SUBBAND CONVERSION FOR FEATURE EXTRACTION FROM COMPRESSED AUDIO

Tobias Friedrich, Matthias Gruhne, Gerald Schuller

Fraunhofer Institute for Digital Media Technology, Ehrenbergstr. 29, 98693 Ilmenau, Germany

We describe an efficient conversion method, which directly converts a desired spectral representation from compressed audio material. The conversion method provides a feature extraction algorithm with a suitable complex frequency representation of an audio signal. The presented conversion allows us to trade off computational complexity with accuracy. We then test several operating points with an MPEG audio feature extraction system. That leads, in general, to a reduction of the computational complexity from $O(N \log N)$ to $O(N)$, compared to the conventional method of first decoding and then applying the DFT to the resulting time domain audio signal.

Index Terms— subband conversion, polyphase description, spectrum approximation, feature extraction, matrix multiplication

1. INTRODUCTION

Nowadays, compressed audio content is ubiquitous. It is usually based on a specific, often time-varying, spectral representation of the audio signal, as is the case for MP3 and MPEG-2/4-AAC. An emerging application is the search for audio pieces using audio features. The latter are extracted from the uncompressed audio signal and are used for a search in uncompressed audio libraries. But often audio libraries contain only compressed audio signals. Hence it would be useful to obtain an audio feature extraction system which works directly on the compressed audio signals. The MPEG feature extraction uses a different spectral representation. Therefore a system for the direct conversion of the given spectral representation of the encoded signals into the desired spectral representation for the feature extraction would be useful.

2. PREVIOUS APPROACHES

Several previous approaches are dealing with conversion methods between subband domain representations, especially in the topic of image and video coding. A method for linear filtering in the transform domain is proposed in [1]. In [2, 3] the conversion between different sizes of DCT transforms is given. The patent in [4] proposes a conversion method between the MDCT and the DFT domain. Unfortunately, the amount of saved calculations is the same as for the conventional method, only the memory allocation is slightly reduced. The architecture presented in [5] does not restrict the types of the used filter banks. However, in all previously mentioned approaches the number of subbands of the different filter banks have to be multiples of each other. A general method is proposed in [6], which can be applied to any maximally-decimated filter banks without any condition on their sizes. All of those mentioned approaches show that a direct conversion is working in practice, nevertheless, the computational complexity is not reduced. [7] approaches the problem of converting the spectral representation as a linear interpolation problem with constant weighting factors. This approach allows only a coarse approximation, however, while having a very small computational complexity.

3. NEW APPROACH

Our goal is to develop a conversion system generating a spectral representation of an audio signal, which is at least exact enough to enable running a musical feature extraction algorithm successfully while having the lowest computational complexity as possible. The new approach builds on the method proposed in [6] and further extends it. We describe the filter bank as a polyphase filter matrix. A maximally-decimated filter bank can be described by the so-called polyphase description [8]. The main advantage of the polyphase description is its mathematical compactness, so that a filter bank is fully described by a polyphase filter matrix. In our notation, bold face characters denote vectors or matrices, and capital letters denote z-transformed signals. Figure 1 shows a synthesis filter bank followed by an analysis filter bank having the transfer functions $g_l(n)$ and $h_m(n)$ respectively, where $l = 0 \dots L-1$ and $m = 0 \dots M-1$. $\mathbf{Y}(z)$ denotes the row vector of the z-transformed subband signal of the compressed bitstream. $\hat{\mathbf{X}}(z)$ is the row vector of z-transforms of the polyphase components of the reconstructed time signal, and $\hat{\mathbf{Y}}(z)$ is the row vector of the z-transformed desired subband signal.

3.1. The conversion system

An overview of the conversion system is presented in Figure 2, where the upper half shows conventional transcoding and the lower half our direct conversion approach. In conven-

1-4244-1484-9/08/$25.00 ©2008 IEEE

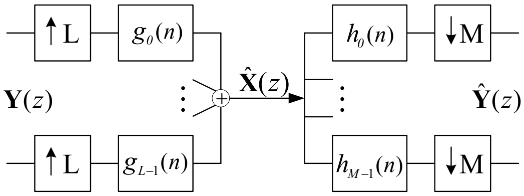

Fig. 1. Block diagram of a synthesis filter bank followed by an analysis filter bank.

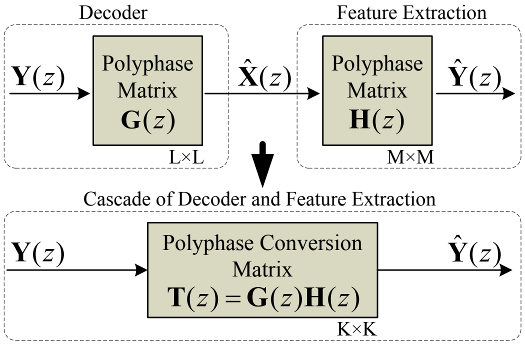
Decoder
Feature Extraction
$\mathbf{Y}(z)$
Polyphase Matrix $\mathbf{G}(z)$
L×L
$\hat{\mathbf{X}}(z)$
Polyphase Matrix $\mathbf{H}(z)$
M×M
$\hat{\mathbf{Y}}(z)$
Cascade of Decoder and Feature Extraction
$\mathbf{Y}(z)$
Polyphase Conversion Matrix
$\mathbf{T}(z) = \mathbf{G}(z)\mathbf{H}(z)$
K×K
$\hat{\mathbf{Y}}(z)$

Fig. 2. Block diagram of the conventional transcoding and of our direct conversion method.

tional transcoding, the time signal $\hat{\mathbf{X}}(z)$ is first reconstructed and then transformed into the targeted frequency representation. However, the intermediate step of calculating the time signal is not necessary in our context and can be avoided. This is achieved by our direct conversion approach, where the conversion matrix $\mathbf{T}(z)$ is the matrix product of the polyphase matrices of the synthesis filter bank $\mathbf{G}(z)$ and the analysis filter bank $\mathbf{H}(z)$. Note that if $L = M$, i.e. the polyphase matrices are of equal sizes, the solution is trivial because the size of our conversion matrix results to $K = L = M$. The solution for $L \neq M$ is more complex and requires the construction of a subband polyphase vector of a different size. For that purpose we introduce the polyphase subband vectors $\mathbf{U}(z)$ and $\mathbf{V}(z)$. The subband signal vector $\mathbf{Y}(z)$ contains the L subband signals of the compressed domain. The output of our conversion system is $\hat{\mathbf{Y}}(z)$. It contains M subband signals of the target domain, which in our example application is the DFT.

$$\mathbf{Y}(z) = \sum_{m=0}^{\infty} Y_m z^{-m} \ , \quad \hat{\mathbf{Y}}(z) = \sum_{m=0}^{\infty} \hat{Y}_m z^{-m} \qquad (1)$$

where Y_m and $\hat{Y}_m$ are the vectors of subband values at time m. $\mathbf{T}(z)$ is our conversion matrix of size $K \times K$, which converts L subband coefficients from the source domain into M subband coefficients of the target domain. K is the least common multiple of L and M. Further, p_g and p_h are the fractions of K and the number of subbands.

$$p_g = \frac{K}{L} \ , \quad p_h = \frac{K}{M} \qquad (2)$$

Since the multiplication of an $L \times L$ with a $K \times K$ matrix is not possible, we need to introduce $\mathbf{U}(z)$, which is constructed from the vector $\mathbf{Y}(z)$. $\mathbf{U}(z)$ is obtained according to

$$\mathbf{U}(z) = \sum_{m=0}^{\infty} \left[Y_{p_g m} z^{-m}, Y_{p_g m+1} z^{-m}, \ldots, Y_{p_g m+p_g-1} z^{-m} \right]. \qquad (3)$$

The output of our conversion system is also of size $K \times K$ and defined as $\mathbf{V}(z)$. The desired signal $\hat{\mathbf{Y}}(z)$ can be extracted by knowing that $\mathbf{V}(z)$ contains the subband coefficients in the following arrangement:

$$\mathbf{V}(z) = \sum_{m=0}^{\infty} \left[\hat{Y}_{p_h m} z^{-m}, \hat{Y}_{p_h m+1} z^{-m}, \ldots, \hat{Y}_{p_h m+p_h-1} z^{-m} \right] \qquad (4)$$

$\mathbf{V}(z)$ is obtained by applying the conversion matrix $\mathbf{T}(z)$ to $\mathbf{U}(z)$ according to

$$\mathbf{V}(z) = \mathbf{U}(z)\mathbf{T}(z). \qquad (5)$$

3.2. The conversion matrix T

This section answers the question of how to obtain a suitable conversion matrix $\mathbf{T}(z)$. If the synthesis polyphase matrix $\mathbf{G}(z)$ and the analysis matrix $\mathbf{H}(z)$ are of different sizes, we need to extend the transform matrices to their least common multiple of K. Their extended versions are defined as $\mathbf{A}(z)$ and $\mathbf{B}(z)$, respectively. Thus, the conversion matrix $\mathbf{T}(z)$ is obtained by simply multiplying them.

$$\mathbf{T}(z) = \mathbf{A}(z)\mathbf{B}(z) \qquad (6)$$

$\mathbf{A}(z)$ is obtained using the formula

$$\mathbf{A}(z) = \sum_{j=0}^{J} \mathbf{D}_j(z) \otimes \mathbf{G}_j, \qquad (7)$$

where $\otimes$ denotes the Kronecker matrix product. J is the degree of $\mathbf{G}(z)$, which is one for the MDCT in our case. Further note, that $\mathbf{G}_j$ represents one set of coefficients of the polynomial matrix $\mathbf{G}(z)$,

$$\mathbf{G}(z) = \sum_{j=0}^{J} \mathbf{G}_j z^{-j}. \qquad (8)$$

$\mathbf{D}_j(z)$ is a delay matrix

$$\mathbf{D}(z) = \mathbf{D}_0(z)\mathbf{S}^j(z), \qquad (9)$$

whereas $\mathbf{D}_0(z)$ is defined as

$$\mathbf{D}_0(z) = \begin{bmatrix} 0 & \cdots & & 0 & 1 \\ z^{-1} & 0 & \cdots & & 0 \\ 0 & z^{-1} & & \vdots & \\ & & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & z^{-1} & 0 \end{bmatrix}_{p\times p} . \qquad (10)$$

$\mathbf{S}(z)$ is a shift matrix that advances a block or vector by one entry (see [8]).

$$\mathbf{S}(z) = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & & \vdots \\ \vdots & & 0 & \ddots & 0 \\ 0 & \vdots & \vdots & & 1 \\ z^{-1} & 0 & 0 & & 0 \end{bmatrix}_{p\times p} \qquad (11)$$

For instance, when $p = 4$ and $\mathbf{G}(z)$ is an MDCT polyphase matrix (i.e. $J = 1$), applying equation 7 results in

$$\mathbf{A}(z) = \begin{bmatrix} \mathbf{G}_1 z^{-1} & 0 & 0 & \mathbf{G}_0 \\ \mathbf{G}_0 z^{-1} & \mathbf{G}_1 z^{-1} & 0 & 0 \\ 0 & \mathbf{G}_0 z^{-1} & \mathbf{G}_1 z^{-1} & 0 \\ 0 & 0 & \mathbf{G}_0 z^{-1} & \mathbf{G}_1 z^{-1} \end{bmatrix}, \qquad (12)$$

which has the size $K \times K$. Applying the same procedure to the M-sized $\mathbf{H}(z)$ results in the extended matrix $\mathbf{B}(z)$.

3.3. Time Varying Case

Since the spectral representation in audio coders are usually time-varying, we need to consider a time varying synthesis filter bank $\mathbf{H}(z)$ (in our case this is the inverse MDCT), whose polyphase matrices have time-varying entries. In order to express this time dependency, the parameter m, denoting the time instance, is introduced. Thus $\mathbf{G}(z)$ becomes $\mathbf{G}(z, m)$ (see [8]), and the time signal $\hat{\mathbf{X}}(z)$ is now obtained using the formula

$$\hat{\mathbf{X}}(z) = \mathbf{Y}(z)\mathbf{G}(z, m). \qquad (13)$$

The matrix containing the filter coefficients for the next time step is obtained as follows:

$$\mathbf{G}(z, m + 1) = \mathbf{G}_0(z, m + 1)z^0 + \mathbf{G}(z, m)z^{-1} \qquad (14)$$

$\mathbf{T}(z, m)$ can be calculated incorporating $\mathbf{A}(z, m)$ according to equation 6. $\mathbf{A}(z, m)$ is obtained by combining $\mathbf{G}(z, m)$ of different time steps. In the same way we obtain $\mathbf{B}(z, m)$.

3.4. Approximation

The most important characteristic of a conversion matrix is, that its components have a strong similarity to diagonal matrices. The most significant values are evenly spread along the main diagonal, whereas they decrease the further we move away from it. This property allows us to approximate our desired spectral representation by calculating the strongest diagonals while neglecting the less important. Hence to obtain a lower complexity, matrix entries with a smaller magnitude than a chosen threshold are set to zero. Depending on whether we like to have a more precise approximation or a computational efficient one, we can use more or fewer matrix entries for the calculation. Since only elements along the main diagonals are left, we obtain a computational complexity of $O(N)$. This is clearly lower than the approach of decoding/encoding using an FFT, which has a computational complexity of $O(N \log N)$ (see [4]).

4. FEATURE EXTRACTION AND CLASSIFICATION

The direct conversion from compressed domain audio files into a short time frequency domain representation can be utilized in a vast number of different applications. In order to analyze this technique satisfactorily, results of initial tests have been evaluated with an audio identification system. The approach used in this paper is similar to the system described in [9] with the major difference, that a different feature extraction method was chosen. Based on MPEG core experiments [10], it turned out, that feature extraction based on the spectral envelope is much more robust against different kinds of distortions than feature extraction methods based on spectral flatness measures. Our feature extractor is based on subbands which are logarithmically spaced using a Short Time Fourier Transform (STFT). The bandwidth of the lower bands is smaller to the bandwidth of the higher bands, since the perception of the human ear is more accurate in the lower frequencies too. The squared STFT values within each band are averaged and constitute the raw feature vector. In order to be invariant to the loudness of the signal, the raw feature vector is logarithmized and the difference to the last feature vector is taken. To compare the query feature vector against a database of feature vectors, a simple nearest neighbor classifier is chosen. This classifier returns a result list with the closest items to the query feature vector and a distance value. Since the distance value cannot be directly used for identifying the song, a confidence measure has been introduced, which is based on the differences of the ordered sequence of distance values. A jump in the sequence indicates a high confidence for the smaller distance values.

5. EVALUATION

In order to evaluate the accuracy of the direct conversion, the MPEG-7 core experiment test set has been used (compare [10]). This test set has been created to evaluate different identification algorithms and contains nine musical pieces in uncompressed format with different common western genres. For our test set, the musical pieces have been compressed into the MP3 format. The compressed files from the test set have been directly converted into the STFT domain by using a number of different thresholds for the coefficients of the conversion matrix. Altogether 20 different thresholds have been chosen and the usual feature extraction algorithm has been applied to these STFT values. The resulting $9 \cdot 20 = 180$ feature matrices have been classified and statistical confidence values computed. A confidence value above 50 % indicates that the item has been successfully identified. Figure 3 shows the results of the classification from the original and the different distortions.

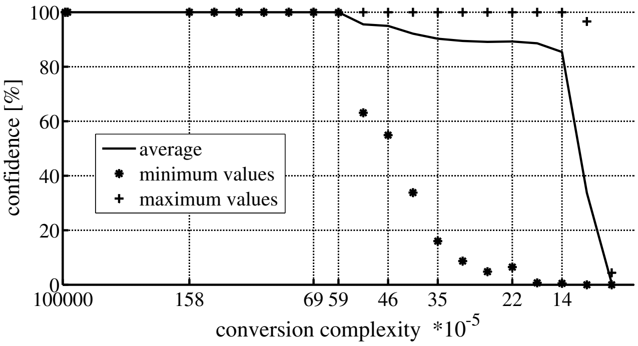

Fig. 3. Result after classification, confidence vs. conversion complexity

The conversion complexity is the fraction of the number of matrix entries unequal to zero relative to the total number of conversion matrix entries. The solid line shows the average confidences after direct conversion, extraction and classification of the original compressed files, and the symbols ”+” and ”*” describe maxima and minima respectively. A conversion complexity of 1 means the conversion matrix is unchanged with no loss of information, and the recognition rates should be as good as the conventional method (MP3 decoding and performing an FFT). Based on our results, a conversion complexity above $59 \cdot 10^{-5}$ does always reach a confidence value of 100 % and even a complexity of $46 \cdot 10^{-5}$ results in 50 % confidence for the worst items. An average confidence of about 90 % can still be reached at a complexity of $22 \cdot 10^{-5}$. But the minimum values at this threshold are very low and a successful recognition of all musical pieces cannot be guaranteed anymore. Therefore, it is recommended to add more complexity (lower the conversion threshold) for receiving good recognition rates in all circumstances.

6. CONCLUSIONS

Our goal was to obtain a direct conversion system from a given spectral representation to a desired spectral representation to reduce the required computational complexity. Our approach was to use a polyphase conversion matrix. We evaluated the resulting conversion in the context of an audio feature extraction system, and found we can reduce the computational complexity from an order of $O(N \log N)$ to $O(N)$ without reducing the recognition accuracy of the system.

7. ACKNOWLEDGEMENTS

Parts of this work are funded by the European Union under the 6th framework programme (IST-2-045081-STP).

8. REFERENCES

[1] J. B. Lee et al., “Transform domain filtering based on pipelining structure,” *IEEE Transactions on Signal Processing*, vol. 40, pp. 2061–2064, 1992.

[2] A. N. Skodras, “Direct transform to transform computation,” *IEEE Signal Processing Letters*, vol. 6, pp. 202–204, 1999.

[3] J. B. Lee et al., “2-d transform-domain resolution translation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 704–714, 2000.

[4] M. M. Goodwin, “Efficient system and method for converting between different transform-domain signal representations,” *US Patent 2003/0093282*, 2005.

[5] R. K. Sande et al., “An efficient VLSI/FPGA architecture for combining an analysis filterbank following a synthesis filterbank,” *IEEE ISCAS*, vol. 3, pp. 517–520, 2004.

[6] A. B. Touimi et al., “Efficient conversion method between subband domain representations,” *IEEE ICME*, 2005.

[7] B. Edler et al., “Arrangement and method for the generation of a complex spectral representation of a time-discrete signal,” *EU Patent 2003/0766165*, 2004.

[8] G. Schuller et al., “Modulated filter banks with arbitrary system delay: Efficient implementations and the time-varying case,” *IEEE Transactions on Signal Processing*, vol. 48, no. 3, pp. 737–748, 2000.

[9] E. E. Allamanche et al., “Content-based identification of audio material using mpeg-7 low level description,” *ISMIR, 2nd Int. Conf.*, pp. 197–204, 2001.

[10] M. Gruhne, “Core experiment on enhanced audiosignature ds,” *MPEG input paper, M12047*, 2005.