Audio Watermarking of MPEG-2 AAC Bit Streams

Christian Neubauer, Jürgen Herre
Fraunhofer Institute for Integrated Circuits IIS
D-91058 Erlangen, Germany

Abstract

Today music distribution of compressed audio material, e.g. over the Internet, is an increasingly important technology. However, controlling use and further distribution of the delivered works is usually not possible. Within this context watermarking can provide a useful mechanism to track illicit copies or to attach property rights information to the multimedia content. This paper discusses the embedding of robust watermarks directly into MPEG-2 AAC bit streams such that the watermark can be detected in the decompressed audio data.

1 Introduction

Today music distribution over the Internet is an increasingly important technology. Most of the music content is compressed [1, 2, 3] in order to save disk space and speed up transmission over band limited channels. However, controlling the use of the distributed works or tracking illicit copies thereof still presents major problems. Within this context watermarking can provide a useful mechanism to track such illicit copies or to attach property rights information to the material.

Watermarking of uncompressed multimedia data e.g. images, video, audio is a well known technique [4, 5, 6, 7, 8, 9]. To embed watermarks into compressed material, however, a fast, quality preserving watermarking scheme is needed. While this has been accomplished for video signals [10], for audio signals such a scheme is still missing. Decoding the (audio) bit stream, embedding the watermark in the uncompressed domain and re-encoding it again is not favorable for both reasons of computational complexity and signal quality. Instead, a system capable of watermarking compressed bit streams presents a much more attractive solution.

This paper presents a method for embedding watermarks directly into compressed MPEG-2 AAC bit streams. First an overview over the state of the art in audio coding and spread spectrum watermarking is given. Next some requirements for bit stream watermarking are discussed and the basic idea of the scheme is presented. It is based on a convenient usage of relevant parts of a MPEG decoder, PCM watermarking and MPEG encoder. Finally, results gathered from a sample implementation are presented including achieved audio quality, watermark detection performance and computational complexity.

2 Motivation and Requirements

The basic idea of watermarking bit streams is to partly decode the input bit stream, add a perceptually hidden watermark in the frequency domain and finally quantize and code the signal again. There are some basic considerations that apply to such a system in general as described in [10]. Nevertheless, the most significant aspects will be revisited here briefly.

Let us consider a "pay audio" scenario. Typically, the provider stores the audio content in compressed format. During download of music, the customer identifies himself with his unique customer ID, which therefore is known to the provider during delivery. In order to embed the customer ID into the audio data using a watermark technique, a scheme is needed that is capable of watermarking compressed audio on the fly during the download. It can be seen from the above that there are some basic requirements to such a scheme which will be outlined in following paragraphs.

Inaudibility of the Watermark: The watermark is an additional noise-like signal added to the original audio data. Therefore special attention must be paid in order to hide the watermark perfectly from a human listener. This is achieved by applying a perceptual model in order to guarantee that the watermark signal does not exceed the masking threshold.

Handling of Bit Streams In the pay audio scenario described above the provider will typically store the compressed content. Therefore it is necessary that the watermarking system is capable of of handling compressed domain signals. The watermarking system shall produce a compressed domain signal as output. This requirement is the essential feature of the system.

Fast Operation Consider delivery of compressed content in a streaming scenario. In order to serve many parallel requests for content the delivery server must be able to watermark bit streams much faster that realtime.

Robustness Robustness of watermarking systems is often considered to be the most important issue. It refers to the idea that intentional or unintentional removal of the watermark should only be possible by accepting a degradation of the audio signal quality. In this paper, the robustness of the system is not assessed. Nevertheless, correct use of spread spectrum techniques provides a certain amount of robustness. This originates from the facts that the spreading sequence used for transmission can be considered as a key and secondly because the watermark signal is completely spread over frequency and time. In this way does not depend on an undisturbed transmission in a particular frequency region or point in time.

Interoperability with PCM Watermarking There are two different aspects of interoperability, as shown in Fig. 1. The first one is to ensure that the same watermark extractor can be used for both the bit stream and the PCM watermark embedding scheme. In other words the same decoder should be used for PCM watermarking as well as for bit stream watermarking. In order to achieve this, the same watermark signal representation and modulation must be used in both systems, PCM and bit stream watermarking.

The second issue is conformance of input bit streams that can be worked up by the bit stream watermarking system. There are two choices, an incompatible one, that relies on the fact that the preceding perceptual coder left some "space" in order to be filled up with a watermark and a compatible one that can watermark standard bit streams. Currently the incompatible choice is taken by our sample implementation with the constraint of slightly increased bit rate of the output bit stream.

Bit Rate Concerning the bit rate two choices are possible. The first maintains a fixed bit rate. This can be achieved by watermarking only those coefficients that do not increase the bit rate when coded by the noiseless coder. We choose the second approach which leads to a slight increase in bit rate. However it ensures that the watermark signal is present at all times. This enhances the detection performance.

3 State of the Art

3.1 Perceptual Coding

A block diagram of the perceptual MPEG-2 AAC audio coder is shown in Fig. 2. It is based on the idea to hide quantization noise below the so called masking threshold. It comprises of the following blocks.

- Analysis Filterbank
  The filterbank in AAC is realized as a critically subsampled Modified Discrete Cosine Transform (MDCT) with an overlap of 50% between subsequent analysis windows. Its purpose is to provide a spectral representation of the input signal, which finally is quantized and coded. Together with the filterbank in the decoder the analysis filterbank forms an analysis/synthesis system.

- Perceptual Model
  The perceptual model is based on the psychoacoustic phenomena of masking. Frequency domain masking is modeled as well as time domain masking. Examples of both masking phenomena are shown in Fig. 3. The masking model estimates the amount of (noise-like) energy that can be added to the original audio signal without becoming perceptible. This energy varies over frequency. Therefore the implementation of the masking model introduces frequency band partitions. The maximum energy allowed in a particular band is referred to as the masking threshold.

- Quantize & Code
  Typically more than one spectral line is quantized with the same quantization step

size. Therefore spectral lines are grouped into so-called scalefactor bands. The quantizer optimizes the quantization step size for each scalefactor band. The quantizer step size is determined so that the quantization error is below or equal to the estimated masking threshold to ensure the quantization noise is inaudible. It is obvious that two constraints which form a trade-off must be considered. On the one hand the bit consumption should be kept as low as possible in order to get high compression ratios, but at the same time the demands of masking must be fulfilled. Typically this optimization process is computed in an iterative loop. The result of this loop is a quantization step size (scalefactor) for each scalefactor band. Note that the actual embedded quantization noise may exceed or fall short of the computed masking threshold. This is an important issue for the bit stream watermark embedder concerning the estimation of the correct masking threshold.

- Bit Stream Multiplexer
  The bit stream multiplexer assembles the output bit stream. It contains the quantized and coded spectral coefficients as well as the scalefactors and some side information.

The decoder is shown in Fig. 4. It parses the input bit stream within the bit stream demultiplexer and outputs scalefactors and a spectral representation. Both are still Huffman coded and quantized. Huffman coding and quantization are countermand within the "decode & quantize" block. Finally the synthesis filter bank is applied which reconstructs the desired time audio signal.

3.2 PCM Watermarking

3.2.1 Watermark Embedder

In a PCM watermarking system [11] the masking thresholds are used in order to hide a watermark signal in very much the same way quantization noise is hidden in a perceptual coder. A block diagram of such a scheme is shown in Fig. 5. While the rate of the data signal is small (typically in the range of a few bits per second) the data signal is expanded in bandwidth by the operation of spreading [12, 13, 14]. Typical bandwidths after spreading are 12 kHz to 16 kHz. This signal is filtered with a time variant filter. The masking threshold provided by the perceptual model is used by this filter in order to shape the energy distribution of the spread spectrum data signal over frequency. This is accomplished by applying a specific attenuation to each frequency partition of the spread spectrum data signal. After performing this operation the spectral energy distribution of the data signal over frequency follows the masking threshold of the audio signal and in this way ensures inaudibility of the watermarking signal after embedding.

3.2.2 Watermark Extractor

The watermark extractor operates on PCM audio signals and consists mainly of a standard direct sequence spread spectrum receiver. The block diagram is shown in Fig. 6.

Note that the extractor only recovers the hidden information (watermark) using the watermarked audio signal as a carrier signal. The original audio signal itself can be considered as a jammer interfering with the transmitted information bearing signal. From this point of view the watermark transmission is a communications problem of detecting a low energy data signal in a channel with a timevariant impulse response that is distorted by a non-stationary non-white interferer (i.e. the audio signal).

The most important part of the extractor is the matched filter. The filter coefficients match the spreading sequence used in the encoder. Behind the matched filter, peaks representing the sent information are observed. They occur approximately in symbol timing. This timing information is used by the synchronizer in order to estimate the exact symbol timing and sampling time.

Knowing the sampling time, the matched filter output signal can be sampled and a threshold decision recovers the sent information. In the case of BPSK the threshold is zero. Thus a positive sample indicates a binary "0", a negative sample indicates a "1", or vice versa.

4 Bit Stream Watermarking Scheme

4.1 Basic Function

The basic idea of the bit stream watermarking system is shown in Fig. 7. It can be seen that the goal of bit stream watermarking can be achieved easiest by only invoking the relevant parts (highlighted in gray in Figs. 2, 4, 5) of perceptual decoder, watermarking system and perceptual encoder. The benefits of this strategy are

- fast operation by avoiding full decoding of the input bit stream
- control over window alignment, better control over leakage of distortion
- control over ratio between coder distortion and watermark distortion.

The block diagram of the bit stream watermarking scheme is shown in Fig. 8. It comprises three blocks.

Parts of Decoder The input bit stream is parsed in the bit stream demultiplexer, followed by Huffman decoding. In order to retrieve the spectral representation of the signal, an inverse quantizer is applied. Furthermore, the scalefactors are provided by the bit stream parser. The information used in the bit stream watermarking scheme consists of scalefactors and an spectral signal representation.

Watermark Generator The watermark generator receives an arbitrary bit stream of "0"s and "1"s which forms the watermark data. This bit stream typically is cyclically repeated in order to retrieve the watermark even in the case of cropping of the audio signal. It is up to the user what kind of information is transmitted. Our demonstration implementation, for example, is capable of transmitting frames according to the specification of the SDMI[1] consortium as well as random bit sequences for bit error measurements. The watermark data, a low bandwidth data signal, usually in the range of a few bit/s, is spread in bandwidth to match the channel bandwidth (e.g. 16 kHz) of the carrier signal (i.e. the audio signal).

Weighting & Adding After spreading, the watermark signal is shaped in the frequency domain by applying a time variant filter. This is done in order to hide the watermark below the masking threshold, thus making it imperceptible to a human listener. The shaped watermark signal is finally added to the audio signal. The estimation of the masking thresholds will be further discussed in Sect. 4.2.

Parts of Encoder Finally the resulting spectrum is quantized again. At this point the question arises how to determine the quantizer interval. In order to avoid tandem coding distortions (i.e. distortion accumulation due to repeated quantization) the same quantization step sizes and scalefactors are used as in the original bit stream. After Huffman coding the output bit stream is generated by the bit stream multiplexer.

4.2 Specifics

4.2.1 Need for Psychoacoustic Weighting

To ensure inaudibility of the watermarking signal the bit stream watermarking system needs to do some weighting of the spread spectrum data signal. For example, within the PCM watermarking system the masking model provides thresholds that are used by a time variant filter in order to shape the data signal in the spectral domain. Within the context of bit stream watermarking, however, threshold information is not available. Therefore, a different approach is needed to solve this problem.

For the following discussion, it is necessary to clearly distinguish between two quantities:

- masking threshold: maximum energy level that is masked by the signal
- quantization distortion: distortion energy introduced into the signal due to the quantization operation carried out in the preceding perceptual encoding stage

Depending on the mode of operation of the perceptual encoder used, different relations may hold between those quantities [15].

[1]SDMI: Secure Digital Music Initiative

4.2.2 Modes of Encoder Operation

Constant Quality - Variable Bit Rate Coding An encoder which tries to introduce exactly the amount of distortion that is given by the masking threshold will produce a constant quality audio signal. Due to the possibly time-varying nature of the input signal, this generally leads to a variable bit rate.

Using well-known principles from quantization theory, it is possible to determine the expected value of quantization distortion based on the quantizer step size (scalefactor) information available in the watermarked bit stream. Taking furthermore into account the close relation between quantization distortion and masking threshold, it is possible to derive an estimate of the masking threshold as required for the spectral weighting of the watermarking signal.

While constant quality coding is an attractive concept in a perceptual sense, the associated variable data rate restricts the use of this approach to application scenarios which support a variable transmission rate, e.g. digital storage applications or transmission over packet switched networks (Internet).

Constant Rate - Variable Quality Coding In most application scenarios of audio coding (e.g. applications using constant rate transmission channels), however, a constant bit rate is required. Due to the time-varying spectral and temporal properties of audio signals, this inevitably leads to a variable quality. In particular, demanding portions of the input signal may end up being coded badly (under-coded) while other easy-to-code segments may be coded with more precision than is required for perceptual transparency (over-coding). Consequently, an estimate of the encoding quantization distortion (which may be based on the quantizer step sizes) will not deliver an adequate indication of the masking thresholds of the audio signal which was encoded. Thus, different approaches for masking threshold estimation are necessary in the context of constant rate / variable quality coding.

Analysis by Synthesis Many contemporary high-performance audio encoder implementations (including MPEG-2 AAC) are based on so-called *analysis by synthesis* (AbS) techniques in the sense that the encoding quantization distortion is measured by comparing quantized and unquantized (original) spectral data. If the measured distortion does not exceed the masking threshold (e.g. because many spectral coefficients fall close to the levels defined by the quantizer grid), no further refinement of the quantizer step size will be made. This is true even if the expected value of the quantization distortion (based on the step size) would be much higher than the masking threshold. While this strategy is efficient in exploiting local rate minima, the quantizer step sizes used will not necessarily reflect the level of the masking threshold as needed for the weighting process of the watermark generation. Consequently, use of the AbS strategy presents a further complication to the bit stream watermarking process, both in the constant and the variable bit rate case.

Based on the preceding observations, a number of options for determining the masking threshold for watermark weighting can be conceived:

- Explicit computation of the masking model
  The solution of computing a complete masking model within the watermark embedder implies full decoding of the bit stream down to the time domain, including the synthesis filterbank. Accordingly, this approach results in a significant increase in both computational and structural complexity.

- Estimation from quantizer data
  As indicated previously, another option is to estimate the masking thresholds from quantizer step size information (scalefactors). However, with reference to the above discussion, such an estimation will only produce meaningful results if the preceding perceptual coder uses the "coding at masking threshold" strategy and does not employ analysis by synthesis. This represents a critical restriction of the estimation approach.

- Transmission from encoder
  A third option is to transmit the masking threshold information which was calculated in the encoder to the bit stream watermarking system, either within or outside of the bit stream. This solution provides both accurate threshold values and low complexity of the bit stream watermarking scheme. While transmission of masking thresholds increases the amount of information to be stored before the watermarking process, it is clear that this information will not be passed on into the final watermarked bit stream and thus does not increase the eventual bit rate.

Because of the advantages of the third option (i.e. transmission of masking thresholds) this alternative was chosen for the implementation of the investigated bit stream watermarking system. The sharing of the threshold information between encoder and watermarking system was implemented via an external data file.

5 Results

In this section, results gathered by a practical implementation of the system are presented. This includes decoding results, measurements of the obtained audio quality as well as an estimate of the computational burden compared to a watermarking system which is not based on bit stream watermarking. It is important to note that, due to the fundamental trade-off between watermark energy and subjective audio quality, the results of listening tests and bit error measurements should always be viewed in the context of each other. Increasing watermark energy increases detection performance (and thus decreases bit error rate in the decoded watermark data) while at the same time decreasing audio quality.

5.1 System under Test

All test signals were processed by the setup shown in Fig. 8. For encoding, an extended AAC encoder was used which additionally stored the computed masking thresholds into a separate threshold file that could be accessed by the bit stream watermark embedder. Furthermore, to simplify experimentation, the encoder operated at a variable bit rate and did not make use of any additional coding tools, such as intra-channel prediction or Temporal Noise Shaping (TNS) [16].

The actual system under test, wherefrom the following results are derived, is an implementation based on parts of a standard MPEG-2 AAC audio encoder, parts of our PCM watermarking system and a standard AAC decoder as shown in Fig. 9 . The listening test was carried out with test items described in Tab. 3, whereas the WBER[2] was determined with items described in Fig. 4. This was done because items used for a listening test should not exceed 20s in length to allow for a reliable assessment of subjective audio quality. On the other hand for watermark bit error testing, the items should be as long as possible to enhance the statistics of the measurement. Therefore we decided to used two distinct sets of items.

5.2 Bit Rate Increase Due to Watermark Embedding

As previously discussed in Section 4.2.1, the watermark embedder is expected to increase the bit rate of the MPEG-2 AAC bit stream by some amount. This increase in bit rate is quantified by experimental results and is listed in Tab. 1. As can be seen from these results, the increase is well below 3% of the original bit rate. Consequently, for applications with a fixed maximum bit rate, this effect can be taken into account by carrying out the encoding process with a slightly lower bit rate and leaving some safety margin to be used by the bit stream watermarking scheme.

Item	Bitr. incr. [%]	Item	Bitr. incr. [%]	Item	Bitr. incr. [%]
Astor11	2.74	Jazz06	1.89	Oboe05	2.70
AstorP13	2.54	Klassik01	2.21	Oboe06	2.56
ChickK05	1.85	Klassik05	2.10	Synth06	1.83
ChickK13	1.93	Klavier02	2.53	Synth07	2.05
Jazz02	1.78	Klavier03	2.05		

Table 1: Bit rate increase caused by watermarking an MPEG-2 AAC coded item

5.3 Listening Test

The evaluation of subjective audio quality of the watermarked material was done by a MUSHRA[3] listening test. The listeners were both experienced and familiar with the used set of test items. These items represent a corpus of audio material which contains excerpts from single instruments, multiple instruments, complex sound sources and speech and have

[2]WBER: Watermark Bit Error Rate, Ratio between wrong watermark bits and total transmitted watermark bits

[3]Multi Stimulus with Hidden Reference and Anchors

been used extensively for assessment of subjective sound quality in the MPEG-4 audio development process before. A short description can be found in Tab. 3.

To give an anchor for comparison, the quality of the watermarked/coded items is compared to the quality of unwatermarked/coded items. The results of the listening test are shown in Fig. 10

As can be seen from the above results, the quality degradation of the bit stream watermarking system is very small for the vast majority of the test items. For all items the confidence intervals of both signals overlap which indicates that there is no significant distortion introduced by the system.

5.4 Watermark Bit Error Rate Measurement

This section presents the results of the watermark bit error rate measurements. For this purpose, a known sequence is transmitted over the watermark channel. Finally, after watermark decoding, the retrieved data sequence is compared to the transmitted (known) sequence on a bit by bit basis and the number of wrong bits is measured. The numbers given in Tab. 2 are the watermark bit error rates (WBER) which are defined as the ratio between the number of erroneous decoded bits and the number of total transmitted bits of the watermarking data. A short description of the items used for this test can be found in Tab. 4.

Item	WBER [%]
Astor11	$1.69 \cdot 10^{-2}$
AstorP13	$4.9 \cdot 10^{-3}$
ChickK05	$1.18 \cdot 10^{-3}$
ChickK13	$4.72 \cdot 10^{-4}$
Jazz02	$5.52 \cdot 10^{-4}$
Jazz06	$< 2.7 \cdot 10^{-4}$
Klassik01	$< 2.0 \cdot 10^{-4}$
Klassik05	$< 1.6 \cdot 10^{-4}$
Klavier02	$4.37 \cdot 10^{-4}$
Klavier03	$< 2.68 \cdot 10^{-4}$
Oboe05	$7.56 \cdot 10^{-2}$
Oboe06	$< 5.5 \cdot 10^{-4}$
Synth06	$< 3.4 \cdot 10^{-4}$
Synth07	$2.24 \cdot 10^{-3}$

Table 2: Watermark bit error rate (WBER) for watermark decoding

5.5 Computational Complexity

Fig. 11 illustrates and compares the implementation complexity of different watermarking options. The y-axis is normalized to the playing time (we refer to as "real-time") of the watermarked signals. These results were obtained on a Pentium II@400 Mhz with file I/O on its local hard disk drive and represent mean execution times, averaged over all test items. The implementation is based on source code containing both C and C++ modules.

Firstly, and for comparison, the execution time is given for processing by a chain of a separate MPEG-2 AAC encoder, a PCM watermarking system and an audio decoder. In contrast, the second entry in the figure gives the equivalent execution speed of the proposed bit stream watermarking system. From these results it can be seen that the bit stream watermarking system performs approximately four times faster than the comparison system which represents a significant increase in efficiency. In absolute terms, the watermarking of bit streams is carried out almost five times faster than the actual playing time of the items. This should be a good basis for using bit stream watermarking techniques for on-the-fly watermarking in "pay audio" and other server-based applications.

6 Conclusions

In this paper, we briefly reviewed the current state-of-the-art in watermarking and discussed requirements for a useful watermarking system. Based on these considerations, the concept of bit stream watermarking was presented as an evolution of the standard PCM watermarking / compression scenario. As one significant advantage, this approach enables efficient on-the-fly watermarking and personalization of compressed audio content stored on a server. A further benefit of such a system is its awareness of the consequences of the preceding encoding step.

Next, a prototype system was presented which allows embedding of additional watermark data into MPEG-2 AAC coded audio signals. It was shown that the embedding process can be performed much faster than real-time on today's personal computers. Furthermore a listening test was performed, confirming that the presented system does not introduce significant distortion into the vast majority of the test items. At the same time, the embedded watermark data can be detected reliably. For the majority of items the number watermark bit errors was around 1 bit out of 10^3 which is a promising result. This can be further improved be applying some kind of error correction.

Future work is to be done for various aspects of the system, including further optimization of the sound quality and support for additional coding AAC modes (such as Temporal Noise Shaping, joint stereo coding etc.). Also, a joint optimization of perceptual encoding and bit stream watermarking is needed to guarantee optimum performance of the overall system.

7 Acknowledgments

The authors would like to thank Frank Siebenhaar of the Fraunhofer Institute for Integrated Circuits for the support while writing this paper as well as all subjects who participated in the listening tests.

References

[1] ISO-IEC/JTC1 SC29 WG11 Moving Pictures Expert Group. Coding of moving pictures and associated audio for digital storage media at up to 1.5 mbit/s. International Standard 11172-3, ISO/IEC, 1993.

[2] ISO/IEC JTC1/SC29/WG11 Moving Pictures Expert Group. Generic Coding of Moving Pictures and Associated Audio: Advanced Audio C oding. International Standard 13818-7, ISO/IEC, 1997.

[3] M. Bosi, K. Brandenburg, S. Quackenbush, K. Akagiri, H. Fuchs, J. Herre, L. Fielder, M. Dietz, Y. Oikawa, and G. Davidson. ISO/IEC MPEG-2 Advanced Audio Coding. *Journal of the AES*, 45(10):789–814, October 1997.

[4] Ingemar J. Cox, Joe Killian, Tom Leighton, and Talal Shamoon. A secure, robust watermark for multimedia. In *Workshop on Information Hiding*. Newton Institute, University of Cambridge, May 1996.

[5] Laurence Boney, Ahmed H. Tewfik, and Khaled N. Hamdy. Digital watermarks for audio signals. In *1996 IEEE Int. Conf. on Multimedia Computing and Systems*, pages 473–480, Hiroshima, Japan, 1996.

[6] Daniel Gruhl, Anthony Lu, and Walter Bender. Echo hiding. In *Proceedings of the Workshop on Information Hiding*, number 1174 in Lecture Notes in Computer Science, Cambridge, England, May 1996. Springer Verlag.

[7] Jonathan K. Su, Frank Hartung, and Bernd Girod. Digital watermarking of text, image and video documents. *Computers & Graphics*, 22(6):678–695, Dec. 1998.

[8] Christian Neubauer, Jürgen Herre, and Karlheinz Brandenburg. Continuous steganographic data transmission using uncompressed audio. In *Workshop on Information Hiding*. Intel Corporation, April 1998.

[9] Ricardo A. Garcia. Digital watermarking of audio signals using a psychoacoustic auditory model and spread spectrum theory. In *107th AES Convention*, New York, Sep. 1999. preprint 5073.

[10] Frank Hartung and Bernd Girod. Watermarking of uncompressed amd compressed video. *Signal Processing*, 66(3):283–301, May 1998.

[11] Christian Neubauer and Jürgen Herre. Digital watermarking and its influence on audio quality. In *105th AES Convention*, San Francisco, Sep. 1998. preprint 4823.

[12] Heinz Schulte and Michael Tangemann. Multiplex- und mehrfachzugriffsverfahren. *Unterrichtsblätter der Deutschen Telekom*, 9:516–527, Sep. 1999.

[13] Robert Dixon. *Spread Spectrum Systems*. Wiley & Sons Inc., 3rd edition edition, 1994.

[14] John G. Proakis. *Digital Communications*. Mac Graw Hill, 3rd edition edition, 1995.

[15] Jürgen Herre. Temporal noise shaping, quantization and coding methods in perceptual audio coding: A tutorial introduction. In *17th AES Conference*, Florence, Sep. 1999.

[16] Jürgen Herre and J.D. Johnston. Enhancing the Performance of Perceptual Audio Coders by Using Temporal Noise Shaping (TNS). In *101st AES Convention*, Los Angeles, Sep. 1996. preprint 4384.

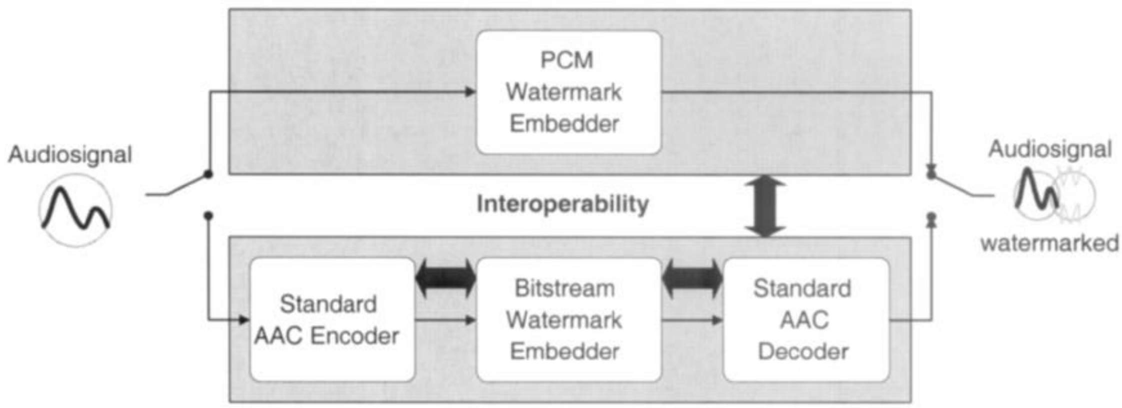

Figure 1: Interoperability between watermarking systems

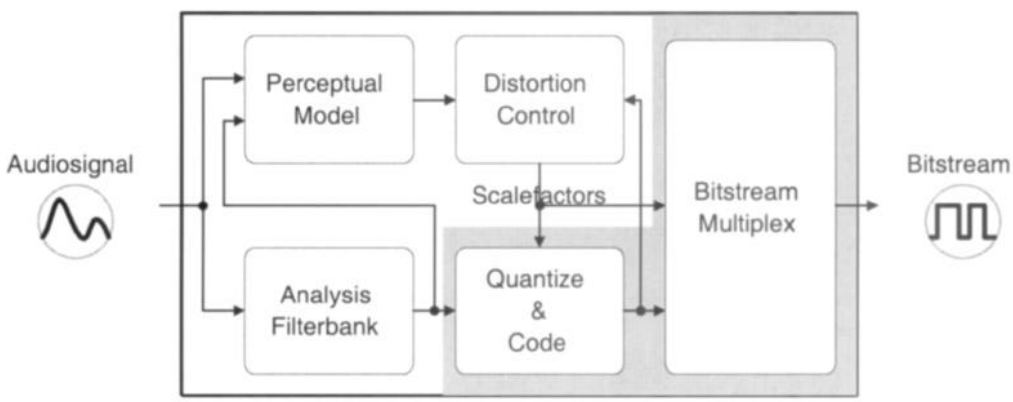

Figure 2: Block diagram of MPEG-2 AAC encoder

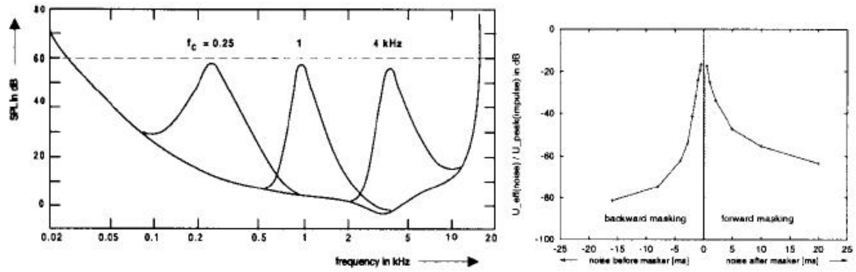

Figure 3: Examples of frequency domain masking (left) and time domain masking (right)

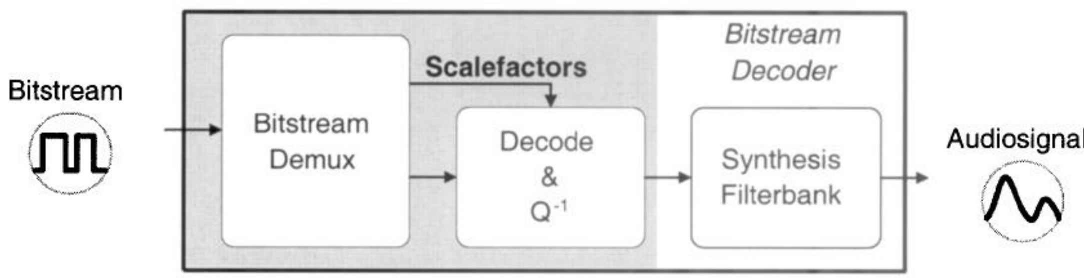

Figure 4: Block diagram of MPEG-2 AAC decoder

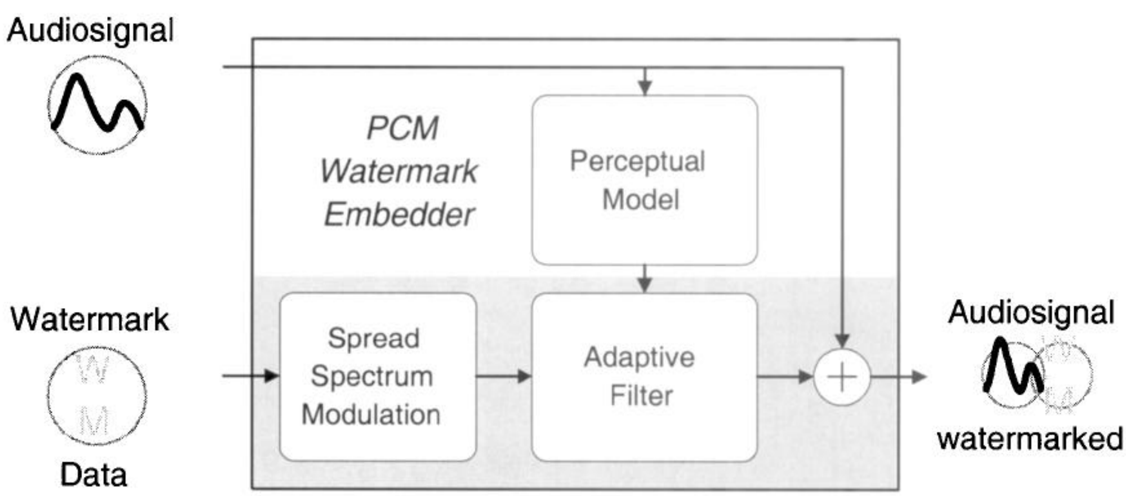

Figure 5: Block diagram of PCM watermark embedder

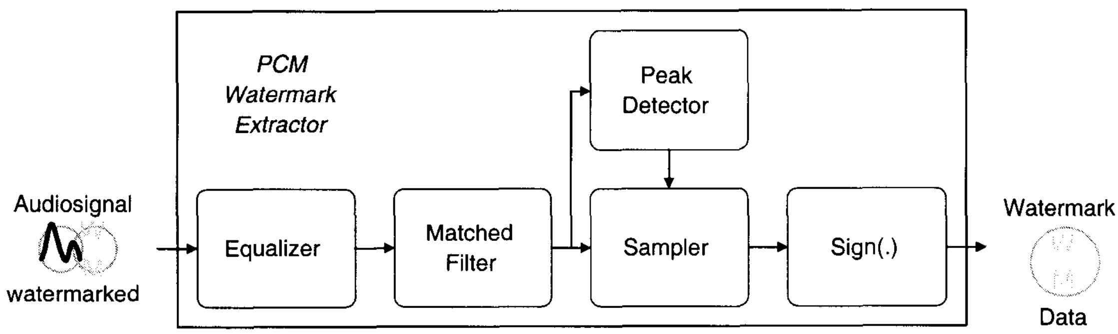

Figure 6: Block diagram of watermark extractor

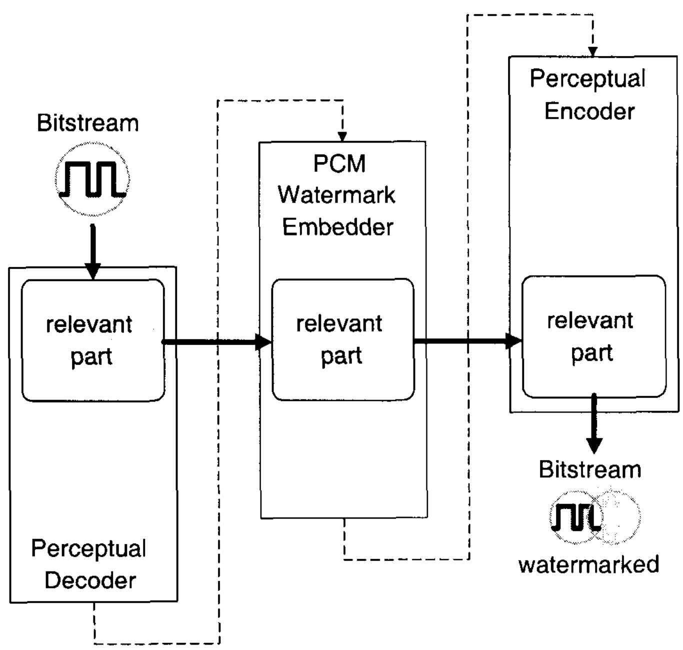

Figure 7: Basic idea behind bit stream watermarking

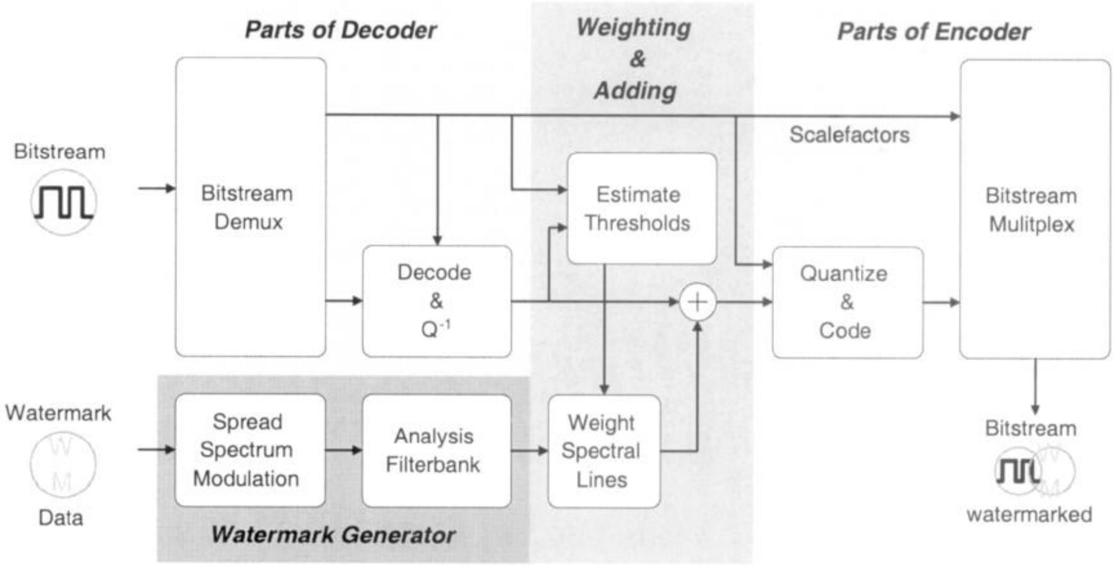

Figure 8: Block diagram of bit stream watermarking system

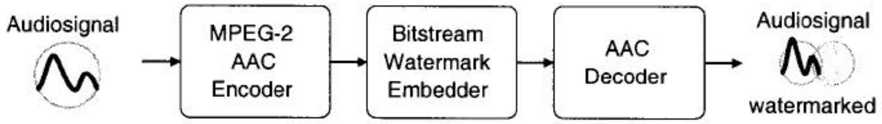

Figure 9: Block diagram of system under test

Item	length [mm:ss]	sfreq [kHz]	descr.
es01	0:11	48 k, stereo	Susan Vega
es02	0:09	48 k, stereo	German male speech
es03	0:08	48 k, stereo	English female speech
sc01	0:11	48 k, stereo	Haydn trumpet concerto
sc02	0:13	48 k, stereo	Wagner
sc03	0:12	48 k, stereo	Pop spot
si01	0:08	48 k, stereo	Harpsichord
si02	0:08	48 k, stereo	Castanets
si03	0:28	48 k, stereo	Pitch pipe
sm01	0:11	48 k, stereo	Bagpipes
sm02	0:10	48 k, stereo	Glockenspiel
sm03	0:14	48 k, stereo	Plucked strings

Table 3: Description of test items for listening tests

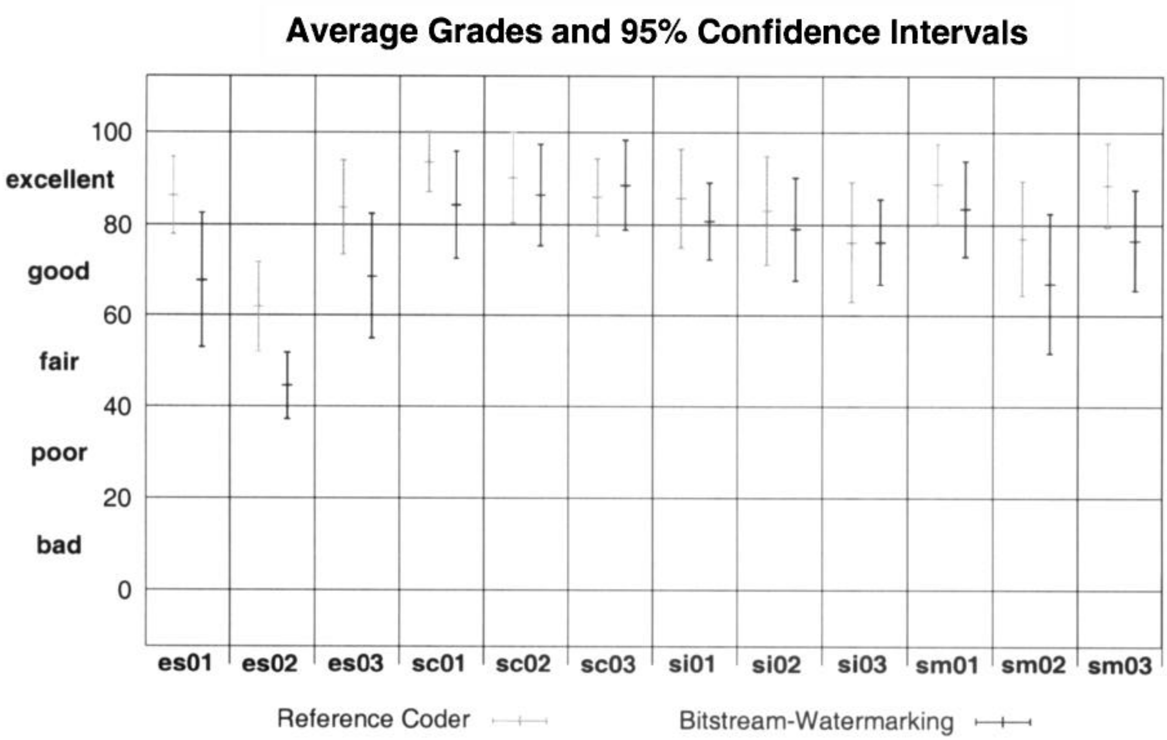

Figure 10: Results of listening test for MPEG test items

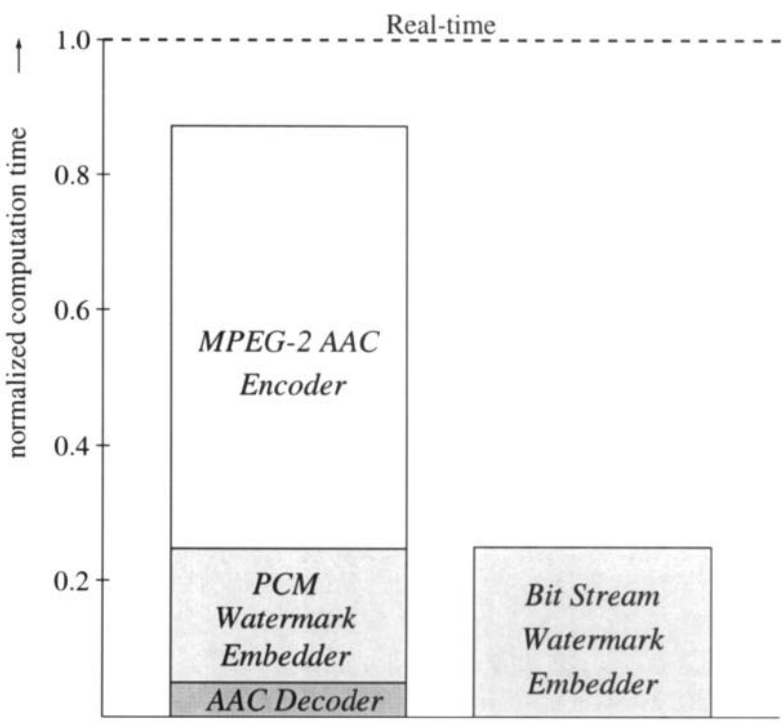

Figure 11: Comparison of complexity of bit stream watermarking, PCM watermarking and perceptual coding

Item	length [mm:ss]	sfreq [kHz]	descr.
Astor11	6:18	48 k, mono	Tango music, accordeon, violin, orchestra
AstorP13	6:30	48 k, mono	Tango music, accordeon, violin, orchestra
ChickK05	6:02	48 k, mono	Chick Korea
ChickK13	6:07	48 k, mono	Chick Korea
Jazz02	5:16	48 k, mono	Count Basie
Jazz06	5:15	48 k, mono	Count Basie
Klassik01	7:08	48 k, mono	Beethoven 9th sinfonie, 1st movement
Klassik05	8:53	48 k, mono	Beethoven 8th sinfonie, 1st movement
Klavier02	6:32	48 k, mono	Keith Jarret, CP Paris
Klavier03	5:22	48 k, mono	Keith Jarret, CP Paris
Oboe05	2:39	48 k, mono	Han de Vries , Oboe solo
Oboe06	2:24	48 k, mono	Han de Vries , Oboe solo
Synth06	3:56	48 k, mono	Synthesizer music, Ataraxia, Passport
Synth07	4:40	48 k, mono	Synthesizer music, Ataraxia, Passport

Table 4: Description of test items for watermark bit error rate measurements