TECHNICAL REPORT

ISO/IEC TR 15938-8

First edition
2002-12-15

AMENDMENT 4
2009-11-15

Information technology — Multimedia content description interface —

Part 8: Extraction and use of MPEG-7 descriptions

AMENDMENT 4: Extraction of audio features from compressed formats

Technologies de l'information — Interface de description du contenu multimédia —

Partie 8: Extraction et utilisation des descriptions MPEG-7

AMENDEMENT 4: Extraction de caractéristiques audio à partir de formats compressés

Reference number
ISO/IEC TR 15938-8:2002/Amd.4:2009(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

In exceptional circumstances, the joint technical committee may propose the publication of a Technical Report of one of the following types:

- type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;
- type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;
- type 3, when the joint technical committee has collected data of a different kind from that which is normally published as an International Standard (“state of the art”, for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Amendment 4 to ISO/IEC TR 15938-8:2002 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Information technology — Multimedia content description interface —

Part 8: Extraction and use of MPEG-7 descriptions

AMENDMENT 4: Extraction of audio features from compressed formats

After 4.8.2.2.6, add Clause 5:

5 Direct audio feature extraction from the compressed domain

5.1 Introduction

Due to efficient MPEG audio compression technologies, such as MPEG 1 – Layer III (MP3), [AMD4-1] or MPEG-2/-4 AAC, (AAC), [AMD4-2, AMD4-3] the number of personal and institutional music stored in archives grew significantly during the last years. At the same time, the need for automatic search and retrieval capabilities for music increased in order to manage these databases. These search and retrieval applications base on low-level features (e.g. described in the MPEG-7 standard [AMD4-4]) which are extracted from the digital audio content. In order to efficiently search in large archives, there is need to perform a faster low-level feature extraction. This technical report describes a method, which allows an extraction of MPEG-7 low-level features [AMD4-4] directly from the compressed domain, by transforming the frequency representation of MPEG compressed audio files into the DFT domain for feature extraction.

5.2 Conventional feature extraction

The conventional approach to obtain MPEG-7 features from compressed audio data is to decode it first and then to generate the MPEG-7 features based on the decoded time signal. But especially when searching large libraries of compressed audio files this approach can become computationally very expensive. Several works deal with the conversion between subband domain representations, especially in the field of image and video coding. In [AMD4-5], [AMD4-6] the conversion between different sizes of DCT transforms is given, having the drawback that they are restricted to non-lapped transforms. The patent in [AMD4-7] proposes a conversion method between the MDCT and the DFT domain. It is restricted to MDCT and DFT and therewith not suitable for our purposes, since we want to include also hybrid filter banks, an integral part of MP3. The architecture presented in [AMD4-8] is not restricted to the type of filter banks used. Unfortunately, the number of subbands of the different filterbanks have to be multiples of each other and this is again unsuitable for our needs. However, this paper serves as the basis for a general conversion method proposed in [AMD4-9], which can be applied to any maximally-decimated filter bank without condition on their sizes. Here, a conversion matrix is generated by multiplying the analysis with a synthesis filter bank. Principally, the same is done in this technical report, though, a universal mathematical description is used, the polyphase description introduced in [AMD4-10]. Additionally, the described method is extended by applying it to arbitrary resolution translations between synthesis and analysis filter banks in a practical way. Furthermore, it is adjusted to MP3 and AAC, and exploits some special properties of the so-called conversion matrix which is explained in the next section. In [AMD4-11] the problem of generating a complex from a real valued spectral representation is picked up from the reverse side. Therein it is said that a desired frequency response can be approximated by means of a linear combination with constant weighting factors. This approach only allows a coarse approximation, nonetheless, having a very small computational complexity load. This approach gave the inspiration for the issue termed as spectral approximation. A completely different approach is worth mentioning here which works directly on the compressed domain. It uses the MDCT coefficients as the basis for the low level feature extraction [AMD4-12]. Since there is no conversion into the DFT domain applied, this approach is restricted to the time/frequency resolution provided by the used codec. It is hence not compatible to existing MPEG-7 feature databases.

5.3 Direct feature extraction

5.3.1 System overview

In order to extract audio features from the compressed domain, we designed a conversion system which directly converts the given time-frequency representations of MPEG-1 Layer III and MPEG-2/-4 AAC into the time-frequency representation needed for calculating MPEG-7 compliant features. After applying the conversion method, the resulting complex-valued spectral coefficients are fed to the feature extraction algorithm.

Before we elaborate on the direct feature extraction system, it is important to know some details about how the conventional approach works and how it deals with compressed audio input material. Figure AMD4.1 shows the basic building blocks of the conventional feature extraction process.

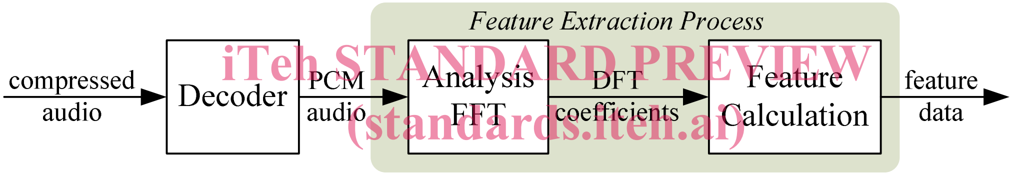


Figure AMD4.1 — Basic building blocks of the conventional feature extraction process

First, the compressed input audio material needs to be decoded to PCM audio data. Then, the feature extraction process, which consists of an analysis and a feature calculation stage, applies a window function to the PCM input samples followed by an FFT prior to the feature calculation. Our goal is to substitute the bulk of the computational amount needed for decoding and analyzing by one direct conversion process. In this context the bulk of the computational amount of the decoding process comprises basically the synthesis filter bank of the particular decoder. For MP3 additionally reordering and anti-aliasing operations take place.

We now take a look at Figure AMD4.2. The synthesis filter bank of the decoder having a transfer function and the analysis filter bank of the feature extraction process having another transfer function exhibit different numbers of subbands, K and L respectively.

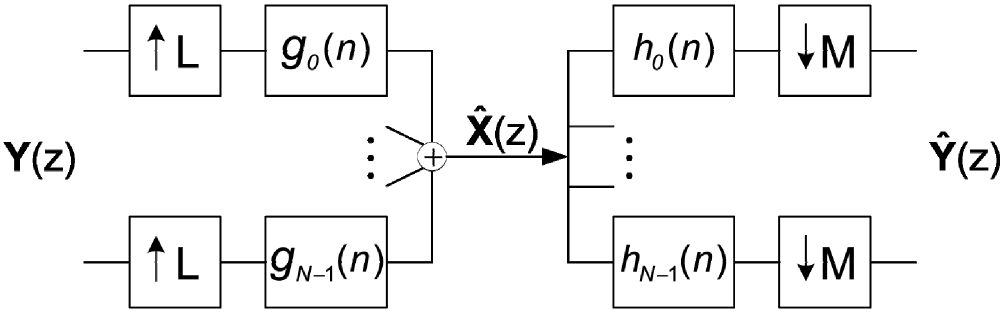


Figure AMD4.2 — Synthesis filter bank with K subbands followed by an analysis filter bank with L subbands. Both filter banks are maximally decimated and linear time-invariant

$Y_k(m)$ denotes the subband coefficient of the compressed bitstream of subband K at block m, x(n) is the decoded time audio signal at time n, and $y_l(m)$ is the subband signal of the desired domain of subband l at block m.

However, a more efficient and useful representation of maximally-decimated filter banks is the so-called polyphase description introduced by Vaidyanathan [AMD4-1]. The main advantage of the polyphase description is its mathematical compactness, so that a filter bank can be fully described by a polyphase filter matrix. The filtering process then reduces to a multiplication of a z-transformed signal vector with a polyphase filter matrix. Furthermore, a concatenation of different filter banks can be achieved by using only one polyphase matrix, which can be obtained by multiplying the individual polyphase matrices of these filter banks. This property enables the construction of a conversion matrix T(z) of size M * M as shown in Figure AMD4.3.

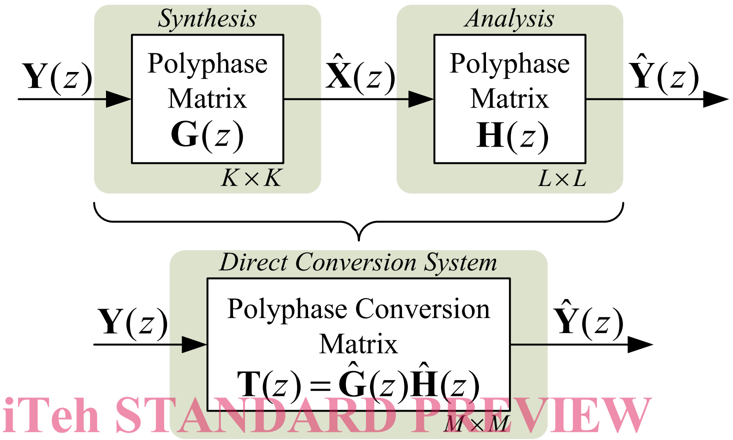


Figure AMD4.3 — Block diagram of the conventional transcoding of the direct conversion method

It is evident, that M^2 multiplications are necessary to calculate the desired spectral values when using an M*M conversion matrix. That is equivalent to a complexity of $O(N^2)$ and, unfortunately, much more complex than deploying the conventional method, since the latter uses efficient implementations of the MDCT and FFT featuring an overall complexity of O(N *log*(N)). We found, that only a fraction of the values inside a conversion matrix is necessary for the calculation of audio features, which still guarantee a successful identification of the underlying audio material. This is possible, since the most significant values of a conversion matrix are evenly spread along the main diagonal, and they decrease quickly the further we move away from it. The most important characteristic of a conversion matrix T(z) is that it exhibits a strong similarity to diagonal and therefore sparse matrices. For instance, Figure AMD4.4 shows an example of such a polyphase conversion matrix, where the white areas corresponds to zeros in the matrix. Observe that three images of matrices can be used, because each corresponds to the coefficients of a different power of z of the polyphase matrix. The analysis time window is set to 30 ms because it is suitable for many tasks of music information retrieval. The sampling frequency is chosen to be 44,1 kHz (generally it is arbitrary), hence the matrix generates 1024 complex Fourier coefficients as output, whereas it takes 576 (the content of one MP3 granule) real valued input samples.

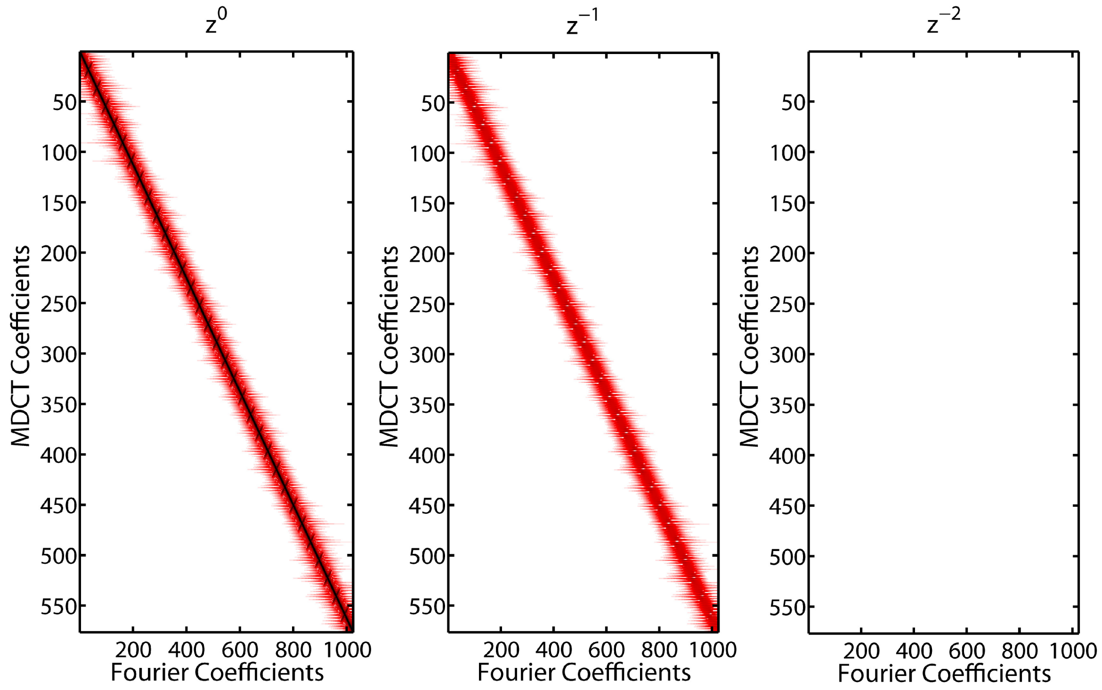


Figure AMD4.4 — Exemplary complex polyphase conversion matrix for MP3 converting one granule of 576 real valued subbands into 1024 DFT coefficients. The figure only shows absolute values.

It can be seen in Figure AMD4.4 that the most significant values are evenly spread along the main diagonal. If only the coefficients necessary for the desired accuracy are kept, the sparse matrix shown in Figure AMD4.5 is obtained. For clarification, Figure AMD4.5 shows an exemplary STFT spectrum and its approximation using sparse matrices for direct conversion. For this example a conversion complexity of about 0,07 % in contrast to a fully populated matrix was used. This property permits to approximate a desired spectral representation by only using the strongest diagonals while omitting the less important ones. Exploiting this property leads, in general, to a reduction of the computational complexity to O(N). To determine the least working complexity, we show identification results of tests performed on a large audio library with different levels of conversion complexities in a further section of this document. These tests further show that an audio feature extraction system can deal with very coarse spectral approximations.

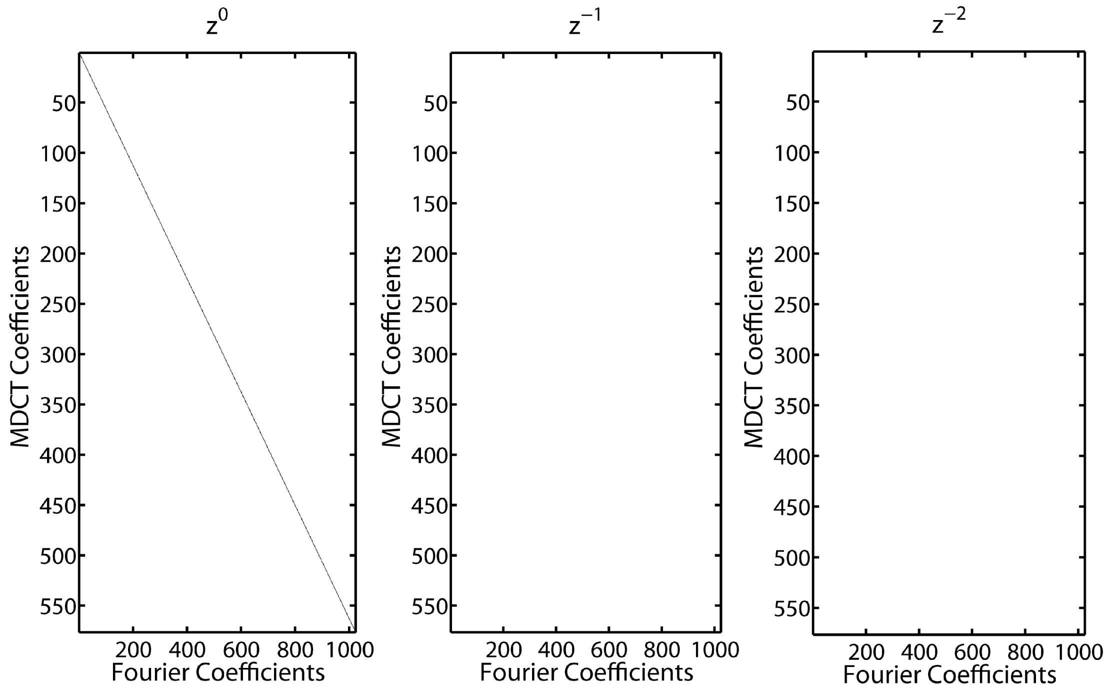


Figure AMD4.5 — Sparse polyphase matrix obtained from the conversion matrix shown in Figure AMD4.4. Only the biggest diagonal values are maintained.

5.3.2 Creation of a conversion matrix

As said before, the conversion matrix T(z) is of size M*M, which is different from those of G(Z) and H(z). But if K=L, the solution is trivial, because the size of the conversion matrix results to K=L=M. The solution for K != L is more complex, since we cannot simply multiply G(z) and H(z). This requires to extend the sizes of G(z) and H(z) to their least common multiple M. p_g = M/K and p_h = M/L, indicating how many times a matrix fits into its expanded version. For instance, given a non-overlapping synthesis filter bank, its polyphase matrix G is not a function of z. Thus, G is simply obtained using the formula

$$\hat{\mathbf{G}} = \mathbf{I}_{(p_g \times p_g)} \otimes \mathbf{G} \ ,$$

where $\otimes$ denotes the Kronecker matrix product and I(p_g x $p_{g)}$ the identity matrix of size p_g x p_g. However, this equation only holds for matrices having scalar entries, or likewise the maximum degree of O. In other words, the filter bank, represented by the polyphase matrix exhibits no overlap to consecutive blocks. For instance, this is the case for a non-overlapping DFT as used for the analysis of the feature extraction process. A general polyphase matrix, e.g. G(z) is composed according to

$$\mathbf{G}(z) = \sum_{j=0}^{J} \mathbf{G}_j z^{-j} \ ,$$

where J is the degree of the polynomials within G(z) and G(j) represents one set of coefficients of the polynomial matrix G(z) for a specific z^{-j}. Since we have an MDCT and even a QMF with different amounts of overlap on the decoder side, we need to define a more general method.

$$\hat{\mathbf{G}}(z) = \sum_{j=0}^{J} \mathbf{S}^{j}(z) \otimes \mathbf{G}_j$$

S(z) is a shift matrix that advances a block or vector by one entry (see next matrix):

$$\mathbf{S}(z) = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & & \vdots \\ \vdots & & 0 & \ddots & 0 \\ 0 & \vdots & \vdots & & 1 \\ z^{-1} & 0 & 0 & & 0 \end{bmatrix}_{p \times p}$$

Observe, if you multiply a row vector from the left then the first entry will be shifted to the second place, the second entry to the third place, and the last entry to the first place but multiplied with z^{-1}, which means it is from the previous block. Note, that $S^0(z)$ is defined as the identity matrix I. The general extension rule in G maps the coefficients of different powers of z corresponding to different time instances, i.e. different blocks of samples, to different matrix entries and vice versa. It can be seen as some kind of unfolding a polyphase matrix to be able to operate on larger block sizes. To demonstrate this, we assume a given K*K polyphase matrix G(z) having 50% overlap (analog to next equation).

$$\mathbf{G}(z) = \mathbf{G}_0 z^0 + \mathbf{G}_1 z^{-1}$$

It processes two successive blocks of size K. We now want to extend this matrix in a way, that it is able to process blocks of size 2K, where each new block consists of two concatenated blocks of size K. It is important to recognize, that the extended version G(z) now provides an overlap within the 2K-sized blocks and to one half to a succeeding one. Using the K*K or the 2K*2K polyphase matrix for calculation delivers the same results, however, the only difference is that G(z) processes blocks of twice the length. According to the rule given in G - we from now on call it extension rule – G(z) has the following shape:

$$\hat{\mathbf{G}}(z) = \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 \\ 0 & \mathbf{G}_0 \end{bmatrix} z^0 + \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{G}_1 \end{bmatrix} z^{-1}$$

Most filter banks used in today's audio coders feature adaptive window switching between windows of different lengths. In MP3 and AAC two different window lengths are used, a long and a short window. In general, long windows provide a better frequency resolution and hence a higher coding gain but can produce pre-echoes in case of occurring transient-like signal portions. These pre-echoes can be reduced and mostly avoided by using shorter block lengths. Specifically, the MP3 windows cover 12 and 36 samples, those of AAC 256 and 2048 samples, respectively. The sizes of the MP3 windows are very small compared to those of AAC. They result from cascading a 6/18-channel MDCT to each channel of a time-invariant 32-channel QMF filter bank using a fixed size window of 512 coefficients.

Two special windows - a start and a stop window - are required to realize transitions between long (L) and short blocks (S) and vice versa. The filter banks and consequently our conversion matrices can be switched between four different possible states: long to long (LL), long to short (LS), short to short (SS) and short to long (SL). Unlike MP3, AAC uses Kaiser Bessel Derived (KBD) windows in addition to sine-shaped windows, where the long blocks usually use KBD-windows and the short blocks sine-windows.

We now consider a time-varying synthesis filter bank, whose polyphase matrix has time-varying coefficients. To express this time dependency, the additional parameter *m*, denoting the time instance as block index, is introduced. Thus, G(z) becomes G(z, m, and the time signal X(z) is now obtained using the formula

$$\hat{\mathbf{X}}(z) = \mathbf{Y}(z)\,\mathbf{G}(z, m)$$

The matrix for time instance *m+1* is obtained according to the following equation.

$$\mathbf{G}(z, m+1) = \mathbf{G}_0(z, m+1)\, z^0 + \mathbf{G}(z, m)\, z^{-1}$$

T(z,m) then can be obtained by combining G(z,m) of different time instances. This procedure also holds for obtaining H(z,m). Another interpretation is, that for every time instance of m represents another time-invariant polyphase matrix. To simplify matters, we use $G_{LL}(z)$, $G_{LS}(z)$, $G_{SL}(z)$ and $G_{SS}(z)$ as those matrices which replace G(z,m) at a specific time instance m. An in-depth description how to obtain G(z,m) for MP3 and AAC, is given in the following publications [AMD4-13][AMD4-12].

5.3.3 Performance

The performance of the direct feature extraction compared to the conventional feature extraction is evaluated on the task of Audio Identification. Audio Identification is possible with the MPEG-7 compliant descriptor AudioSpectrumEnvelope. Therefore, the MPEG-7 AudioSpectrumEnvelope feature has been extracted twice: Once with the conventional method by decoding the MP3 or AAC file to wav and than performing an FFT and calculating the AudioSpectrumEnvelope feature. The second feature consisted of a direct extraction of the FFT coefficients as previously described and an estimation of the AudioSpectrumEnvelope features based on the resulting coefficients. In a first test, a suitable conversion matrix has been selected. The conversion matrix can have a scalable complexity, from very low, which enables a very fast feature extraction to very high, enabling a slower feature extraction. Therefore, 26 different complexity matrixes has been chosen, varying from 0.001% up to 0.1%, compared to a fully populated conversion matrix, were used.

Then, 6 sets of the same 775 music files dividable into 10 genres were created. The 6 sets contained the sample rates 32, 44.1 and 48 kHz and the codecs AAC and MP3 were used. The MPEG-7 AudioSpectrumEnvelope features were extracted from all files, using all complexity matrices. Then, all extracted features were fed to an audio identification algorithm, as described e.g. in ISO/IEC 15938-4:2002/Amd.2:2006 and [AMD4-18], and the identification rates were estimated. The outputs of the identification system are an index of the song having the highest similarity, and a value we call confidence, indicating the reliability of the result. The confidence is a heuristic of the system and is given in percent. In our experience a confidence above 50% indicates a correctly identified song. Figure AMD4.6 shows the results for MP3 with the sample rate of 44.1 kHz. As seen, a conversion matrix with a complexity of 0.03 % of the original conversion matrix size allows a reliable audio identification. The results for MP3 and AAC with different samplerates can be seen in [AMD4-18].

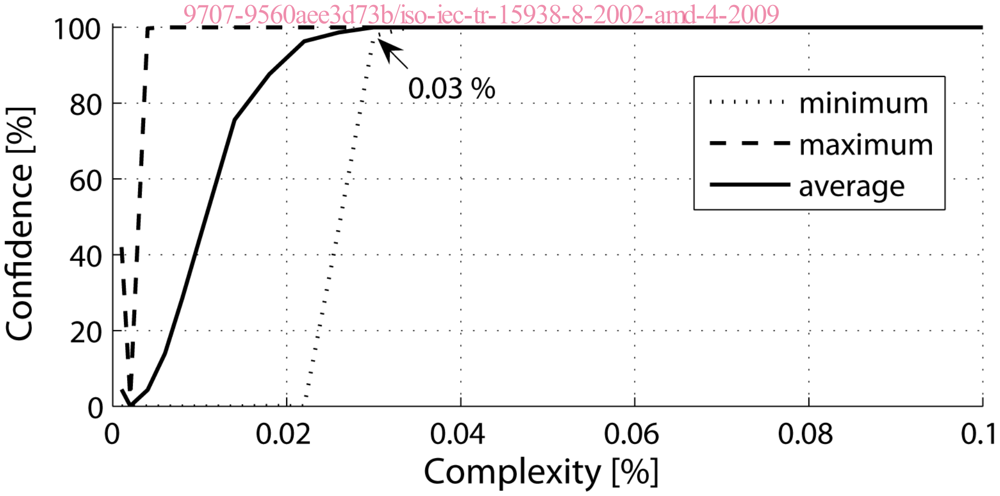
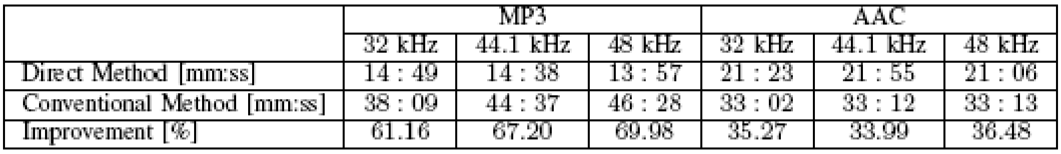


Figure AMD4.6 — Results after classification for MP3 at a sampling rate of 44.1 kHz, confidence vs. conversion complexity for the test set of 100 items.

The final extraction speed can be seen in Table AMD4.1. It shows the duration for the extraction of the whole set of music files with the direct method and with the conventional method.

Table AMD4.1 — Duration for the extraction of the whole set of music files in seconds



	MP3			AAC		
	32 kHz	44.1 kHz	48 kHz	32 kHz	44.1 kHz	48 kHz
Direct Method [mm:ss]	14 : 49	14 : 38	13 : 57	21 : 23	21 : 55	21 : 06
Conventional Method [mm:ss]	38 : 09	44 : 37	46 : 28	33 : 02	33 : 12	33 : 13
Improvement [%]	61.16	67.20	69.98	35.27	33.99	36.48

As seen in the table, the average speed improvement for feature extraction from MP3 files in approx. 66 %. The speed improvement for AAC is approx. 35 %.

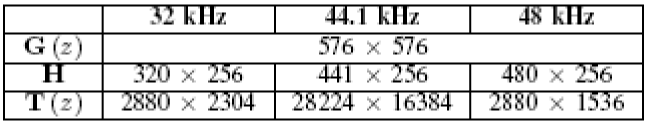
5.4 Implementation

5.4.1 Implementation Details

The practical implementation of the direct feature extraction system is explained in this section. It is useful to initially define the conditions under which we aim to operate our system. We want to be able to process MP3 as well as AAC files having the most common sampling rates, i.e. 32, 44.1 and 48 kHz. Further, the analysis of the feature extraction should use frames of 10 ms. Following the demands described in ISO/IEC 15938-4, we use a Hamming window function and calculate the FFT by means of zero padding in order to obtain an FFT size of the power of two. For instance, we round a given time-frame size of L = 320 to its next larger size of the power of two, which is particularly L=512. However, these processing steps are covered by a time-invariant polyphase matrix H(z) whose entries are furthermore scalar, since we have no overlap between consecutive blocks, i.e. the degree of the polynomials is actually 0 and H(z) reduces to H. Due to the symmetry property of the FFT we discard one half of the values. To be precisely, we need to keep L/2 + 1 FFT coefficients, but due to the negligible effect of omitting one coefficient and to stay inside sizes of the power of two, we only keep L/2 coefficients. This results in a matrix H of size L * L/2.

How to obtain the time varying synthesis matrices G(z,m) for MP3 and AAC, was shown in the previous section. The final conversion matrix T(z,m) is calculated following the method described in last sections using equations G and T. But due to the time-variance, the matrix G(z,m) can be composed of coefficients from different G(z). The bigger G(z,m) gets, the more combinations of the matrices $G_{LL}(z)$, $G_{LS}(z)$, $G_{SL}(z)$ and $G_{SS}(z)$ are thinkable. Thus, one universal conversion matrix T(z,m) meeting our specific requirements is not realizable. Obviously even without time-variance, such a conversion matrix can become very large. The next table exemplarily lists the sizes of the time-invariant decoder synthesis polyphase matrix G(z), the feature extractor analysis matrix H and the final conversion matrix T(z) for MP3 using a 10 ms analysis window length. It is important to keep in mind, that each matrix entry of T(z) contains a complex-valued polynomial of z^{-2}. Thus, in a real implementation we need to allocate memory of three times the numbers given in the next table. For instance, T(z) at a sampling rate of 44.1 kHz using a 10 ms analysis time window would consume 3 * 28224 * 16384 = 1387266048 complex numbers. If we use float precision for computation its size would reach around 10.34 GB.

Table AMD4.2 — Extraction matrix sizes for the different sample rates



	32 kHz	44.1 kHz	48 kHz
G (z)	576 × 576		
H	320 × 256	441 × 256	480 × 256
T (z)	2880 × 2304	28224 × 16384	2880 × 1536

The memory consumptions of G(z), H(z) and T(z) for different sampling rates for MP3 are given in Table AMD4.3.