AN OVERVIEW OF MPEG-4 AUDIO VERSION 2

HEIKO PURNHAGEN

Laboratorium für Informationstechnologie
University of Hannover, Hannover, Germany
purnhage@tnt.uni-hannover.de

The MPEG-4 Audio Standard provides audio and speech coding for natural and synthetic content at bitrates ranging from 2 to 64 kbit/s and above. While the first version of MPEG-4 Audio was finalised in 1998, work continues for Version 2, complementing MPEG-4 Audio by the following new tools: Error Resilience, Low-Delay Audio Coding, Small Step Scalability, Parametric Audio Coding, and Environmental Spatialisation.

INTRODUCTION

In the context of evolving multimedia applications – like digital broadcasting, storage, realtime communication, the World Wide Web, or games – new demands for efficient and flexible representation of audiovisual content arise. Besides high coding efficiency required to cope with the limited bandwidth of the Internet or in mobile communication, also new functionalities like flexible access to coded data and manipulation by the recipient are desired. To address these requirements and develop interoperable solutions, ISO/IEC started its MPEG-4 standardisation activities "Coding of audiovisual objects."

The first version of the MPEG-4 Audio Standard was finalised in 1998 and provides tools for coding of natural and synthetic audio objects and composition of such objects into an "audio scene" [1, 2]. Natural audio objects (such as speech and music) can be coded at bitrates ranging from 2 kbit/s to 64 kbit/s and above using Parametric Speech Coding, CELP-based Speech Coding or transform-based General Audio Coding. The natural audio coding tools also support bitrate scalability. Synthetic audio objects can be represented using a Text-To-Speech Interface or the Structured Audio synthesis tools. These tools are also used to add effects, like echo, and mix different audio objects to compose the final "audio scene" that is presented to the listener.

Because of the very tight schedule of the MPEG-4 standardisation, several promising tools proposed for MPEG-4 were not mature enough to be included in the first version of the standard. Since many of these tools provide desirable functionalities not available in MPEG-4 Version 1, it was decided to continue the work on these tools for an extension of the standard, MPEG-4 Version 2.

With this extension, new tools are added to the MPEG-4 Standard, while none of the existing tools of Version 1 is replaced. Version 2 is therefore fully backward compatible to Version 1, as depicted in Figure 1.

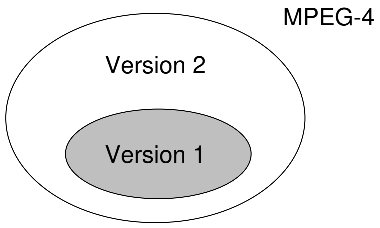

Figure 1: Relation between MPEG-4 Versions (after [1]).

In the area of Audio, new tools are added in MPEG-4 Version 2 to provide the following new functionalities:

- *Error Resilience* tools provide improved performance on error-prone transmission channels.
- *Low-Delay Audio Coding* tools support the transmission of general audio signals in applications requiring low coding delay, such as realtime bi-directional communication.
- *Small Step Scalability* tools provide scalable coding with very fine granularity, i.e. embedded coding with

very small bitrate steps, based on the General Audio Coding tools of Version 1.

- *Parametric Audio Coding* tools combine very low bitrate coding of general audio signals with the possibility of modifying the playback speed or pitch during decoding without the need for an effects processing unit.
- *Environmental Spatialisation* tools enable composition of an "audio scene" with more natural sound source and sound environment modeling than is possible in Version 1.

In the Systems part of MPEG-4, Version 2 specifies a file format to store MPEG-4 encoded content. Besides other new tools, also a backchannel for dynamic control and interaction with a server is specified.

In this paper, first a brief overview of the MPEG-4 Audio Standard and the audio tools available in Version 1 is given. Then the audio tools added in Version 2 are introduced. These tools are described in more detail and their status in the ongoing standardisation process is discussed. Finally an outlook on future developments is presented and conclusions are drawn.

1. THE MPEG-4 AUDIO STANDARD VERSION 1

In audio coding, a variety of source models can be utilised in combination with appropriate models of the human perception to reduce the redundancy and irrelevance contained in the audio signal. Figure 2 depicts the general block diagram of an audio encoder and decoder derived from these considerations.

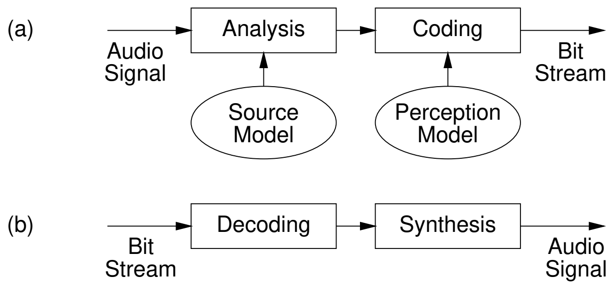

Figure 2: Block diagram of an audio encoder (a) and decoder (b).

The MPEG-4 Audio Standard permits coding of natural audio signals at bitrates ranging from 2 kbit/s up to 64 kbit/s and above. To obtain optimal coder performance over this broad range of bitrates, it is advantageous to utilise different model assumptions for different target bitrates [3]. Also the characteristics of the audio material being coded must be considered, since e.g. specialised speech coders offer better coder performance for speech signals at a given bitrate than coders designed for general audio signals. Because of these considerations, MPEG-4 Audio includes speech coding techniques and general audio coding techniques which are integrated in a common framework.

Besides data compression, MPEG-4 Audio provides further so-called "functionalities":

- *Bitrate scalability*, often also referred to as embedded coding, uses a base layer bitstream and one or more additional enhancement layer bitstreams. If only the base layer is available to the decoder, it still can be decoded into a meaningful signal. If one ore more enhancement bitstreams are also available to the decoder, a signal with better quality can be obtained.
- *Speed and pitch modification* under user control during playback is provided by some of the coding techniques in MPEG-4 Audio.

Besides coding of natural sound (like speech and music), MPEG-4 Audio also provides techniques to synthesise sound based on efficient so-called "structured" representations. Text representations for Text-To-Speech synthesisers are supported as well as musical score or MIDI representations. Music synthesis can be based on so-called "instrument descriptions". These tools can also be used to provide effects, like reverberation, and to mix different independently coded audio objects to compose a complete "audio scene" as output signal of an MPEG-4 Audio decoder.

An overview of the complete MPEG-4 Standard – including the Video and Systems parts – is given in [1, 4]. Further information is also available from the MPEG and MPEG Audio web pages [5, 6].

In the following subsections the different audio tools available in Version 1 are described.

1.1. Speech Coding

Speech coding at bitrates between 2 and 24 kbit/s is supported by Harmonic Vector eXcitation Coding (HVXC) for a bitrate of 2 to 4 kbit/s, and Code Excited Linear Predictive (CELP) coding for a bitrate of 4 to 24 kbit/s [7, 8]. In addition, HVXC can operate down to an average of around 1.2 kbit/s in its variable bitrate mode. In CELP coding, two sampling rates, 8 and 16 kHz, are used to support narrowband and wideband speech.

Both HVXC and CELP speech coding provide bitrate scalability. Since the HVXC is a parametric speech coder, it provides the functionality of speed and pitch modification in the decoder.

1.2. General Audio Coding

Coding of general audio ranging from very low bitrates up to high quality is provided by transform coding techniques [9, 10]. These tools cover a wide range of bitrates and bandwidths. Starting at a bitrate of 6 kbit/s and a bandwidth below 4 kHz, also broadcast quality audio from mono up to multichannel is covered. For very low bitrates up to 16 kbit/s, the Transform-domain Weighted INterleaved Vector Quantisation (TwinVQ) tool is used. For higher bitrates an extended version of the MPEG-2 Advanced Audio Coding (AAC) technology is used. It is backward compatible with the MPEG-2 AAC and includes the new modules Long Term Prediction (LTP) and Perceptual Noise Substitution (PNS).

Both TwinVQ and AAC provide bitrate scalability. The scalable AAC scheme allows to use not only transform coding but also CELP coding for the base layer bitstream.

1.3. Structured Audio

The Structured Audio tools convert a structured representation into a synthetic sound signal [11, 12, 13]. The decoding is described by a special synthesis language called Structured Audio Orchestra Language (SAOL). This language is used to define an "orchestra" made up of "instruments" (downloaded in the bitstream, not fixed in the terminal) which create and process control data. An instrument is a small network of signal processing primitives that might emulate some specific sounds such as those of a natural acoustic instrument. The signal-processing network includes both generation and processing of sounds and manipulation of pre-stored sounds.

Control of the synthesis is accomplished by transmitting "scores" in the bitstream, which invokes or controls various instruments. The score description can be encoded in the Structured Audio Score Language (SASL) or transmitted as MIDI data.

The Structured Audio tools are also used to apply simple effects processing – such as filters, reverbs, and chorus effects – to decoded natural and synthetic audio objects and to compose them to build the "audio scene" presented to the listener.

1.4. Text-To-Speech

Text-To-Speech (TTS) synthesis allows to generate synthetic speech from a text or a text with prosodic parameters (pitch contour, phoneme duration, and so on) conveyed at bitrates ranging from 200 bit/s to 1.2 kbit/s. Parameters that allow synchronization to associated face animation, international languages for text, and international symbols for phonemes are supported. It should be noted that MPEG-4 provides a standardised Text-To-Speech Interface (TTSI) for the operation of a Text-To-Speech decoder, but not a normative TTS synthesizer itself.

1.5. Profiles and Levels

MPEG-4 provides a large and rich set of tools for the coding of audio objects. In order to allow effective implementations of the standard, subsets of the tool set have been identified, that can be used for specific applications. These subsets, called "Profiles", limit the tool set a conforming decoder has to implement. For each of these Profiles, one or more Levels have been specified, restricting the computational complexity. Four Audio Profiles have been defined in Version 1:

- The *Speech Profile* provides HVXC for very-low bitrate parametric speech coding, a CELP narrowband/wideband speech coder, and a Text-To-Speech Interface.
- The *Scalable Profile*, a superset of the Speech Profile, includes General Audio Coding and is suitable for scalable coding of speech and music for networks, such as Internet and Narrowband Audio DIgital Broadcasting (NADIB).
- The *Synthesis Profile* provides the Structured Audio tools for score driven synthesis using SAOL and wavetables, and a Text-to-Speech Interface to generate sound and speech using very low bitrates.
- The *Main Profile* is a rich superset of the other three Profiles, containing tools for natural and synthetic audio.

2. NEW TOOLS OF MPEG-4 AUDIO VERSION 2

By extending the MPEG-4 Standard to Version 2, several new functionalities will become available in the MPEG-4 framework. This extension – formally referred to as "Amendment 1" – was promoted to "Committee Draft" status at the March 1999 MPEG meeting [14, 15]. With most of the technical work being completed, the new audio tools of Version 2 will now undergo subjective verification tests. Version 2 of MPEG-4 is scheduled to be finalised in December 1999.

In this section, the Version 2 tools as specified in the Committee Draft [14, 15] are described. Additionally, tools that are currently in the so-called "Core Experiment Check Phase" and probably will be included in Version 2 are described here as well. It should be noted that the set of tools included in Version 2 is not yet finally decided.

2.1. Error Resilience

The Error Resilience tools provide improved performance on error-prone transmission channels. There are two classes of tools:

- The first class contains algorithms to improve the *error robustness* of the source coding itself, e.g. Huffman codeword reordering for AAC.
- The second class consists of general tools for *error protection*, providing unequal error protection for the MPEG-4 audio coding schemes.

Improved *Error Robustness* for the AAC is provided by a set of tools belonging to the first class of Error Resilience tools. These tools reduce the perceived deterioration of the decoded audio signal that is caused by corrupted bits in the bitstream. The following three tools are provided to improve the error robustness for the different parts of an AAC bitstream frame:

- The *Virtual CodeBooks* tool (VCB11) extends the sectioning information of an AAC bitstream. This permits to detect serious errors within the spectral data of an MPEG-4 AAC bitstream. Virtual codebooks are used to limit the largest absolute value possible within a certain scalefactor band where escape values are allowed, i.e. where the so-called "codebook 11" is used. While referring to the same codes as "codebook 11", the 16 virtual codebooks introduced by this tool provide 16 different limitations of the spectral values belonging to the corresponding section. Due to this, errors within spectral data resulting in spectral values exceeding the indicated limit can be located and appropriately concealed.
- The *Reversible Variable Length Coding* tool (RVLC) replaces the Huffman and DPCM coding of the scalefactors in an AAC bitstream. The RVLC uses symmetric codewords to enable both forward and backward decoding of the scalefactor data. In order to have a starting point for backward decoding, the total number of bits of the RVLC part of the bitstream is transmitted. Because of the DPCM coding of the scalefactors, also the value of the last scalefactor is transmitted to enable backward DPCM decoding. Since not all nodes of the RVLC code tree are used as codewords, some error detection is also possible.
- The *Huffman Codeword Reordering* tool (HCR) extends the Huffman coding of spectral data in an MPEG-4 AAC bitstream. By placing some of the Huffman codewords at known positions, error propagation into these so-called "priority codewords" (PCW) can be avoided. To implement this technique, segments of known length are defined and the PCWs are placed at the beginning of these segments. The remaining (non-priority) codewords are filled into the gaps left by the PCWs using a special algorithm that minimises the effect of error propagation for the non-priority codewords. This reordering algorithm does not increase the size of spectral data. Before the reordering algorithm itself is applied, a pre-sorting process is used to sort all codewords according to their importance and to determine the PCWs.

The *Error Protection* tool (EP tool) provides Unequal Error Protection (UEP) for MPEG-4 Audio and belongs to the second class of Error Resilience tools.

UEP is an efficient method to improve the error robustness of source coding schemes. It is used by various speech and audio coding systems operating over error-prone channels such as mobile telephone networks or Digital Audio Broadcasting (DAB). The bits of the coded signal representation are first grouped into different classes according to their error sensitivity. Then error protection is individually applied to the different classes, giving better protection to more sensitive bits.

To group the bits into different error sensitivity classes, bitstream reordering is necessary. This reordering is specified for the different source coding tools in MPEG-4 Audio, using 4 or 5 error sensitivity classes.

Both Forward Error Correction (FEC) codes and Cyclic Redundancy Check (CRC) codes for error detection can be applied. To accommodate a wide range of transmission channel conditions, the EP tool permits flexible configuration of the error correction and/or error detection capabilities and the redundancy hereby added to the bitstream. Thus the redundancy required to provide the desired error protection can be minimised.

Most of the EP tool configuration is transmitted "out-of-band", e.g. in a bitstream header. However, for some source coding schemes (like AAC), the structure and size of the error sensitivity classes can vary from frame to frame. This is accommodated by transmitting the required configuration "in-band", i.e. in the bitstream frames.

The block diagram of the complete error protection encoder is depicted in Figure 3. First a CRC of up to 32 bits is added to the data bits of each class to permit error detection. Then a Systematic Rate-Compatible Punctured Convolutional (SRCPC) code is applied to provide forward error correction. The puncturing procedure allows fine adjustment of the code rate, i.e. the redundancy added by the code. The CRC and SRCPC coded data bits of the different error sensitivity classes as well as the "in-band" part of the EP tool configuration is then processed by an interleaver to improve the robustness to burst errors. Finally the interleaved data is protected with a Shortened Reed-Solomon (SRS) code. The SRCPC and SRS codes utilised here are very similar to those described in the Annex of H.223 [16].

While the Error Resilience tools and their bitstream format are specified in the standard, it is up to the decoder to decide how much of the error resilience information is actually utilised. In the simplest case, the decoder just strips off the redundancy added by the EP tools and undoes the interleaving and bitstream reordering. However, to make full use of an error resilient bitstream, a more complex decoder is required, which can include a Viterbi decoder for the SRCPC code and advanced error concealment techniques for errors detected by the CRC.

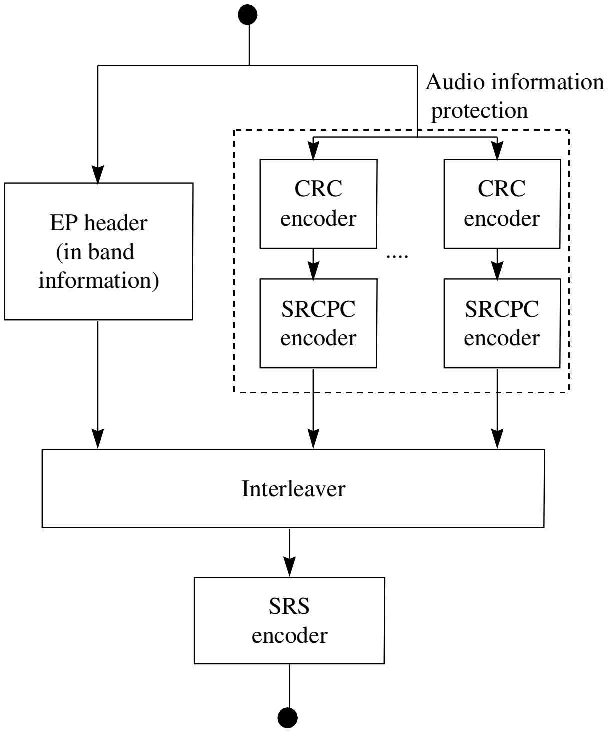

Figure 3: Block diagram of the Error Protection Encoder (after [14]).

2.2. Low-Delay Audio Coding

The MPEG-4 General Audio Coder provides very efficient coding of general audio signals at low bitrates. However it has an algorithmic delay of up to a few 100 ms and is thus not well suited for applications requiring low coding delay, such as realtime bi-directional communication.

The algorithmic delay of the General Audio Coder is determined by the following factors:

- The *frame length* influences delay of any block-based processing.
- The *filterbank delay* is caused by the analysis and synthesis filterbank.
- The *look-ahead* is necessary for the "block-switching" decision that improves coding of transient signal parts.
- The *bit reservoir* facilitates the use of a locally varying bitrate, but implies an additional delay.

For the General Audio Coder operating at 24 kHz sampling rate and 24 kbit/s, this results in an algorithmic coding delay of about 110 ms plus up to additional 210 ms for the bit reservoir.

To enable coding of general audio signals with an algorithmic delay not exceeding 20 ms, MPEG-4 Version 2 specifies a Low-Delay Audio Coder [17, 18], which is derived from the General Audio Coder of Version 1. It operates at 48 kHz sampling rate and uses a frame length of 512 or 480 samples, compared to the 1024 or 960 samples normally used. Also the size of the window used in the analysis and synthesis filterbank is halved.

No block switching is used to avoid the "look-ahead" delay that would be caused by the block switching decision. To reduce pre-echo artefacts in case of transient signals, window shape switching is provided instead. For non-transient parts of the signal the normal sine window is used, while a so-called zero-padded (ZP) window is used in case of transient signals. Both windows are depicted in Figure 4.

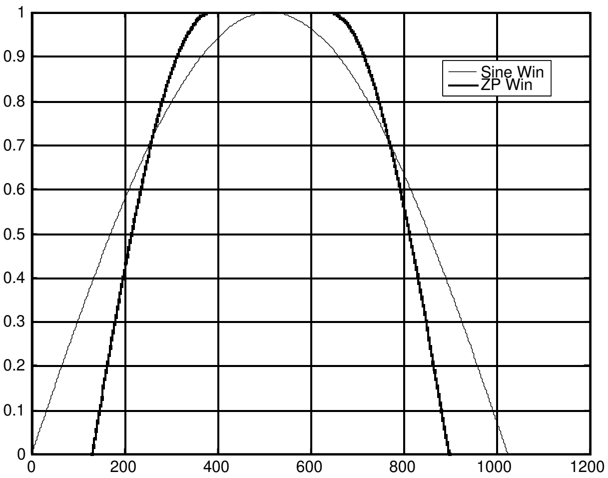

Figure 4: Sine and Zero-Padded (ZP) window for Low-Delay Audio Coding (window length: 1024 samples) (after [18]).

Use of the bit reservoir is minimised in the encoder in order to reach the desired target delay. As one extreme case, no bit reservoir is used at all.

Figure 5 shows the overall block diagram of the Low-Delay Audio Decoder. It consists of the Low Complexity AAC tools including Temporal Noise Shaping (TNS), combined with the Perceptual Noise Substitution (PNS) and the Long Term Predictor (LTP) tools from Version 1. LTP is implemented by a delay buffer and utilises a Frequency Selective Switch (FSS). Compared to the General Audio Coder of Version 1, the coding efficiency of the Low-Delay Audio Coder is slightly reduced due to its lower algorithmic delay.

2.3. Small Step Scalability

Bitrate scalability, also known as embedded coding, is a very desirable functionality. The General Audio Coder of Version 1 supports large step scalability where a base layer bitstream can be combined with one or more enhancement layer bitstreams to utilise a higher bitrate and

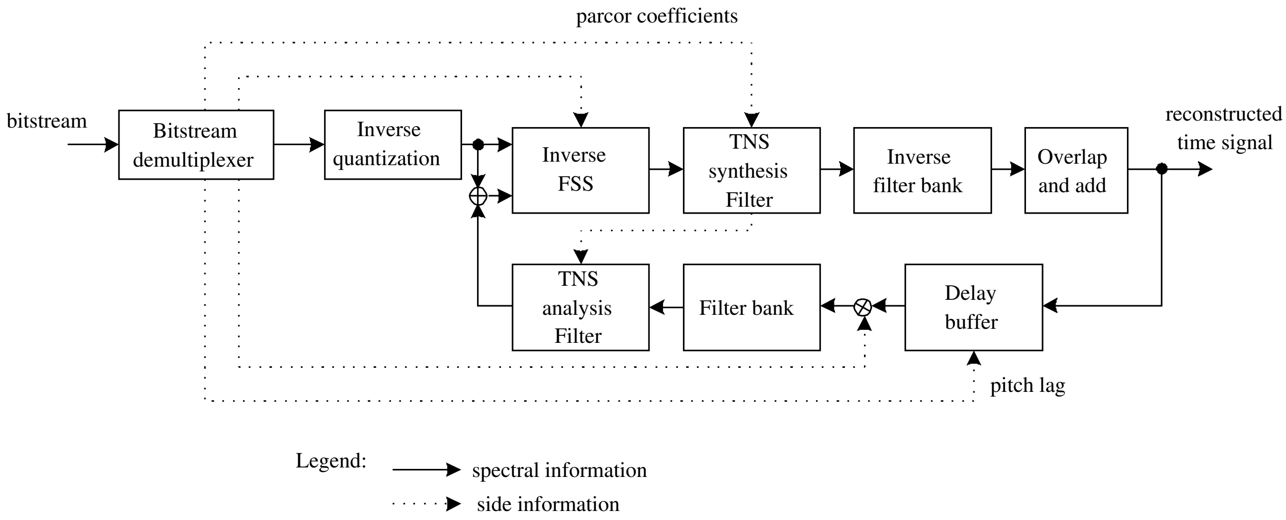

Figure 5: Block diagram of the Low-Delay Audio Decoder (after [14]).

thus obtain a better audio quality. In a typical configuration, a 24 kbit/s base layer and two 16 kbit/s enhancement layers could be used, permitting decoding at a total bitrate of 24 kbit/s (mono), 40 kbit/s (stereo), and 56 kbit/s (stereo). Due to the side information carried in each layer, small bitrate enhancement layers are not efficiently supported in Version 1.

To address this problem and to provide efficient small step scalability for the General Audio Coder, the Bit-Sliced Arithmetic Coding (BSAC) tool is available in Version 2 [19]. This tool is used in combination with the AAC coding tools and replaces the noiseless coding of the quantised spectral data and the scalefactors..

BSAC provides scalability in steps of 1 kbit/s per audio channel, i.e. 2 kbit/s steps for a stereo signal. One base layer bitstream and many small enhancement layer bitstreams are used. The base layer contains the general side information, specific side information for the first layer and the audio data of the first layer. The enhancement streams contain only the specific side information and audio data for the corresponding layer.

To obtain fine step scalability, a bit-slicing scheme is applied to the quantised spectral data. First the quantised spectral values are grouped into frequency bands. Each of these groups contains the quantised spectral values in their binary representation. Then the bits of a group are processed in slices according to their significance. Thus first all most significant bits (MSB) of the quantised values in a group are processed, etc.

These bit-slices are then encoded using an arithmetic coding scheme to obtain entropy coding with minimal redundancy. Various arithmetic coding models are provided to cover the different statistics of the bit-slices. The scheme used to assign the bit-slices of the different frequency bands to the enhancement layer is constructed in a special way. This ensures that, with an increasing number of enhancement layers utilised by the decoder, the quantisation of the spectral values is refined by providing more of the less significant bits. But also the bandwidth is increased by providing bit-slices of the spectral data in higher frequency bands.

2.4. Parametric Audio Coding

The Parametric Audio Coding tools combine very low bitrate coding of general audio signals with the possibility of modifying the playback speed or pitch during decoding without the need for an effects processing unit. In combination with the speech and audio coding tools of Version 1, improved overall coding efficiency is expected for applications of object based coding allowing selection and/or switching between different coding techniques.

MPEG-4 Parametric Audio Coding uses the Harmonic and Individual Line plus Noise (HILN) technique to code general audio signals at bitrates of 4 kbit/s and above using a parametric representation of the audio signal [20, 21]. The basic idea of this technique is to decompose the input signal into audio objects which are described by appropriate source models and represented by model parameters. Object models for sinusoids, harmonic tones, and noise are utilised in the HILN coder.

This approach allows to introduce a more advanced source model than just assuming a stationary signal for the duration T of a frame, which motivates the spectral decomposition used e.g. in the MPEG-4 General Audio Coder. As known from speech coding, where specialised source models based on the speech generation process in the human vocal tract are applied, advanced source models can be advantageous in particular for very low bitrate coding schemes.

In Figure 6 the block diagram of the HILN Parametric Audio Encoder is depicted. First the input signal is decomposed into different objects and then the model parameters for the appropriate source models are estimated.

- An *individual sinusoid* is described by its frequency and amplitude.
- A *harmonic tone* is described by its fundamental frequency, amplitude, and the spectral envelope of its partials.
- A *noise* signal is described by its amplitude and spectral envelope.

Due to the very low target bitrates, only the parameters for a small number of objects can be transmitted. Therefore a perception model is employed to select those objects that are most important for the perceptual quality of the signal.

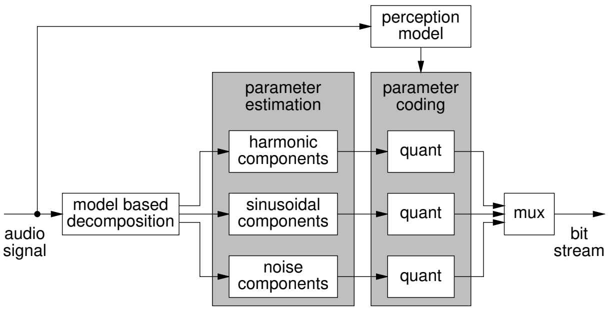

Figure 6: Block diagram of the HILN Parametric Audio Encoder.

Recently, the quality of the HILN was significantly improved by an advanced parameter representation, quantisation, and coding [22, 23]. The frequency and amplitude parameters are quantised according to the "just noticeable differences" known from psychoacoustics. The spectral envelope of the noise and the harmonic tone is described using LPC modeling as known from speech coding. Correlation between the parameters of one frame and between consecutive frames is exploited by parameter prediction. The quantised parameters are finally entropy coded and multiplexed to form a bitstream.

The signal decomposition and the parameter estimation form the core of the HILN encoder. A more detailed block diagram of this process using an analysis/synthesis approach is shown in Figure 7.

The block diagram of the HILN Parametric Audio Decoder is shown in Figure 8. First the parameters of the objects are decoded and then the object signals are resynthesised according to the transmitted parameters. By combining these object signals, the output signal of the

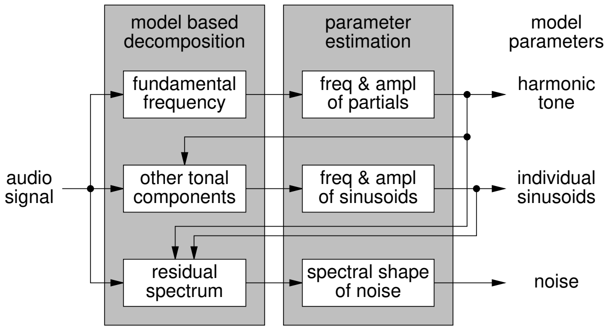

Figure 7: Signal decomposition and parameter estimation for the HILN utilising the object models *harmonic tone*, *individual sinusoid*, and *noise*.

HILN decoder is obtained. Because of the low phase sensitivity of the human ear, phase information for sinusoids is usually not transmitted.

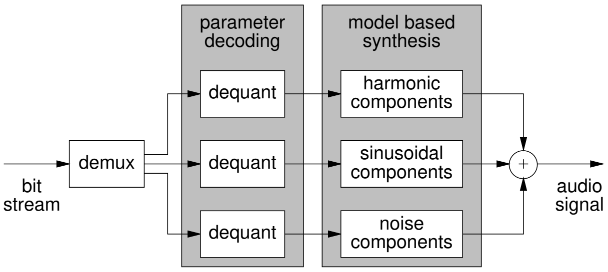

Figure 8: Block diagram of the HILN Parametric Audio Decoder.

A very interesting property of this parametric coding scheme arises from the fact that the signal is described in terms of frequency and amplitude parameters. This signal representation permits speed and pitch change functionality by simple parameter modification in the decoder.

The HILN Parametric Audio Coder can be combined with MPEG-4 Parametric Speech Coder (HVXC) to form an integrated parametric coder covering a wider range of signals and bitrates [24, 25]. This integrated coder is depicted in Figure 9 and supports speed and pitch change. Using a speech/music classification tool in the encoder, it is possible to automatically select the HVXC for speech signals and the HILN for music signals. Such automatic HVXC/HILN switching was successfully demonstrated [24] and the classification tool is described in the informative Annex of the Version 2 standard.

Since the bitrate as well as the audio quality depend on the number of object parameters that can be transmitted in the bitstream, scalability can be implemented by transmitting the most important objects in a base layer bitstream and further, less important objects in one or more enhancement bitstreams [26]. The specification of such HILN enhancement bitstreams is currently under

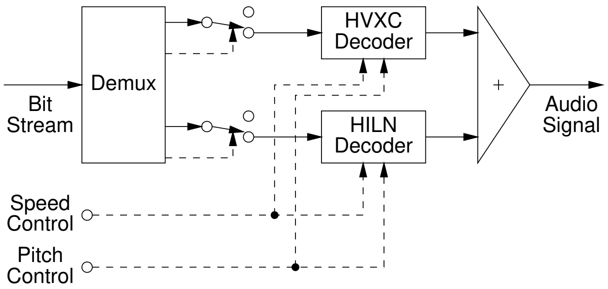

Figure 9: Integrated parametric decoder combining HVXC and HILN.

discussion and might be included in the standard.

To be able to use the HILN as so-called "core coder" for the large step scalable General Audio Coder of Version 1, it is possible to transmit an additional HILN bitstream carrying the phase information for the sinusoidal objects and thus permit to calculate a residual signal in the time domain, which is then coded by the scalable General Audio Coder [25].

Although it is possible to apply an additional time envelope to some or all of the HILN sinusoidal objects in order to improve the modelling of transient signals, recently promising results were presented for a coding scheme that combines parametric coding for stationary parts of the signal with transform coding for the transient parts of the signal [27]. Therefore it is proposed to permit simple switching between the HILN coder (a parametric coder) and the General Audio Coder (a transform coder) as depicted in Figure 10 [28]. Since the transform coder is only used during short, transient regions of the signal, speed and pitch change is still available in the switched coder. This proposal is currently under discussion and might be included in the standard.

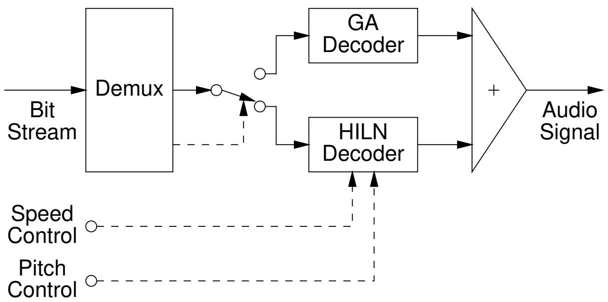

Figure 10: Switched parametric / transform decoder using HILN and General Audio (GA) Coding.

2.5. Environmental Spatialisation

The Environmental Spatialisation tools enable composition of an "audio scene" with more natural sound source and sound environment modeling than is possible in Version 1. Both, a physical and a perceptual approach to spatialisation are supported.

- The *physical approach* is based on a description of the acoustical properties of the environment (e.g. room geometry, material properties, position of sound source) and can be used in applications like 3-D virtual reality [29].
- The *perceptual approach* on the other hand permits a high level perceptual description of the "audio scene" based on parameters similar to those of a reverberation effects unit. Thus, the audio and the visual scene can be composed independently as usually required by applications like movies [30].

Although these tools are related to audio, they belong to the BInary Format for Scene description (BIFS), which is a part of MPEG-4 Systems [15, 31, 32]. The Environmental Spatialisation tools are also referred to as Version 2 Advanced AudioBIFS.

The *physical approach* allows to describe the acoustical properties of an MPEG-4 scene (e.g. a 3-D model of a furnished room or a concert hall) created with the BIFS scene description tools. Such properties are, for example, room reverberation time, speed of sound, boundary material properties (reflection, transmission), and sound source directivity. The new sound source description permits modeling of air absorption and more natural distance dependent attenuation, as well as sound source directivity modeling. With this description of the acoustical properties of the scene, the sound can be rendered so that it corresponds to the visual parts of the scene. For example, when the listener in a scene moves from a very small room to a larger hall, the change in the acoustical and graphical rendering is immediately perceived. New functionality made possible with these scene description parameters includes advanced and immersive audiovisual rendering, detailed room acoustical modeling, and enhanced 3-D sound presentation.

This physical and geometrical description of the Audio Scene is designed to allow:

- dynamic modeling of room acoustics that correspond and relate to a visual scene
- sound source directivity modeling
- tracking and rendering sound reflections according to geometry and positions
- generating occlusion effects automatically when walls or obstacles are present between the source and the listener
- modeling reverberation, air absorption, distance attenuation, and Doppler effect.

In the *perceptual approach*, the sound transformation associated with room reflections and reverberation is described by a set of perceptual attributes (such as source presence and brilliance, room reverberance, envelopment and so forth). These attributes may be manipulated directly and individually for each sound source in the scene.

This approach provides simple and intuitive parameters to the content author, allowing:

- the manipulation of environmental effects for each sound event directly (without requiring that the source or the point of view be moved)
- sound design adjustments beyond the physical constraints implied by the graphic representation, for example:
  - distorted or exaggerated distance sensation and room-related effects
  - unconstrained spatial sound effects for audio-only scenes (no visual correspondence) or when the point of view is out of the room.

Based on the set of perceptual attributes, the parameters of a general reverberation response model are calculated. In Figure 11, a typical room response described by the reverberation model is shown.

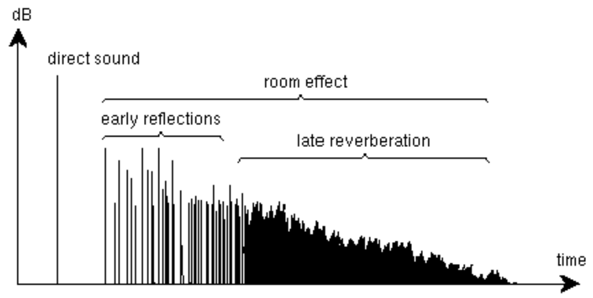

Figure 11: Typical room impulse response (after [32]).

While the *physical approach* is part of MPEG-4 Version 2, the *perceptual approach* was proposed only recently and is therefore still under discussion – but will likely also be included in Version 2.

2.6. CELP Silence Compression

In order to reduce the bitrate required by the MPEG-4 CELP speech coder in situations without voice activity, a silence compression tool for Version 2 was proposed [33]. This tool is currently under discussion and might be included in the standard.

In the encoder, a voice activity detector is used to distinguish between regions with normal speech activity and those with silence or background noise. During normal speech activity, the CELP coding as in Version 1 is used. Otherwise a Silence Insertion Descriptor (SID) is transmitted. This SID enables a Comfort Noise Generator (CNG) in the decoder. The amplitude and spectral shape of this comfort noise is specified by energy and LPC parameters similar as in a normal CELP frame. These parameters are an optional part of the SID and thus can be updated as required. A block diagram of a CELP decoder with the proposed Silence Compression tool is shown in Figure 12.

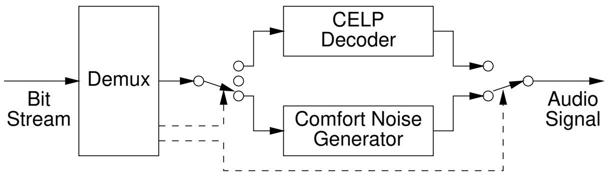

Figure 12: Block diagram of the CELP decoder with the Silence Compression tool.

2.7. Profiles and Levels in Version 2

The specification of Audio Profiles in MPEG-4 Version 2 is still under development. The current status of this work is reflected by the following proposal:

- The *Low Delay Audio Profile* provides speech and audio coding for applications demanding a low coding delay, like bi-directional communication.
- The *Scalable Internet Audio Profile* extends the “Scalable Profile” of Version 1 by Small Step Scalability and Parametric Audio Coding for applications like audio transmission over the Internet.
- The *MainPlus Audio Profile* provides a rich superset of all Audio Profiles.

The Error Resilience tools can be used optionally for most of the Version 1 and Version 2 Audio Profiles. This might be realised as a set of “Error Resilience Profiles” derived from the different Audio Profiles.

2.8. Other tools in Version 2

Besides the audio-related tools in Version 2 described in the previous Subsections, there are two further tools in Version 2 that should also be mentioned here:

- The *MPEG-4 File Format* (MP4) is designed to contain the coded data of an MPEG-4 presentation in a flexible, extensible format that facilitates interchange, management, editing, and presentation of the media. This presentation may be “local” to the system containing the presentation, or may be via a network or other stream delivery mechanism. The file format is designed to be independent of any particular delivery mechanism. It is a streamable format, as opposed to a streaming format. That is, the file format does not define an on-the-wire protocol, and is never actually streamed over a transmission medium. Instead, metadata in the file known as "hint tracks" provide instructions, telling a server application how to deliver the coded data over a particular medium. The coded data itself can be located in the MP4 file or can be referenced via Universal Resource Locators (URL). The MP4 file format is based on the QuickTime format from Apple Computer Inc.

- A *Backchannel* specification permits adaptive or user-controlled streaming from a server.

3. FUTURE DEVELOPMENTS IN MPEG AUDIO

As the earlier standards MPEG-1 and MPEG-2, also MPEG-4 is a so-called "asymmetric" standard. This means that only the bitstream syntax and the decoding process are fixed in the normative part of the standard, while a possible encoding process is described in an informative annex. This permits to improve the quality of the coding scheme by ongoing encoder optimisation even after the standard has been finalised.

Since MPEG-4 Audio describes the coding of audio objects and their composition into an "audio scene", a high degree of flexibility is offered for encoding and authoring of an MPEG-4 presentation. The flexibility of such an "object based" coding system also permits improved overall coding efficiency. Imagining a radio program composed of a "speech" object and a "background music", two different approaches can be used to encode this "audio scene":

- The complete radio program is coded as a single audio object by a general audio coder.
- The "speech" object is coded by a dedicated speech coder and the "background music" object is for example coded by a parametric audio coder or even synthesised using the Structured Audio tools. Finally both decoded objects are mixed to compose the radio program in the decoder.

Especially for very low target bitrates, the second approach can provide a better overall coding efficiency [25]. This advantage can be exploited easily if the different objects are available in the authoring process. However, the automatic decomposition of an "audio scene" into objects is a difficult task and subject of ongoing research. This problem is closely related to the Computational Auditory Scene Analysis (CASA) [34].

4. CONCLUSIONS

In this paper the current status of the MPEG-4 Audio standardisation activities was presented. Version 1 of the MPEG-4 Standard, which already has been finalised, was briefly reviewed. The various new functionalities covered by MPEG-4 Audio Version 2 were explained and the tools that enable these functionalities were described. The tools in Version 2 will now undergo verification tests before the standard will be finalised in the end of 1999.

With the additional functionalities available in MPEG-4 Audio Version 2, the requirements of many evolving multimedia applications are addressed and an inter-operable solution is provided.

Up to now, MPEG Standards covered the efficient coding of audio-visual content. With the rapidly growing amount of audio-visual information available in digital form, however, it becomes more and more difficult to search and locate potential interesting material in a data base or on a network. To address this problem, MPEG started a new work item, the "Multimedia Content Description Interface" – also called MPEG-7.

ACKNOWLEDGEMENTS

The author gratefully acknowledges the work of the editors and authors of the "MPEG-4 Overview" [1], the MPEG-4 Standard [2, 14, 15], and the various contributions to MPEG-4 Version 2. These documents provided comprehensive sources for the preparation of this paper.

REFERENCES

[1] R. Koenen, "Overview of the MPEG-4 Standard," ISO/IEC JTC1/SC29/WG11 N2725, Mar. 1999. `http://www.cselt.it/mpeg/standards/mpeg-4/mpeg-4.htm`

[2] ISO/IEC, *Final Draft International Standard 14496-3: MPEG-4 Audio*, ISO/IEC JTC1/SC29/WG11 N2503, Oct. 1998.

[3] B. Edler, "Very Low Bit Rate Audio Coding Developement," *Proc. AES 14th International Conference*, Jun. 1997. `http://www.tnt.uni-hannover.de/project/coding/audio/asac/aes_iao.html`

[4] R. Koenen, "MPEG-4 Multimedia for our time," *IEEE Spectrum*, Vol. 36, No. 2, Feb. 1999, pp. 26–33. `http://www.cselt.it/mpeg/koenen/mp4ieee.htm`

[5] *Official MPEG Home Page*. `http://www.cselt.it/mpeg/`

[6] *MPEG Audio Web Page*. `http://www.tnt.uni-hannover.de/project/mpeg/audio/`

[7] M. Nishiguchi, "MPEG-4 speech coding," *Proc. AES 17th International Conference*, Sep. 1999.

[8] B. Edler, "Speech Coding in MPEG-4," submitted to *Int. J. Speech Technology*.

[9] K. Brandenburg and M. Bosi, "Overview of MPEG Audio: Current and Future Standards for Low Bit Rate Audio Coding," *J. Audio Eng. Soc.*, Vol. 45, No. 1/2, pp. 4–21, Jan./Feb. 1997.

[10] B. Grill, "The MPEG-4 General Audio Coder," *Proc. AES 17th International Conference*, Sep. 1999.

[11] B. Vercoe, W. Gardner, and E. Scheirer, "Structured Audio: Creation, Transmission, and Rendering of Parametric Sound Representations," *Proc. IEEE*, No. 86:5, pp. 922–940, May 1998.

[12] L. Ray, "MPEG-4 Structured Audio Authoring Considerations," *Proc. AES 17th International Conference*, Sep. 1999.

[13] *MPEG-4 Structured Audio Home Page*.
`http://sound.media.mit.edu/mpeg4/`

[14] ISO/IEC, "Committee Draft 14496-3 AMD1: MPEG-4 Audio Version 2", *ISO/IEC JTC1/SC29/WG11*, N2670, Mar. 1999.

[15] ISO/IEC, "Committee Draft 14496-1 AMD1: MPEG-4 Systems Version 2," *ISO/IEC JTC1/SC29/WG11*, N2739, Mar. 1999.

[16] ITU-T, *Recommendation H.223 - Multiplexing protocol for low bit rate multimedia communication*, International Telecommunication Union, Mar. 1996.

[17] J. Herre, E. Allamanche, R. Geiger, and T. Sporer, "Proposal For a Low Delay MPEG-4 Audio Coder Based on AAC," *ISO/IEC JTC1/SC29/WG11*, M4139, Oct. 1998.

[18] J. Herre, E. Allamanche, R. Geiger, and T. Sporer, "Information & Proposed Enhancements for MPEG-4 Low Delay Audio Coding," *ISO/IEC JTC1/SC29/WG11*, M4560, Mar. 1999.

[19] S-H. Park and Y-B. T. Kim, "Detailed description of BSAC," *ISO/IEC JTC1/SC29/WG11*, M4339, Dec. 1998.

[20] H. Purnhagen, B. Edler, and C. Ferekidis, "Object-Based Analysis/Synthesis Audio Coder for Very Low Bit Rates," *AES 104th Convention*, Preprint 4747, May 1998.
`http://www.tnt.uni-hannover.de/project/coding/audio/asac/aes_104.html`

[21] H. Purnhagen, B. Edler and C. Ferekidis, "Proposal of a Core Experiment for extended 'Harmonic and Individual Lines plus Noise' Tools for the Parametric Audio Coder Core." *ISO/IEC JTC1/SC29/WG11*, M2480, Jul. 1997.

[22] H. Purnhagen and N. Meine, "Core Experiment Proposal on Improved Parametric Audio Coding," *ISO/IEC JTC1/SC29/WG11*, M4492, Mar. 1999.

[23] H. Purnhagen and N. Meine, "Pre-Screening Results for CE on Improved Parametric Audio Coding," *ISO/IEC JTC1/SC29/WG11*, M4493, Mar. 1999.

[24] H. Purnhagen, B. Edler, Y. Maeda, K. Iijima, and M. Nishiguchi, "Proposal for the Integration of Parametric Speech and Audio Coding Tools based on an Automatic Speech/Music Classification Tool." *ISO/IEC JTC1/SC29/WG11*, M2481, Jul. 1997.

[25] B. Edler and H. Purnhagen, "Concepts for Hybrid Audio Coding Schemes Based on Parametric Techniques," *AES 105th Convention*, Preprint 4808, Sep. 1998.
`http://www.tnt.uni-hannover.de/project/coding/audio/asac/aes_105.html`

[26] B. Feiten, R. Schwalbe, and F. Feige, "Dynamically Scalable Audio Internet Transmission," *AES 104th Convention*, Preprint 4686, May 1998.

[27] S. Levine and J. O. Smith III, "A Switched Parametric & Transform Audio Coder," *Proc. ICASSP*, 1999.
`http://www-ccrma.stanford.edu/~scottl/icassp99.pdf`

[28] S. Levine, T. Verma, and H. Purnhagen, "Time-varying, Signal Adaptive Switching between Parametric and Transform Coding for MPEG-4 Audio," *ISO/IEC JTC1/SC29/WG11*, M4496, Mar. 1999.

[29] L. Savioja, J. Huopaniemi, T. Lokki, and R. Väänänen, "Virtual environment simulation - Advances in the DIVA project." *Proc. Int. Conf. Auditory Display (ICAD'97)*, Nov. 1997, pp. 43-46.
`http://www.hut.fi/TKK/Akustiikka/publications/ICAD97.ps.gz`
`http://www.tcm.hut.fi/Research/DIVA/`

[30] J-M. Jot, "Efficient Models for Reverberation and Distance Rendering in Computer Music and Virtual Audio Reality," *Proc. ICMC*, Sep. 1997.
`http://mediatheque.ircam.fr/articles/textes/Jot97b/`

`http://www.ircam.fr/produits/logiciels/log-forum/spat-e.html`

[31] J-M. Jot, J-B. Rault, "Extensions of Advanced AudioBIFS: a Perceptual Paradigm for the Environmental Spatialization of Audio," *ISO/IEC JTC1/SC29/WG11*, N2578, Dec. 1998.

[32] ISO/IEC, "MPEG-4 Systems Version 2 BIFS Verification Model 6.0," *ISO/IEC JTC1/SC29/WG11*, N2741, Mar. 1999.

[33] M. Serizawa, T. Nomura, H. Ito, M. Iwadare, and K. Ozawa, "A Proposal of a Silence Compression Tool for MPEG-4/CELP and its Subjective Test Results,", *ISO/IEC JTC1/SC29/WG11*, M4514, Mar. 1999.

[34] D. Ellis, *Prediction-driven computational auditory scene analysis*, PhD thesis, MIT Media Lab, 1996.