EFFICIENT PARTIAL SPECTRUM RECONSTRUCTION USING AN ASYMMETRIC PQMF ALGORITHM FOR MPEG-CODED STEREO AUDIO

Wendong Huang and Ye Wang

School of Computing, National University of Singapore
3 Science Drive, Singapore 117543
{Huangwd, wangye}@comp.nus.edu.sg

ABSTRACT

This paper presents a novel algorithm of a scalable and efficient Pseudo-Quadrature Mirror Filters (PQMF), which is employed for partial decoding a single-layer audio bitstream such as MP3, typically coded in joint/MS mode. The proposed algorithm is a new extension to our previous work on scalable audio decoding and is designed for *asymmetric* partial spectrum reconstruction (APSR), where perceptually irrelevant computations are removed. Furthermore, an efficient up-sampling operation is introduced for right channel output. The slight distortions introduced by our simple up-sampling method are inaudible according to a set of perceptual evaluations. Simulation results show that 64.6% energy savings can be achieved for a typical configuration in comparison to the standard PQMF algorithm employed by MPEG-1 audio.

1. INTRODUCTION

Energy efficiency is a critical design consideration for battery-powered mobile devices, such as mobile phones, PDAs and audio/video players, due to their limited battery capacity. With the rapid growth of multimedia processing applications, e.g., audio/video decoders, being executed on these platforms, energy efficiency methods optimized for these applications are becoming increasingly important.

Among various ways to reduce energy consumption in multimedia processing, an important approach is to reduce computational complexity by sacrificing some playback quality. In [1] , a video decoder adjusts the resolution of a frame to achieve energy savings. The attractiveness of this method stems from the fact that we can gain significant power savings at the cost of small quality degradation [2]. Moreover, in a typical application environment of portable devices, such as moving and noisy buses or trains, users are more tolerant to output quality degradation or may even not perceive it.

In audio decoding techniques, a fundamental approach to reducing computation complexity is partial spectrum reconstruction (PSR). Here, only the spectrum of a part of the coded subbands is reconstructed, resulting in a low-pass version of the original audio. Much work on PSR has been reported in the literature. In [3], general principles of PSR via digital filter banks are discussed. In [4], the design of PSR synthesis filter banks for MPEG audio is presented.

Although the PSR techniques discussed above are relevant to our work, we address the problem of reducing computational complexity in audio decoding from a different perspective. We reduce computational workload by applying asymmetric partial spectrum reconstruction (APSR) to the joint/MS stereo mode in MPEG audio, where fewer side channel subbands than middle channel subbands are decoded [2]. APSR exploits the property of joint/MS stereo mode, where the middle channel contains the most essential information from both left and right channels, and the side channel only provides the stereo image. Hence, the middle channel is perceptually more significant than the side channel even though signals from both the middle and side channels require the same computational workload to decode. Thus, APSR can effectively reduce the computational workload required for side channel signals at the cost of slight degradation of the stereo image. According to our observation, a large fraction of MP3 audio files on the market is coded in joint/MS mode. This justified the significance of APSR.

In typical asymmetric decoding, lower [0, L+M-1] subbands of the middle channel and lower [0, L-1] subbands of the side channel are used to reconstruct audio samples [2]. Processed by the modules preceding the synthesis filter bank in the MPEG audio decoder, three blocks of data are generated, namely, [0, L-1] subbands of the left and right channels, and [L, L+M-1] subbands of the middle channel, which form the input of PQMF. Since middle channel data provide common high frequency components for both the left and right channels, a straightforward way to deal with middle channel data is to add them to both the left and right channels before performing PQMF [2]. However, this method results in a significant amount of redundant and irrelevant computation. The main contribution of this paper is on how to eliminate the redundant and irrelevant

computation while maintaining the same perceptual quality of the decoded audio.

Notations: In the rest of the paper, $A_L^{m\times n}$ means matrix *A* has *m* rows and *n* columns, and it is labeled by *L*. *L* denotes either the left channel or the number of left channel subbands involved; the exact meaning can be determined from the context. The same applies for *M* and *R*. Where number of subbands is concerned, *R=L* holds. The superscript T denotes transpose.

2. CONCEPTUAL FRAMEWORK

As defined in [5], the PQMF algorithm used in MPEG audio is essentially based on cosine-modulated filter banks. Concerning the synthesis filterbank which performs PQMF at the decoder, the set of filters are derived from a single low-pass prototype filter by cosine modulation, which yields a series of frequency shifts of the prototype [8]. To reduce computational complexity, MPEG audio makes use of polyphase decomposition to implement the filterbank. Consequently, the synthesis process comprises two computationally intensive steps to reconstruct the analyzed signal, namely cosine re-modulating and polyphase subfiltering. A block diagram of the synthesis process is shown in Figure 1 [9].

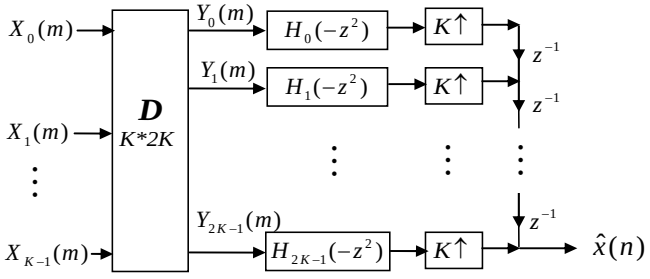

Figure 1 Structure of synthesis filter bank in MPEG-1 audio

The structure of the synthesis filter bank allows us to interpret the synthesis process in an alternative way which provides insights into the proposed APSR algorithm: the cosine re-modulation module performs frequency shifting operation on input frequency coefficients, and polyphase subfilters transfer shifted coefficients of the frequency domain into samples of the time domain. In the light of this interpretation, M data (high frequency components) are distinguishable from L and R data (low frequency components) at the output of cosine re-modulation. On the other hand, different spectral components are merged after polyphase subfiltering, and it is impossible to separate them. These two facts influence the design of our proposed approach.

The key idea of our approach is to eliminate redundant computation by enabling the right channel to share processed M data of the left channel. That is, the dimensions of the input data to the PQMFs of the left and right channels are (L+M) and R respectively. As the computational complexities of cosine re-modulation and polyphase subfiltering are $O(2\cdot K^2)$ and $O(16\cdot K)$ respectively, where *K* is the number of subbands involved, the computational workload of the synthesis process of the right channel is significantly reduced in comparison to the original scheme in [2].

An important part in the proposed APSR is the reconstruction of M data which is removed from the right channel for reducing computational workload. One possibility is to share the cosine-re-modulated M data before polyphase subfiltering as in our earlier scheme [2]. This can be easily accomplished as M data are separated from low frequency components (L data) at this stage. The polyphase subfiltering of the left and right channels can then be executed separately, and the reconstructed data of both channels are of the same sampling rate. While this scheme may be implemented with ease, the redundant computation of M data in the step of polyphase subfiltering for the right channel is not removed.

The proposed technique presented in this paper tackles above-mentioned problem. Towards this, we postpone the sharing of M data till after polyphase subfiltering. The main challenge in implementing this scheme lies in the extraction of processed M data for the right channel, since the output of the left channel contains the sum of converted L and M data which are not easily separable. In order to solve this problem, we compute first the residue between R and L data (R-L), which is used as the input of the right channel to the filterbank instead of the original R data. After the polyphase subfiltering step, the (R-L) time samples are up-sampled to yield the same sampling frequency of the left channel. As the final step, the sum of time samples from both PQMFs produces the desired right channel samples with high frequency coefficients (R+M).

To facilitate easy sampling rate conversion, we limit the subband dimension of *(L+M)* as a multiple of *R*. As a result, our proposed technique incurs computational workload close to that of processing *(2L+M)* subbands in comparison to *(2L+2M)* in our earlier scheme.

3. IMPLEMENTATION

3.1. Cosine Re-modulation

For the sake of generality, we represent cosine re-modulation in terms of the number of subbands *K* as follows:

$$Y = D^{2K\times K}\cdot X^{K\times 1} \tag{3.1}$$

The *(k,n)-th* element of the cosine re-modulation matrix $\boldsymbol{D}$ in (2.1) can be defined as in [6]:

$$d_{k,n} = \cos\left[\pi\left(\frac{k+0.5}{K}\right)\left(n+\frac{N-1}{2}+\frac{N-1-K}{K}\right)\right] \tag{3.2}$$

where N is the number of prototype filter coefficients, subject to $N=K*16+1$.

From (3.2), we can then derive the respective cosine re-modulation matrices $\boldsymbol{D}$s for the left and right channels by substituting the specified values of K.

According to (3.1), calculation of the left channel is represented as (3.3a). By partitioning the matrix $\boldsymbol{D}$ as (3.3b), we can obtain the re-modulated coefficients as (3.3c):

$$Y_L = D^{2(L+M)\times(L+M)} \cdot \begin{bmatrix} X_L \\ X_M \end{bmatrix} \quad (3.3a)$$

$$= \begin{bmatrix} D_L^{2(L+M)\times L} & D_M^{2(L+M)\times M} \end{bmatrix} \cdot \begin{bmatrix} X_L^{L\times 1} \\ X_M^{M\times 1} \end{bmatrix} \quad (3.3b)$$

$$= D_L^{2(L+M)\times L} \cdot X_L + D_M^{2(L+M)\times M} \cdot X_M \quad (3.3c)$$

For right channel input, we make use of (3.4) to re-modulate it:

$$Y_{R-L} = D^{2R\times R} \cdot (X_R - X_L) \quad (3.4)$$

As we have mentioned, the residue between R and L data is calculated, rather than the original R data, which facilitates the reconstruction of right channel samples. This can be justified as follows: Let $f_L(t)$ and $f_R(t)$ denote sample values at instant t of the left and right channels, respectively, and P denote the PQMF operation. Thus, we have:

$$f_L(t) = P\left(\begin{bmatrix} X_L \\ X_M \end{bmatrix}\right) \text{ and } f_R(t) = P\left(\begin{bmatrix} X_R \\ X_M \end{bmatrix}\right) \quad (3.5)$$

Since PQMF is a linear system [9], we have:

$$f_R(t) = P\left(\begin{bmatrix} X_L \\ X_M \end{bmatrix}\right) + P\left(\begin{bmatrix} X_R - X_L \\ 0 \end{bmatrix}\right) \quad (3.6)$$

Therefore, the desired right channel samples can be obtained from the sum of the residual samples and the left channel samples. For convenience, we denote the second item in (2.6) as $f_{R-L}(t)$.

By comparing (3.3c) and (3.4), we can see that $D^{2R\times R}$ is lower in dimension than D_L, which implies that Y_{R-L} only provides a low-pass version of $f_{R-L}(t)$. Moreover, some distortion is introduced into $f_{R-L}(t)$ by the additional up-sampling operation (see Section 3.3). Thus, our scheme only yields an approximation to $f_R(t)$. Fortunately, the distortion introduced by the up-sampling is inaudible to the human ear under appropriate profiles, which we will verify in Section 4.1.

3.2. Polyphase Subfilters

In this section, we present a generalized version of polyphase subfiltering capable of conducting calculation according to the number K of subbands involved, which is required in our scheme. For this, we need to re-design the prototype filter in terms of K, which has been proposed in [4].

The redesigned prototype filter can be decomposed into $2K$ polyphase components as follows:

$$H(z) = \sum_{i=0}^{I-1}\sum_{j=0}^{2K-1} h(2iK+j)\cdot z^{-(2iK+j)} = \sum_{j=0}^{2K-1} z^{-j} H_j(z^{2K}) \quad (3.7)$$

Based on these polyphase components, the polyphase subfiltering calculation is represented in (3.8) [9], and it accomplishes the required calculations:

$$d(z)\cdot\{z^{-K}\cdot S(0) + S(K)\}\cdot Y \quad (3.8)$$

where $d(z) = [z^{-K+1} \quad z^{-K+2} \quad \dots \quad 1]$ and

$$S(x) = \begin{bmatrix} H_{x+0}(-z^{2K}) & & & 0 \\ & H_{x+1}(-z^{2K}) & & \\ & & \ddots & \\ 0 & & & H_{x+K-1}(-z^{2K}) \end{bmatrix}$$

3.3. Up-Sampling by Repetition

A common method for up-sampling rate conversion is to employ an interpolation filter [7]. The operation can be represented as:

$$\tilde{X}_R^{(L+M)\times 1} = U^{(L+M)\times R} \cdot \hat{X}_R^{R\times 1} \quad (3.9)$$

Although the interpolation filter method can provide optimized performance, its computational complexity is very high: (3.9) leads to $(L+M)*R$ multiplications and $(L+M)*(R-1)$ additions, suggesting it contradicts our main design objective.

As discussed in Section 1, small distortions are tolerable in the application scenarios of portable devices. This allows us to exploit computationally efficient up-sampling methods. Through investigation, we choose repetition to perform up-sampling rate conversion; this choice yields satisfactory performance with very low computational complexity, especially in the case that $(L+M)$ is a multiple of R.

4. EXPERIMENTAL RESULTS

4.1. Subjective Test of Asymmetric PQMF

To evaluate the quality degradation introduced by our proposed new algorithm, we carried out experiments on a group of 10 subjects (male and female graduate students with normal hearing). For evaluation, we used five music clips selected from pop music MP3 clips. These MP3 clips were all of joint stereo mode, sampling rate 44.1KHz, and bitrates ranging from 128kbits/s to 192kbits/s. For each program, we prepared four copies for testing. These copies

were generated by our algorithm with profiles of (M:32, S:32), (M:32, S:16), (M:32, S:8) and (M:32, S:4), respectively. Each program had an additional copy with (M:32, S:32) given as references. For fairness, all test samples were arranged in random order. All subjects were asked to evaluate audio quality using the mean opinion score (MOS), which is a five-point scale (5-excellent, 4-good, 3-fair, 2-poor, and 1-bad).

	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
S:32	4.90	4.85	4.93	4.90	4.90
S:16	4.90	4.90	4.90	4.90	4.95
S:8	4.75	4.85	4.70	4.85	4.87
S:4	4.35	4.65	4.80	4.33	4.57

Table 1. Perceptual evaluation results for different APSR profiles

The result of our subjective evaluation is shown in Table 1. We can see that profiles (M:32, S:4) and (M:32, S:8) only incur slight quality degradation. Especially, the profile (M:32, S:16) is almost indistinguishable from the full decoding profile used in the standard MPEG audio decoder. These observations show that the proposed filterbank can provide satisfactory playback quality with significantly reduced computational workload.

4.2. Energy Consumption Evaluation

We evaluated the energy savings made possible with our algorithm using two different classes of audio clips: those having a bitrate of 160 kbits/sec and others having a bitrate of 128 kbits/sec. All the audio clips were of a sampling rate of 44.1K samples/sec and in joint stereo mode. Our processor model was based on a Sim-Profile configuration of the SimpleScalar instruction set simulator. We simulated the decoding of several audio clips of duration 20 secs, and measured cycle numbers required for a granule to perform the PQMF algorithm. Table 2 lists these required frequency values for the high bitrate class of clips. We obtained almost identical results for the low bitrate (128 kbits/sec) class of clips as well.

	(M:32,S:32)	(M:32,S:16)	(M:32,S:8)	(M:32,S:4)
Cycles/gr	1132400	800912	659360	609320
Freq.(MHz)	86.44	61.14	50.33	46.51

Table 2. Clock frequency (MHz) required by APSR for four different profiles

The estimated frequency can then be used to calculate energy consumption according to the relationship that it is proportional to $f^3 t$ while decoding a clip of duration *t*. Figure 2 shows the normalized energy consumption for the high bitrate class of clips. Clearly, the decoding profile (M:32, S:32) is specified in the MPEG audio standard. Compared to this baseline, the decoding profile (M:32, S:16), which presents indistinguishable playback quality, achieves energy saving of 64.6% in comparison to the standard filterbank.

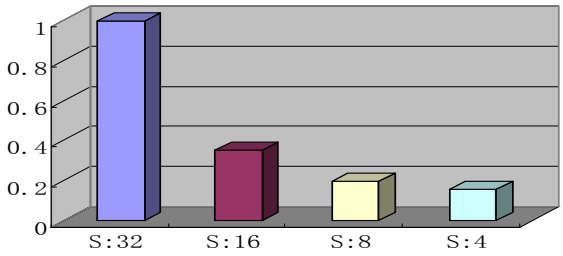

Figure 2. Normalized energy consumption for different profiles

5. CONCLUSION

We have presented an efficient and scalable PQMF algorithm for MPEG audio in joint/MS stereo mode. It is especially useful for low power audio decoding in portable devices. Based on the unequal perceptual significance of middle/side channels, the proposed scheme is also of good potential to be deployed in future media players.

6. REFERENCES

[1] M. Mesarina and Y. Turner, "Reduced energy decoding of mpeg streams", *Multimedia Systems*, 9(2),2003, pp.202–213

[2] W.Huang, Y.Wang, and S.Chakraborty, "Power-Aware Bandwidth and Stereo-Image Scalable Audio Decoding", *ACM Multimedia*, Singapore, Nov. 2005, pp.291-294

[3] T.Q. Nguyen, "Partial Spectrum Reconstruction using Digital Filter Banks", *IEEE Trans. Signal Process.*, 41(9), 1993, pp.2778-2795

[4] F. Argenti, F.Del Taglia, and E.Del Re, "Audio Decoding with Frequency and Complexity Scalability", *IEE Proc. Vis. Image Signal Process*, 149(3), 2002, pp.152-158

[5] ISO/IEC 11172-3, Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s, 1993.

[6] M.Bosi and R.E.Goldberg, "Introduction to Digital Audio Coding and Standards", *Kluwer Academic Publishers*, 2002

[7] R.E.Crochiere and L.R.Rabiner, "Multirate Digital Signal Processing", *Prentice-Hall*, 1983

[8] P.P. Vaidyanathan, "Multirate Systems and Filter Banks", *Prentice-Hall*, 1992

[9] P. S. Diniz, E. A. da Silva, and S. L. Netto, "Digital Signal Processing : System Analysis and Design", New York : Cambridge University Press, 2001