "RTP Payload Formats for Carriage of MPEG-4 Content over IP Networks," ISO/IEC JTC 1/SC 29/WG 11 N4428, December 2001.

## ACKNOWLEDGEMENTS

This paper is in part based on the work done by the video coding team in the Advanced Technology department of Motorola Broadband Communications, and of course on the work done by the entire MPEG committee. Our thanks go out to all these engineers, without whose commitment and expertise the MPEG process would never have produced such impressive successes.

## CONTACT INFORMATION

Dr. Paul Moroney, Dr. Ajay Luthra
Motorola Broadband
6450 Sequence Dr
San Diego, CA 92121