MULTI-RATE EXTENSION OF THE SCALABLE TO LOSSLESS PSPIHT AUDIO CODER

Mohammed Raad, Ian Burnett

School of Electrical,Computer and
Telecommunications Engineering,
University Of Wollongong, Australia

Alfred Mertins

University of Oldenburg
School of Mathematics and Natural Sciences
D-26111 Oldenburg, Germany

ABSTRACT

This paper extends a scalable to lossless compression scheme to allow scalability in terms of sampling rate as well as quantization resolution. The scheme presented is an extension of a perceptually scalable scheme that scales to lossless compression, producing smooth objective scalability, in terms of SNR, until lossless compression is achieved. The scheme is built around the Perceptual SPIHT algorithm, which is a modification of the SPIHT algorithm. An analysis of the expected limitations of scaling across sampling rates is given as well as lossless compression results showing the competitive performance of the presented technique.

1. INTRODUCTION

The aim of Lossless compression is the reduction of bandwidth or memory required to transmit or store the original audio signal. That is, the error between the original Pulse Code Modulated (PCM) signal and the compressed version is zero.

Lossless audio coding may be approached from a signal modeling perspective [1],[2],[3] where the signal is typically modeled using a linear predictor, which may either be FIR or IIR [2]. The aim of using a linear predictor is to decorrelate the audio samples in the time domain and to reduce the signal energy that must be entropy coded [1]. As these coding schemes rely on entropy compression, the statistics of the signal being coded have a strong influence on the performance of the coder. Compression ratios reported range between 1.4 and 5.3 [1].

Lossless audio compression has also been approached from a transform coding perspective [4][5][6]. This approach employs time to frequency transforms as de-correlation engines instead of linear predictors. In the cases of [5] and [6] the integer MDCT (IMDCT) is employed to allow the integer representation of the transform coefficients, and hence streamlining the lossless compression of these coefficients.

Recently, there has been a growing interest in the development of scalable compression schemes as well as scalable to lossless compression schemes [7][5][6][8]. Considering the advances in the bandwidth availability for cellular telephone and internet users, it is clear that a compression scheme that combines both scalability and lossless compression is of interest and potential use. The ability to smoothly scale from narrower bandwidth signals to wider bandwidth signals with different quantization resolution is also of interest, as pointed out in [8]. In this paper, we present an extension of the scalable audio coder presented in [9] to allow for the expansion of bandwidth as well as increase in quantization resolution. The scheme presented in [9] allows very fine granular scalability as well as competitive compression at the lossless stage across different bandwidths and quantization resolutions. The compression scheme is built around transform coding of audio, similar to [4], [6] and [5]. Particularly, a modified version of the Set Partitioning In Hierarchical Trees (SPIHT) algorithm [10], named Perceptual SPIHT (PSPIHT), is used to allow scalability as well as perfect reconstruction. The use of PSPIHT and SPIHT allows the coder to quantize the transform coefficients in such a manner that only the input audio segment's statistics are required, avoiding the necessity to design dedicated entropy code books. Scalability in bandwidth is obtained through the scalable transmission of the error between the wider bandwidth signal and the narrower bandwidth signal in the frequency and time domains.

This paper is organized as follows. Section 2 describes the different components of the scalable-to-lossless scheme. Section 3 gives an outline of the bandwidth scalable scheme as well as an analysis of expected performance. Section 4 presents the lossless results obtained across four bandwidths, and Section 5 provides a brief conclusion.

2. THE SCALABLE TO LOSSLESS SCHEME

The scalable to lossless scheme presented in [9] is the basis upon which the bandwidth scalable coder is built, as such it will be described first. Figure 1 illustrates the PSPIHT scalable to lossless scheme. It consists of the combination of the lossy coder presented in [11], which is based on the Modulated Lapped Transform (MLT) and SPIHT, and a lossless coder for transmitting the error incurred from the lossy part. The lossy part is given by the right half of the structure in Fig. 1, and the error coding (if present) takes place in the left half. Note that both parts of the coder are based on the SPIHT algorithm. In this section we mainly focus on the lossy part of the structure, referred to as MLT-PSPIHT.

The input signal is transformed using the MLT where floating point calculations are used. The transform coefficients are encoded using PSPIHT, and the bitstream is transmitted to the decoder. We will refer to this bitstream as $bst1$ which is further divided into $bst1a$ and $bst1b$ by PSPIHT. This second stage division aims to separate perceptually significant coefficients from perceptually insignificant coefficients such that $bst1a$ contains the perceptually significant coefficients and is transmitted before $bst1b$. $bst1$ is decoded at the encoder and the synthesized audio is subtracted from the original audio to obtain the output error. Here integer operations are used, so that the error output is integer and has a dynamic range that is typically less than that of the original integer signal. The time-domain error signal is then encoded into Bitstream $bst2$, using SPIHT. At the decoder, both bitstreams are received as part of one global bitstream, with $bst1$ making up the first part of the total bitstream for this section of the scheme. The decoder may decode up to any rate desired.

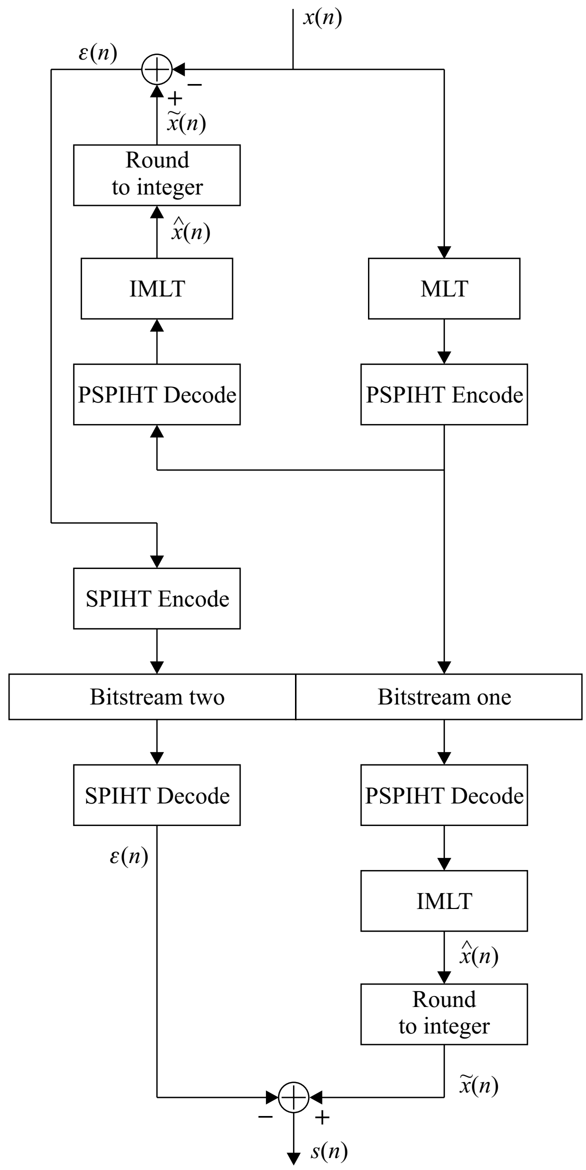

Fig. 1. The scalable-to-lossless scheme based on SPIHT and PSPIHT

2.1. The PSPIHT algorithm

SPIHT [10] is a coding algorithm that allows the transmission of coefficients in a pseudo-sorted fashion where the most significant bits of the largest coefficients are sent first. The sorting is carried out according to the magnitudes of the coefficients. The generated bitstream is fully embedded, allowing optimal reduction of coding noise with every additional bit sent [10]. It can be truncated at any point to achieve the best reconstruction for the actual number of bits sent. The original design of SPIHT was aimed at image compression, and the intent was to use the algorithm in the frequency domain [10]. However, the algorithm may also be used in the time domain.

PSPIHT is a modification of SPIHT in the frequency domain that allows the transmission of perceptually significant coefficients ahead of perceptually insignificant coefficients whilst quantizing both sets of coefficients with the same resolution. Such an algorithm can maintain the potential for lossless synthesis as energy significant spectral components, that are perceptually insignificant, are not distorted more than perceptually significant spectral components. The modification focuses on introducing a perceptual significance test to allow the required bitstream formatting. The perceptual significance test is based on the perceptual entropy of the given coefficient as determined in [12]. For PSPIHT a few new definitions are added to those used by SPIHT and listed in [10]. Firstly, v_{pe} is defined to be a binary vector with perceptual significance information for the sub-band coefficients. That is, if $v_{pe}(n) = 1$ then coefficient n is perceptually significant, otherwise it is perceptually insignificant. Also, $LPISP$ is defined to be the list of perceptually insignificant, but energy significant coefficients. That is, $LPISP$ contains pointers to coefficients that are significant in terms of energy (or magnitude) but lie in spectral bands that contain other more significant coefficients which have masked them. Finally, the perceptually significant component of Bitstream one is labelled as $bst1a$ and the perceptually insignificant component as $bst1b$. Fixed limits can be set for the size of $bst1a$ and $bst1b$. The complete algorithm is listed in [9] and so will not be listed here. The operation of PSPIHT differs primarily from SPIHT in the application of a perceptual significance test after the energy significance test.

In the sorting pass, the energy significance test is maintained as the first test. Sorting bits are sent to $bst1a$ until an energy significant coefficient is encountered. This coefficient is then tested for perceptual significance by checking the corresponding entry in v_{pe}, if the coefficient is found to be significant (and $bst1a$ is not full) then the sign bit and further refinement bits are sent to $bst1a$, otherwise these bits are sent to $bst1b$. The perceptual significance test is only applied to individual coefficients and not to whole sets as is the energy significance test. The same process is followed at the decoder which obtains the test results (the sorting information), sign bits and significant bits from the bitstream. Note that the major task of the algorithm is to re-arrange the bitstream produced so that it reflects perceptual significance, allowing more perceptually accurate synthesis at lower rates. Some extra overhead is encountered in the bitstream formatting as a pointer must also be transmitted indicating the length of $bst1a$. This is necessary for the decoder to be able to divide the total bitstream correctly and to allow $bst1a$ to be less than its hard-coded maximum length, should the signal contain fewer significant components than expected. Although the listed algorithm outputs perceptual significance information it does so only for energy significant components and even then only when there is space in $bst1a$, hence it would be rare to encounter a situation where all of v_{pe} is transmitted.

3. EXTENDING THE SCALABLE TO LOSSLESS CODER

The scalable to lossless coder presented in Section 2 may be expanded to achieve scalability in terms of sampling frequency (and thus bandwidth) as well as quantization resolution. In this paper we consider the sampling frequencies 44.1, 48, 96 and 192 kHz with quantization resolutions of 16, 20 and 24 bits. It should be noted that the 192 kHz sampled signals may be considered synthetic data as they are up-sampled versions of the 96 kHz sampled signals. One may approach this form of scalability from two opposing perspectives; a top-down approach which dictates that the highest sampled signal be coded losslessly hence allowing the lower sampled, coarser quantized signals to be extracted from it through the use of bandlimiting and quantization resolution reduction. This approach is adopted in [6]. On the other hand, one may take a continuous refinement approach that codes the lower sampled, coarser quantized signals first and scales (continuously refining) until the higher sampled signals are also losslessly represented.

Since the scalable to lossless coding scheme proposed in [9] produced lossless compression results competitive with the state of the art, it is adopted as the compression scheme for signals with a sampling frequency of 44.1 kHz and quantization resolution of 16 bits per sample. In other words, the continuous refinement approach is adopted in this paper.

3.1. An analysis of bandwidth scalable compression

Before describing how bandwidth scalability may be achieved, it is useful to consider the possible limitations that scaling in frequency may place on the compression ratios achieved. Let b_{01} be the number of bits per sample used to quantize a given signal (x_1) that is sampled at f_1 Hz. Similarly, let b_{02} be the number of bits per sample used for another version of the signal (x_2) sampled at f_2 Hz with $f_2 > f_1$ and $b_{02} \geq b_{01}$. Also, let the compressed versions of these signals be represented at b_{11} and b_{12} bits per sample. The compression ratios of each signal are given by $\alpha_1 = \frac{b_{01}}{b_{11}}$ and $\alpha_2 = \frac{b_{02}}{b_{12}}$.

Using the continuous refinement approach means that the total number of bits used to compress the higher sampled signal is a sum of the total number of bits used to compress the lower sampled signal as well as the number of bits spent coding the resulting error signal. The total number of bits used to code the higher sampled signal, per second, may thus be expressed as:

$$B_2 = B_1 + B_e \tag{1}$$

where B_1, B_2, and B_e are the total number of bits used to code x_1, x_2 and the residual signal e required to expand x_1 to x_2. Now, $B_1 = b_{11} \times f_1$ and hence,

$$\begin{aligned} b_{12} \times f_2 &= b_{11} \times f_1 + b_e \times f_2 \\ \frac{b_{02}}{\alpha_2} \times f_2 &= \frac{b_{01}}{\alpha_1} \times f_1 + b_e \times f_2 \end{aligned}$$

Assuming that the compression ratio is to be kept constant at α, then

$$\begin{aligned} (b_{12} - b_e)f_2 &= b_{11}f_1 \\ \frac{b_{02}}{\alpha} - b_e &= \frac{b_{01}}{\alpha}\frac{f_1}{f_2} \\ \therefore b_e &= \frac{1}{\alpha}(b_{02} - b_{01}\frac{f_1}{f_2}) \end{aligned} \tag{2}$$

The foregoing analysis indicates that as the differences in the sampling frequency and quantization resolution increase, more bits may be spent on the residual signal whilst maintaining a constant compression ratio. That is, if x_1 is sampled at a significantly lower frequency to x_2 and the quantization resolution between the two signals is also significant then one may spend more bits on b_e whilst maintaining a constant compression ratio.

3.2. The bandwidth scalable coder

Figure 2 shows the proposed technique for scaling in frequency and quantization resolution. Signal x_1 is losslessly coded before any bits are transmitted that allow the higher sampled frequency to be reconstructed. Thus, both the encoder and decoder have the complete x_1 signal. The MLT (popularly known as the MDCT) is used to transform x_1, the obtained coefficients are then zero padded and scaled. The scaling factor may be one if the MLT coefficients are originally normalized, otherwise it is given by $k = \sqrt{\frac{f_2}{f_1}}$. Whilst this results in a good approximation of the frequency domain representation of x_2, there is some error that must be accounted for. Figure 3 shows an example frame that has been sampled at 96 kHz. The MLT coefficients of the frame are approximated by zero padding and scaling the coefficients of the 48 kHz sampled version of the frame. The error is also shown on two scales, the left hand scale shows the detail in the error whilst the right hand scale compares the magnitude of the approximation error to that of the actual MLT coefficients. It is notable that the error is considerably smaller in magnitude than the original set of coefficients which means it is easier to code losslessly (due to the smaller dynamic range required).

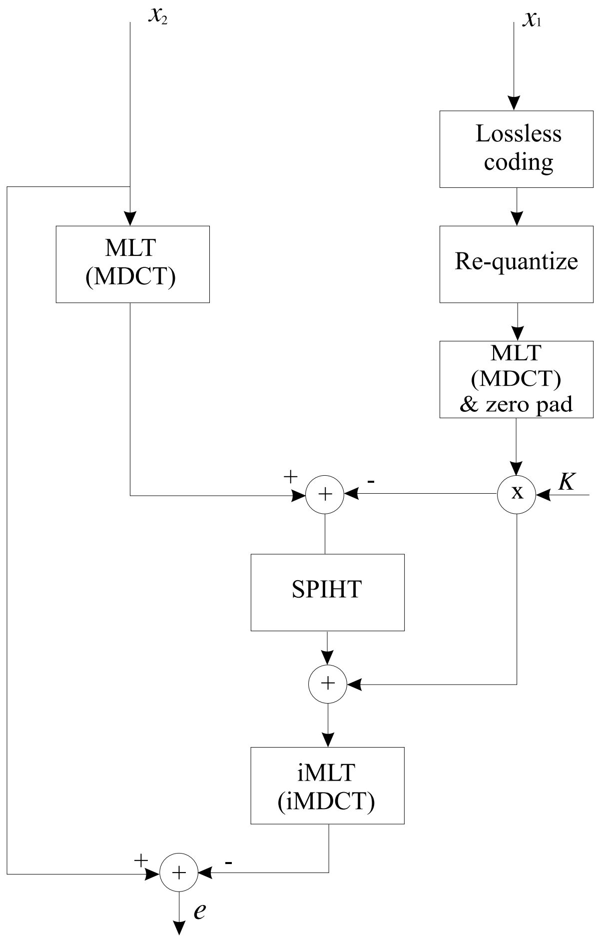

Fig. 2. The proposed scaling in bandwidth and resolution scheme

The error coefficients are coded using SPIHT and then added to the approximated coefficients. This new set of coefficients then provides a better approximation of the actual MLT coefficients and so when the inverse MLT is applied, the time-domain error signal is small. The time domain error signal is calculated in integer form and transmitted via the use of SPIHT leading to the lossless representation of x_2.

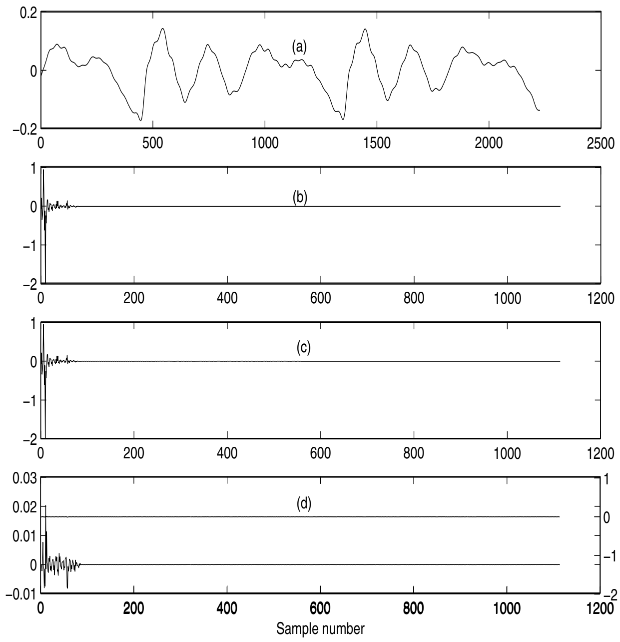

Fig. 3. (a) The original 96 kHz sampled signal (b) The MLT coefficients at 96 kHz (c) The approximated 96 kHz coefficients (d) The error in approximation

Table 1. objective results for 96 kHz sampled files

	Content (compression ratio, bits per sample)	
Bits, Sampling (kHz)	violin	vocal
16, 44.1	2.67, 6.0	3.70, 4.33
20, 48	1.64, 12.2	2.01, 9.93
24, 96	1.84, 13.05	1.99, 12.05
24, 192	2.13, 11.29	2.27, 10.53

4. RESULTS

Lossless compression results are presented in this section at sampling rates 44.1, 48, 96 and 192 kHz. The subjective performance of the PSPIHT coder has been previously reported to be comparable with that of the MPEG AAC (VM) at 16, 32 and 64 kbps [9]. Similarly, the smooth objective scalability that may be achieved when SPIHT is employed was described in that paper.

The lossless results presented here are for two files originally recorded at 96 kHz sampling rate and 24 bits resolution. These files were then resampled to obtain higher and lower sampled versions. As such, the 192 kHz signal may only be considered a synthetic test signal. The content of the test signals as well as the obtained lossless compression results are listed in Table 1. The lossless compression results are competitive with the state of the art at 16 bits, 44.1 kHz (the CD standard) [1]. Also, as the bandwidth increases the compression ratio for both signals varies in agreement with the behaviour predicted by the analysis in Section 3. Namely, the reduction in compression ratio decreases as the difference in sampling frequency (i.e. $f_2 - f_1$)and quantization resolution ($b_{02} - b_{01}$) increases. It is notable that the compression ratios obtained are also competitive with the state of the art.

5. CONCLUSION

A scalable to lossless compression scheme that scales in bandwidth and quantization resolution has been presented. The scheme is based on the PSPIHT and SPIHT algorithms as well as the MLT. The PSPIHT algorithm is a modified version of the SPIHT algorithm that allows the transmission of perceptually significant coefficients in a set sorted manner whilst maintaining the same quantization resolution for perceptually insignificant coefficients. This allows energy significant components of the signal to be maintained at higher rates. Bandwidth scalability was achieved by approximating the MLT coefficients of the higher sampled signal from the losslessly decoded lower sampled signal and then transmitting the resulting error in the frequency domain as well as the time domain. The lossless performance was shown to be better maintained if the difference in the sampling rate and quantization resolutions of the signals was not trivial. The lossless compression results presented showed this to be true. The presented results also showed that competitive lossless compression may be achieved using the proposed scheme at a number of sampling frequencies and quantization resolutions.

6. REFERENCES

[1] M. Hans and R.W. Schafer, “Lossless compression of digital audio,” *IEEE Signal Processing magazine*, vol. 18, no. 4, pp. 21–32, July 2001.

[2] P.G. Craven and M.J. Law, “Lossless compression using IIR prediction filters,” AES 102nd convention, AES preprint 4415, March 1997.

[3] A.A.M.L. Bruekers, W.J. Oomen, and R.J. van der Vleuten, “Lossless coding for DVD audio,” AES 101st convention, AES preprint 4358, November 1996.

[4] T. Liebchen, M. Purat, and P. Noll, “Lossless transform coding of audio signals,” *Proceedings of the 102nd AES convention, AES preprint 4414*, March 1997.

[5] R. Geiger, J. Herre, J. Koller, and K. Brandenburg, “Intmdct - a link between perceptual and lossless audio coding,” *Proceedings of ICASSP-2002*, vol. 2, pp. 1813–1816, May 2002.

[6] J. Li and J. D. Johnston, “A progressive to lossless embedded audio coder (PLEAC) with multiple factorization reversible transform,” *ISO/IEC JTC1/SC29/WG11 M9136*, December 2002.

[7] T.S. Verma, *A perceptually based audio signal model with application to scalable audio compression*, Ph.D. thesis, Department of Electrical Engineering, Stanford university, October 1999.

[8] T. Moriya, “Report of AHG on issues in lossless audio coding,” ISO/IEC JTC1/SC29/WG11 M7955, March 2002.

[9] M. Raad, A. Mertins, and I. Burnett, “Scalable to lossless audio compression based on perceptual set partitioning in hierarchical trees (PSPIHT).,” in *Accepted for publication in ICASSP03*, 2003.

[10] Amir Said and William A. Pearlman, “A new, fast, and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Transactions on Circuits and Systems For Video Technology*, vol. 6, no. 3, pp. 243–250, June 1996.

[11] M. Raad, A. Mertins, and I. Burnett, “Audio compression using the MLT and SPIHT,” *Proceedings of DSPCS' 02*, pp. 128–132, 2002.

[12] J.D. Johnston, “Estimation of perceptual entropy using noise masking criteria,” *Proceedings of ICASSP-88*, vol. 5, pp. 2524–2527, 1988.