MULTI-RATE AND MULTI-RESOLUTION SCALABLE TO LOSSLESS AUDIO COMPRESSION USING PSPIHT

Mohammed Raad, Ian Burnett

School of Electrical, Computer and Telecommunications Engineering
University Of Wollongong,
Northfields Ave Wollongong NSW 2522, Australia

Alfred Mertins

University of Oldenburg
Institute of Physics
D-26111 Oldenburg, Germany

ABSTRACT

This paper presents a scalable to lossless compression scheme that allows scalability in terms of sampling rate as well as quantization resolution. The scheme presented is perceptually scalable and it also allows lossless compression. The scheme produces 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. Objective and subjective results are given that show perceptual as well as objective scalability. The subjective results given also show that the proposed scheme performs comparably with the MPEG-4 AAC coder at 16, 32 and 64 kbps.

1. INTRODUCTION

Currently, lossless audio coding has been approached from a signal modeling perspective [1],[2],[3]. 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 coded [1]. The compression ratio of such coders typically depends on the nature of the audio signal being coded. Values reported range between 1.4 and 5.3 [1].

Similarly, scalable audio compression has been approached from a signal model point of view. Recent scalable coding schemes, such as the one described in [4], use a composite signal model. The model is built through the combination of Sinusoids, Transients and Noise (STN).

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. For example, MPEG have started a process of standardization for such a scheme [5]. 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 [5]. In this paper, we present a scalable audio coder that 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 [6], [7] and [8]. Particularly, a modified version of the Set Partitioning In Hierarchical Trees (SPIHT) algorithm [9], named Perceptual SPIHT (PSPIHT), is used to allow scalability as well as perfect reconstruction. A Multistage application of SPIHT in the time domain is used to allow scalability between different bandwidths and quantization resolution. 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.

This paper is organized as follows. Section 2 describes the different components of the proposed scalable-to-lossless scheme. Section 3 gives a brief outline of the SPIHT algorithm as well as a complete listing of PSPIHT. Section 4 presents the lossless and scalable-to-lossless results obtained, and Section 5 provides a brief conclusion.

2. THE SCALABLE TO LOSSLESS SCHEME

The overall structure of the coder proposed in this paper is depicted in Fig. 1. The presented structure is an expansion of that presented in [10]. The proposed scheme is a multi-stage application of SPIHT and PSPIHT. The first stage is the extraction of narrower bandwidth and coarser resolution signals (x_k) from the original signal x_o. These signals are arranged such that x_k has a narrower bandwidth and coarser or equal quantization resolution as x_{k+1}. The synthesized signal at a given bandwidth (that is less than the original), denoted as s_k, is interpolated and re-quantized to have the same bandwidth and quantization resolution of the next stage signal x_{k+1}. This allows the calculation of the interpolation and quantization error in the time domain which is transmitted by the use of SPIHT. Such an approach allows a signal sampled at B kHz to be compressed in a manner that allows one to reconstruct, in a lossless manner, the same signal sampled at B' kHz, where $B' \leq B$. It also allows the lossless extraction of coarser quantized versions of the original signal whilst maintaining the use of an embedded bitstream that ensures every new bit transmitted adds some information to the synthesized signal. To appreciate how this is achieved, a closer look at the compression of the narrowest bandwidth signal (x_1) follows.

Figure 2 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 made by the lossy part. The lossy part is given by the right half of the structure in Fig. 2, 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 $bst2$,

0-7803-7946-2/03/$17.00 ©2003 IEEE

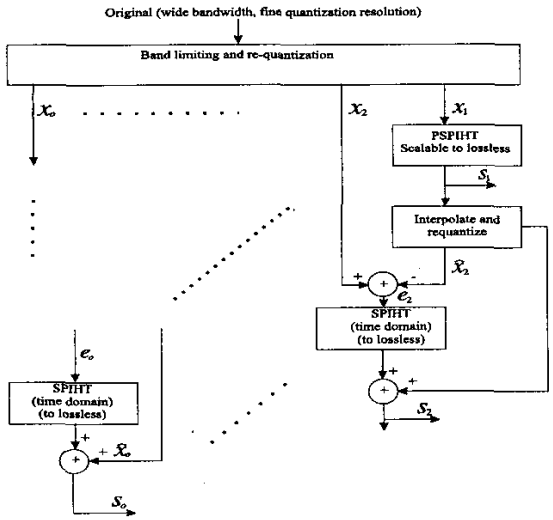

Fig. 1. The overall structure of the proposed coder

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 remaining parts of the global bitstream are made up of $bste_2$, $bste_3$ and so on. The decoder may decode up to any rate desired. If $bst1$ containing the transform coefficients is exhausted, then the decoder recognizes that the remaining bitstream represents the time-domain error signals, which it reconstructs and adds to the synthesized signal.

3. THE PSPIHT ALGORITHM

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 [9]. Let v_{pe} 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 let $LPISP$ 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, denote the perceptually significant component of Bitstream one 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 below.

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

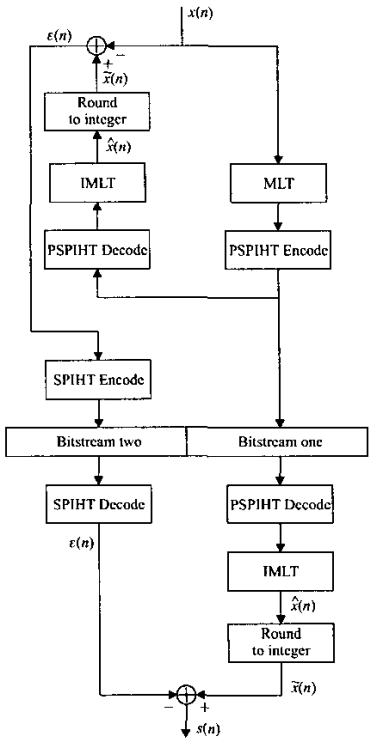

Fig. 2. The scalable-to-lossless scheme based on SPIHT and PSPIHT

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.

Algorithm PSPIHT:

1) **Initialization:** output $n = \lfloor log_2(max_i(|c_i|)) \rfloor$;
   set the LSP as an empty list, and add the coordinates $(i) \in H$ to the LIP, and only those with descendants also to the LIS, as type A entries.
   Set LPISP as an empty set.

2)**Sorting Pass:**
   2.1) for each entry (i) in the LIP do:
      If $bst1a$ is not full
         2.1.1) output $S_n(i)$ to $bst1a$
      Else
         2.1.1) output $S_n(i)$ to $bst1b$
      2.1.2) if $S_n(i) = 1$ and $bst1a$ is not full then output $v_{pe}(i)$ to $bst1a$

else if $bst1a$ is full
do not output $v_{pe}(i)$ move (i) to LPISP
and output the sign of c_i to $bst1b$;
If $v_{pe}(i) = 1$ then move (i) to the LSP
If $bst1a$ is not full
output the sign of c_i to $bst1a$;
Else output the sign of c_i to $bst1b$;
Else move (i) to the LPISP and output
the sign of c_i to $bst1b$

2.2) for each entry (i) in the LIS do:
2.2.1) if the entry is of type A then
If $bst1a$ is not full
- output $S_n(D(i))$ to $bst1a$

Else
- output $S_n(D(i))$ to $bst1b$
- if $S_n(D(i)) = 1$ then
  * for each $(k) \in O(i)$ do:
    If $bst1a$ is not full
    - output $S_n(k)$ to $bst1a$

    Else
    - output $S_n(k)$ to $bst1b$
    - if $S_n(k) = 1$
      If $bst1a$ is not full
      output $v_{pe}(k)$ to $bst1a$
      Else do not output $v_{pe}(k)$,
      move (k) to the LPISP
      and output the sign of c_i to $bst1b$
      If $v_{pe}(k) = 1$ then
      add (k) to the LSP
      If $bst1a$ is not full
      output the sign of c_k to $bst1a$
      Else output the sign of c_k to $bst1b$
      Else add (k) to the LPISP and
      output the sign of c_k to $bst1b$
    - if $S_n(k) = 0$ then add (k) to the
      end of the LIP;
  * if $L(i) \neq \emptyset$ then move (i) to the
    end of the LIS as an entry of type B,
    and go to Step **2.2.2)**; otherwise, remove
    entry (i) from the LIS;

2.2.2) if the entry is of type B then
If $bst1a$ is not full
- output $S_n(L(i))$ to $bst1a$
- output $S_n(L(i))$ to $bst1b$
- if $S_n(L(i)) = 1$
  * add each $(k) \in O(i)$ to the end of
    the LIS as an entry of type A;
  * remove (i) from the LIS.

3) **Refinement Pass:** for each entry (i) in the LSP,
except those included in the last sorting pass (i.e., with
same n), output the nth most significant bit of $|c_i|$
to $bst1a$ if it is not full,
otherwise output the nth most significant bit of $|c_i|$ to $bst1b$
for each entry (i) in the LPISP,
except those included in the last sorting pass (i.e., with
same n), output the nth most significant bit of $|c_i|$ to $bst1b$

4) **Quantization-Step Update:** decrement n by 1 and go
to step **2.**

Table 1. objective results for 96 kHz sampled files

	Content (compression ratio, bits per sample)	
Bits, Sampling (kHz)	violin	vocal
16, 44.1	2.66, 6.02	3.52, 4.55
20, 48	1.59, 12.57	1.87, 10.72
24, 96	1.41, 17.07	1.54, 15.54

Table 2. The subjective comparison scale used

Score	Subjective opinion
1	A much better than B
2	A better than B
3	A and B are the same
4	A worse than B
5	A much worse than B

4. RESULTS

Objective and subjective results are presented in this section. The objective results illustrate the objective scalability of the coder and the lossless compression performance of the proposed scheme. The objective results presented here are for two files digitized at 96 kHz sampling rate and 24 bits resolution. The content of the test signals as well as the obtained lossless compression results are listed in Table 1. The lossless compression results are impressive at 16 bits, 44.1 kHz (the CD standard) and are competitive with the state of the art [1]. However, as the bandwidth increases the compression ratio for both signals decreases. This decrease is primarily due to the increase in meaningful information that the time domain error signals e_k hold (resulting in a greater dynamic range and less noise-like behavior). Whilst the final compression ratios at the wider bandwidths are lower than one would aim for, the advantage of the proposed scheme may be better appreciated from Fig. 3 that displays the SNR of a single frame of the violin file at various rates (with 96 kHz sampling). The figure illustrates the smooth scalability that this scheme provides until lossless representation is achieved. At lower rates than those shown on the figure, narrower bandwidth versions of the signal are synthesized losslessly. For the frame used to generate the results of Fig. 3, a rate of 362.5 kbps (7.55 bits per sample) produced lossless compression for the 48 kHz version of the signal whilst a rate of 142 kbps (3.22 bits per sample) produced a lossless copy of the 44.1 kHz sampled version of the signal. All three lossless signals are obtained from the same bitstream.

The subjective results compare the lossy part of the proposed scheme to the MPEG-4 AAC (VM) coder at 16, 32 and 64 kbps. The synthesized audio of PSPIHT based schemes at these rates is also compared subjectively to illustrate the subjective scalability. The implementation used to obtain the presented results limits $bst1$ to 96 kbps, and $bst1a$ to 64 kbps. The MLT coefficients are quantized using an 18 bit uniform quantizer. SQAM files, obtained from [13] and digitized at 44.1 kHz 16 bits per sample (which is the format on which the lossy part of the coder operates), were used for these tests.

A total of 39 subjects were asked to listen to two differently coded versions of the same sound (labelled A and B) and indicate which version was more preferable. A scale of 1 to 5 was used as indicated in Table 2. The comparison tests involved comparing the AAC coder and the PSPIHT coder at 16, 32 and 64 kbps. The PSPIHT coder at lower rates has also been tested against the PSPIHT coder at higher rates. Specifically, the PSPIHT coder at 16 kbps was compared to the PSPIHT coder at 32 kbps and the latter compared to the PSPIHT coder at 64 kbps.

The mean scores for all the comparisons have been obtained

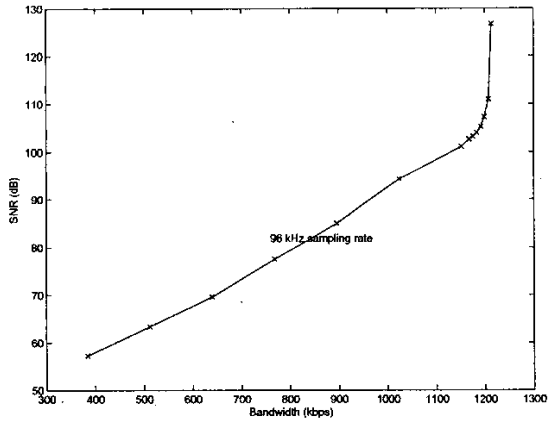

Fig. 3. An example of the SNR results obtained using the proposed scheme (violin)

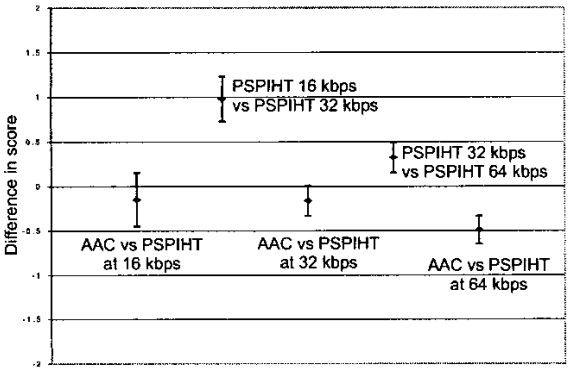

Fig. 4. Listening test results

as well as the difference between the mean scores and the value 3. Fig. 4 shows the results with their 95% confidence intervals. In the results shown in Fig. 4 a score that is below 0 indicates that the first coder mentioned in the label is better than the second coder according to the scale used in the test. For example, the first data point plotted is below 0 indicating that, on average, the AAC coder is better than the PSPIHT coder (both at 16 kbps) within the shown 95% confidence interval.

The subjective results shown in Fig. 4 indicate that the AAC coder and the proposed MLT-PSPIHT coder perform quite similarly at 16 and 32 kbps (with 95% confidence). The AAC outperforms the MLT-PSPIHT scheme at 64 kbps by a margin of 0.49 indicating a small overall difference in quality. The perceptual scalability of the MLT-PSPIHT scheme is clearly shown by the presented results with a clear difference between the 16 kbps coder and the 32 kbps coder, as well as a clear (although reduced) difference between the 64 kbps coder and the 32 kbps coder.

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. The objective results presented show that the scheme scales smoothly and objectively (in terms of SNR) from lossy to lossless compression. The lossless compression obtained is competitive with the state of the art at a bandwidth of 44.1 kHz, although, like all lossless compression schemes, the obtained compression depends on the signal content. The subjective listening tests conducted show the perceptual scalability of the PSPIHT scheme as well as its comparable performance with the MPEG-4 AAC coder. The main advantage of the PSPIHT scheme is that given an encoded bit stream the synthesized audio at any lower bit rate (as well as pre-determined bandwidth) is obtainable. For example, one can easily obtain the 16 kbps or 32 kbps bitstream from the 64 kbps bit stream, which is a very useful property for variable rate transmission.

6. ACKNOWLEDGMENTS

Mohammed Raad is a recipient of an Australian Postgraduate Award of Industry (APAI) grant. This work is supported by the Motorola Australian Research Centre. The authors wish to thank Ms. Melanie Jackson for her help in conducting the listening tests and Mr. Guillaume Potard for his assistance in obtaining 96 kHz sampled audio.

7. 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.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.

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

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[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.

[13] "Mpeg web site at http://www.tnt.uni-hannover.de/project/mpeg/audio/," .