HILN – THE MPEG-4 PARAMETRIC AUDIO CODING TOOLS

Heiko Purnhagen, Nikolaus Meine

University of Hannover
Laboratorium für Informationstechnologie
Schneiderberg 32, 30167 Hannover, Germany
{purnhage,meine}@tnt.uni-hannover.de

ABSTRACT

The MPEG-4 Audio Standard combines tools for efficient and flexible coding of audio. For very low bitrate applications, tools based on a parametric signal representation are utilised. The parametric speech coding tools (HVXC) are already available in Version 1 of MPEG-4. The main focus of this paper is on the parametric audio coding tools "Harmonic and Individual Lines plus Noise" (HILN) which are included in Version 2 of MPEG-4. As already indicated by their name, the HILN tools are based on the decomposition of the audio signal into components which are described by appropriate source models and represented by model parameters. This paper gives an overview of the HILN tools, presents the recent advances in signal modelling and parameter coding, and concludes with an evaluation of the subjective audio quality.

1. 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 inter-operable solutions, ISO/IEC started its MPEG-4 standardisation activities "Coding of audiovisual objects."

The MPEG-4 Audio Standard provides tools for coding of natural and synthetic audio objects and composition of such objects into an "audio scene" [1, 2, 3, 4]. 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 (HVXC), CELP-based speech coding, parametric audio coding (HILN) or transform-based general audio coding (AAC, TwinVQ). The natural audio coding tools support bitrate scalability, also known as embedded coding. In addition, the parametric coding tools also provide speed and pitch change functionality in the decoder. Synthetic audio objects can be represented using a Text-To-Speech Interface (TTSI) or the Structured Audio (SA) synthesis tools. The SA tools are also used to add effects – like reverberation – and mix different audio objects to compose the final "audio scene" that is presented to the listener.

The first version of the MPEG-4 Standard [2], finalised in October 1998, provides a first set of natural audio coding tools, namely HVXC and CELP speech coding [5] and transform-based general audio coding [6]. With Version 2 [7], finalised in December 1999, additional tools are added to MPEG-4 to provide new functionalities not available in MPEG-4 Version 1. These are tools for fine-step scalability for the general audio coding tools (BSAC), tools for low-delay general audio coding, tools for very low bitrate parametric audio coding (HILN – "Harmonic and Individual Lines plus Noise"), and techniques to improve the error-robustness of all MPEG-4 audio and speech coding tools [8].

The main focus of this paper is on the HILN tools. While a first version of these tools [9] was scheduled for Version 1 of MPEG-4, it was decided in autumn 1998 to move the HILN tools to Version 2 of MPEG-4 to permit further advances in coding efficiency and flexibility [10]. In this process, various aspects of the HILN tools were significantly improved [11, 12, 13]. Thus better subjective audio quality and bitrate scalability is provided now.

Section 2 gives an overview of the MPEG-4 parametric audio coding tools. The recent improvements of the HILN tools are described in Section 3. Results of listening tests evaluating the subjective quality of the improved HILN tools are presented in Section 4 and conclusions are drawn in Section 5.

2. MPEG-4 PARAMETRIC AUDIO CODING

The MPEG-4 parametric audio coding tools "Harmonic and Individual Lines plus Noise" (HILN) permit coding of general audio signals at bitrates of 4 kbit/s and above using a parametric representation of the audio signal [9, 14]. The basic idea of this technique is to decompose the input signal into components which are described by appropriate source models and represented by model parameters. This approach allows to utilise more advanced source modelling 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 1 the block diagram of the HILN parametric audio encoder is depicted. First the input signal is decomposed into different components and then the model parameters for the components' 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.

The modelling of transient components is improved by optional parameters describing their amplitude envelope.

Due to the very low target bitrates of typically 6 to 16 kbit/s, only the parameters for a small number of components can be transmitted. Therefore a perception model is employed to select those components that are most important for the perceptual quality of the signal. The components' parameters are finally quantised, coded, and multiplexed to form a bitstream.

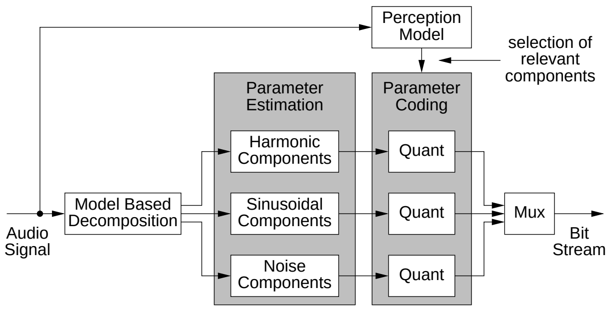

Figure 1: Block diagram of the HILN parametric audio encoder.

The block diagram of the HILN parametric audio decoder is shown in Figure 2. First the parameters of the components are decoded and then the component signals are re-synthesised according to the transmitted parameters. By combining these signals, the output signal of the HILN decoder is obtained. Because of the low phase sensitivity of the human ear, phase information for sinusoids is usually not transmitted, while phase continuity of sinusoidal tracks is provided by the synthesis tool.

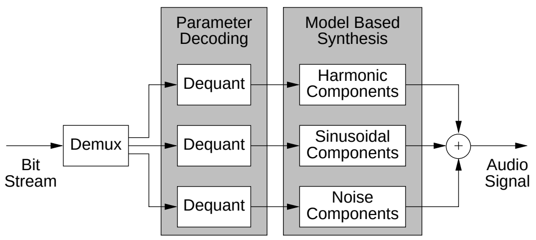

Figure 2: Block diagram of the HILN parametric audio decoder.

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.

3. IMPROVEMENTS OF HILN TOOLS

In the period since autumn 1998, coding efficiency and flexibility of the MPEG-4 HILN tools were significantly advanced. To achieve better coding efficiency, the following aspects of the HILN tools were improved [11, 12]:

- Better spectral modelling of harmonic and noise components.
- Perception-based quantisation of component parameters.
- Advanced entropy coding of quantised parameters.

To increase flexibility, also bitrate scalability for the HILN tools was introduced. These various improvements are described in detail in the following Subsections.

3.1. Spectral Modelling for Noise and Harmonic Components

The first approach for noise modelling in the HILN tools was an approximation of the noise magnitude spectrum by transmitting the first few coefficients of its Discrete Cosine Transform (DCT). The resulting synthesis algorithm is equivalent to filtering a white noise signal by an FIR filter whose order corresponds to the number of transmitted DCT coefficients. The resulting synthesised noise did not sound very natural. Especially for speech signals a lack of perceptual fusion occurred.

Motivated by the good results of Linear Predictive Coding (LPC) for speech, the DCT coding of the noise spectrum has been substituted by an LPC approach. When using LPC parameters to describe the noise spectrum, the noise synthesis can be implemented by applying an all-pole filter (IIR filter) to a white noise signal. The filter poles can form sharp resonances and therefore model narrow-band noise signals much better than an FIR filter at a given order.

Also the parametric representation of a harmonic tone components was improved. Instead of transmitting the amplitudes of each partial in a harmonic tone, now the spectral envelope of the partials is coded using LPC parameters in the same way as for the noise spectrum.

3.2. Spectral Model Parameter Coding

There are different possibilities to quantise and code the LPC parameters. The most popular representations of LPC parameters are Logarithmic Area Ratios (LARs) and Line Spectrum Frequencies (LSFs, sometimes called Line Spectrum Pairs, LSPs). The latter are used in several speech codecs and deliver a good coding efficiency. The LARs have a slightly lower coding efficiency but have advantages over LSPs when a variable LPC order is desired. Since LARs as well as LSFs are correlated between successive frames an inter-frame prediction is used to reduce the bitrate. However, if the number of LPC parameters varies from frame to frame, inter-frame prediction is much easier for LARs than for LSFs. A parameter set of N LSFs and one of $N+1$ LSFs of the same signal are completely different and their statistical dependencies are hard to exploit. When using LARs instead of LSFs, the first N LARs are identical in both sets. If the number of parameters increases from one frame to the next all LARs of the old frame can be used for prediction. Only the additional LARs in the new frame have to be coded without prediction. Since the higher order LARs have only little inter-frame correlation, a variable LPC order has only a minor negative effect on coding efficiency.

The LARs for the noise and harmonic spectrum are quantised uniformly with empirical determined quantiser step sizes. The inter-frame prediction error of the LARs has a probability density function (PDF) that is well approximated by the Laplace distribution. After quantisation with the desired step size the probability of the quantised values decreases approximately by a factor of $\frac{1}{2}$ when the absolute value is increased by one. This property permits to use an algorithmic code instead of a Huffman code without losing efficiency. The code employed here consists of a sign bit, a run of zero bits and a one bit denoting the end of the run. The number of zero bits represents the value of the coded number. If the probabilities are $\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \ldots$ for the negative as well as for the positive quantised numbers, the code is optimal.

3.3. Line Parameter Quantisation and Sub-Division Coding

The frequency parameter quantisation has been improved by using a simplified bark scale to quantise at the limit of just noticeable differences over the whole spectrum. In a corresponding way, the amplitude parameters are quantised uniformly on a logarithmic scale. A step size of $\frac{1}{32}$ Bark and 1.5 dB respectively is used in HILN now.

The first version of the HILN tools used less advanced frequency quantisation and fixed length codes for the frequency and amplitude parameters of new individual lines and for the frequency and amplitude changes of lines continued from the previous frame.

Since the changes in frequency and amplitude of a continued line are not uniformly distributed, a variable length code is now used to provide entropy coding, thus requiring less bits.

A more sophisticated approach is employed to code the quantised parameters of the new individual lines in a frame, exploiting the statistical dependencies of the frequency-ordered line parameters. In this scheme – referred to as Sub-Division Coding – all parameter pairs (frequency and amplitude) are sorted by their frequency. After that the frequencies f_n are coded in increasing order $0 <= f_1 <= f_2 <= \ldots <= f_N < F$. For each frequency f_n the range of possible values is limited by the previously coded frequency f_{n-1} (0 for the first one) and the highest possible frequency F which is half the sample rate of the analysed signal. The range is divided into two parts of equal probability (i.e. $\frac{1}{2}$). One bit is sent to indicate in which part the actual frequency is located. This part becomes the new range for the frequency which is then divided again. This step is repeated recursively until the remaining frequency range contains only one possible value.

Figure 3 shows an example of coding $N = 3$ frequency parameters f_n. In this example each of the codewords is complete after two sub-division steps. It is also assumed that the N quantised frequencies are uniformly distributed in the range $[0 \ldots F-1]$, so that the PDF of the first (i.e. lowest) frequency is approximated by Equation 1. Since the real PDFs differ slightly from this assumption, empirically collected data is used in the boundary tables for the sub-division process in the actual implementation.

$$p_1(f_1) = \frac{N}{F}\left(1 - \frac{f_1}{F}\right)^{N-1} \tag{1}$$

The same sub-division algorithm operates in the decoder using the transmitted bits to select the appropriate parts. The exact positions of the boundaries between parts are stored in boundary tables. Since these positions differ depending on the number $N-n$ of frequency parameters that follow, the boundaries are stored only for zero up to seven following parameters. If more than seven frequencies follow, the boundary table for seven is used. The boundaries are stored in a normalised form and have to be scaled to match the actual width $F - f'_{n-1}$ of the initial range when used. Only the boundaries for the first five sub-division steps are stored. If further sub-division is necessary, the boundary is located in the middle of the remaining frequency range, assuming a locally uniform distribution.

The quantised line amplitudes are coded relative to a maximum amplitude which represents the loudest component in the frame, including noise and harmonics. The difference to this maximum level is coded using the same sub-division algorithm as for the frequencies with a specific boundary table.

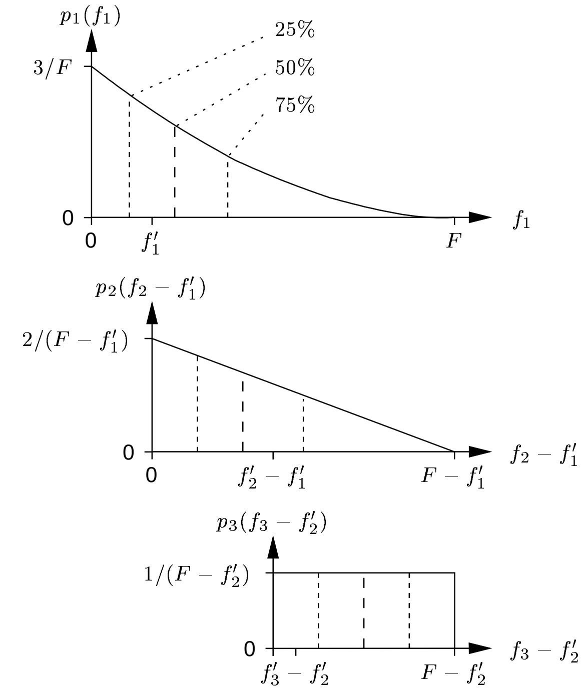

Figure 3: Example of Sub-Division Coding of $N = 3$ frequency parameters $0 <= f'_1 <= f'_2 <= f'_3 < F$ as bit sequence 011000.

The presented algorithm exploits a non-uniform PDF nearly as well as a special Huffman code but requires only a table of only 32 entries of 10 bit fixed point values which are calculated from the PDF.

Using the improved quantisation and coding for the individual line parameters, the coding efficiency for these parameters was improved by 20 to 30 % [11].

3.4. Bitrate Scalability

Since the bitrate as well as the audio quality depends on the number of components and their parameters that can be transmitted in the bitstream, bitrate scalability can be implemented by transmitting the most important components in a base layer bitstream and further, less important components in one or more enhancement bitstreams [13]. In case of limited transmission bandwidth, only the base layer is received and decoded. If a higher bandwidth is available, also enhancement layers are received. Together with the base layer information the audio signal can then be reconstructed with higher quality.

4. SUBJECTIVE QUALITY EVALUATION

The MPEG-4 Audio Version 2 coding tools have recently undergone a verification test [15]. As part of this listening test, the subjective audio quality of the HILN coding tools was compared to the transform-based coding tools TwinVQ and AAC available in MPEG-4 Version 1 at bitrates of 6 kbit/s and 16 kbit/s using critical audio signals. The HILN tools were operated at 6 and 16 kbit/s (non-scalable) and in a scalable configuration using a 6 kbit/s base and a 10 kbit/s enhancement layer. The BS.1284 quality scale with the 8 kHz bandwidth original signal as reference was used for grading. The overall test results for all 7 audio items and 16 listeners are shown in Figure 4.

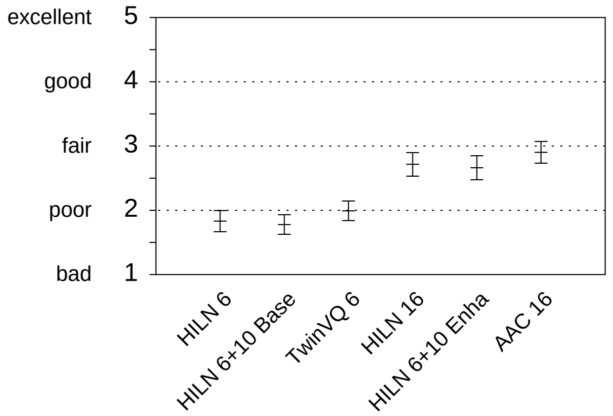

Figure 4: MPEG-4 Version 2 verification test results for HILN at 6 and 16 kbit/s showing overall mean grading and 95% confidence interval for 7 items and 16 listeners (from [15]).

From the test results, it can be concluded that HILN at both 6 and 16 kbit/s shows performance comparable to other MPEG-4 coding technology operating at similar bitrates while providing the additional capability of independent speed and pitch change in the decoder. The bitrate scalability feature of HILN did not show a penalty in quality compared to non-scalable HILN. When comparing the current test results for HILN to those of the Version 1 verification test [10], it becomes obvious that the subjective quality of HILN was significantly improved by the modifications described in Section 3.

Since the MPEG-4 Audio standard combines state-of-the-art schemes for very low bitrate audio and speech coding, it permits direct comparison of the different schemes. Such comparisons have shown that performance of speech, transform-based, and parametric coding tools at very low bitrates (like 6 kbit/s) highly depends on the type of the audio material being coded and that each of these three schemes performs best for a certain class of audio material. This fact is reflected in the flexible structure of MPEG-4, which permits to select the coding scheme optimal for the actual content of an audio object and allows for combination of different audio objects in an "audio scene."

5. CONCLUSIONS

In this paper, the HILN parametric audio coding tools in the MPEG-4 Audio Version 2 standard were presented and the recent improvements in coding efficiency and flexibility were described. Listening tests evaluating the subjective audio quality of the HILN tools show that the HILN tools can complement the tools available in Version 1 of MPEG-4. In addition, speed and pitch change functionality is provided. Because only the bitstream syntax and the decoding process are standardised by MPEG, future encoders can utilise advanced decomposition and parameter estimation techniques currently under development [16, 17, 18, 19]. Another attractive field for future research is the development of encoding schemes that automatically select the optimal MPEG-4 coding tools or even a combination thereof.

6. 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, *14496-3 FDIS: MPEG-4 Audio*, ISO/IEC JTC1/SC29/WG11 N2503, Oct. 1998.

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

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

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

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

[7] ISO/IEC, *14496-3 FDAM1: MPEG-4 Audio Version 2*, ISO/IEC JTC1/SC29/WG11 N3058, Dec. 1999.

[8] H. Purnhagen, "An Overview of MPEG-4 Audio Version 2," *Proc. AES 17th International Conference*, Sep. 1999.

[9] 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.

[10] ISO/IEC, *MPEG-4 Audio verification test results: Audio on Internet*, ISO/IEC JTC1/SC29/WG11 N2425, Oct. 1998.

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

[12] N. Meine, *LPC-Spektralmodelle für einen Analyse/Synthese Audio Coder*, Diplomarbeit, Universität Hannover, Jan. 1999.

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

[14] B. Edler, H. Purnhagen, and C. Ferekidis, "ASAC - Analysis/Synthesis Audio Codec for Very Low Bit Rates," *AES 100th Convention*, Preprint 4179, May 1996.

[15] ISO/IEC, *Report on the MPEG-4 Audio Version 2 Verification Test*, ISO/IEC JTC1/SC29/WG11 N3075, Dec. 1999.

[16] H. Purnhagen, "Advances in Parametric Audio Coding," *Proc. IEEE WASPAA*, Sep. 1999.

[17] S. Levine, *Audio Representations for Data Compression and Compressed Domain Processing*, PhD thesis, CCRMA, Stanford University, Dec. 1998.

[18] K. Vos, R. Vafin, R. Heusdens, and W.B. Kleijn, "High-Quality Consistent Analysis-Synthesis In Sinusoidal Coding," *Proc. AES 17th International Conference*, Sep. 1999.

[19] T. Verma, *A Perceptually Based Audio Signal Model with Application to Scalable Audio Compression*, PhD thesis, Stanford University, 2000.

Further references and related links can be found at:
`http://www.tnt.uni-hannover.de/~purnhage/`