2003

Multi-rate and multi-resolution scalable to lossless audio compression using PSPIHT

M. Raad  
*University of Wollongong, mraad@uow.edu.au*

I. Burnett  
*University of Wollongong, ianb@uow.edu.au*

Alfred Mertins  
*University of Oldenburg, Germany, mertins@uow.edu.au*

Publication Details  
Multi-rate and multi-resolution scalable to lossless audio compression using PSPIHT

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.

Keywords
audio coding, data compression, quantisation (signal)

Disciplines
Physical Sciences and Mathematics

Publication Details

This conference paper is available at Research Online: http://ro.uow.edu.au/infopapers/159
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 presented 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 \( x_0 \). 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 \( x_2 \) 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 \( bsl \) which is further divided into \( bsl1a \) and \( bsl1b \) by PSPIHT. This second stage division aims to separate perceptually significant coefficients from perceptually insignificant coefficients such that \( bsl1a \) contains the perceptually significant coefficients and is transmitted before \( bsl1b \). \( bsl1 \) 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 \( bsl2 \),
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 while 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 \( \nu_{pe} \) be a binary vector with perceptual significance information for the sub-band coefficients. That is, if \( \nu_{pe}(n) = 1 \) 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 \( bstla \) and the perceptually insignificant component as \( bstlb \). Fixed limits can be set for the size of \( bstla \) and \( bstlb \). The complete algorithm is listed below.

Algorithm PSPIHT:
1) Initialization: output \( n = \lfloor \log_2(\max(c_n)) \rfloor \);
   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 \( bstla \) is not full
      2.1.1) output \( S_a(i) \) to \( bstla \)
   Else
      2.1.1) output \( S_a(i) \) to \( bstlb \)
   2.1.2) if \( S_a(i) = 1 \) and \( bstla \) is not full then
         output \( \nu_{pe}(i) \) to \( bstlb \)

In the sorting pass, the energy significance test is maintained as the first test. Sorting bits are sent to \( bstla \) until an energy significant coefficient is encountered. This coefficient is then tested for perceptual significance by checking the corresponding entry in \( \nu_{pe} \), if the coefficient is found to be significant (and \( bstla \) is not full) then the sign bit and further refinement bits are sent to \( bstla \), otherwise these bits are sent to \( bstlb \). 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 incurred in the bitstream formatting as a pointer must also be transmitted indicating the length of \( bstla \). This is necessary for the decoder to be able to divide the total bitstream correctly and to allow \( bstla \) 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 \( bstla \), hence it would be rare to encounter a situation where all of \( \nu_{pe} \) is transmitted.

![Fig. 1. The overall structure of the proposed coder](image1)

![Fig. 2. The scalable-to-lossless scheme based on SPIHT and PSPIHT](image2)
2.2.2) if the entry is of type B then
If bstla is not full
  • output $S_n(D(i))$ to bstla
Else
  • output $S_n(D(i))$ to bstlb
  • if $S_n(k) = 1$ then
    • for each $k \in O(i)$ do:
      • if bstla is not full
        • output $S_n(k)$ to bstla
      Else
        • output $S_n(k)$ to bstlb
      • add $k$ to the LIS
    Else
      • output $S_n(k)$ to bstlb
  • if $L(i) \neq \emptyset$ then move (i) to the end of the LIS
  • if $L(i) = \emptyset$ 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;

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 bstla if it is not full,
otherwise output the nth most significant bit of $[c_i]$ to bstlb 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 bstlb

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

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 source data contains and the decoder would have to store.

<table>
<thead>
<tr>
<th>Content</th>
<th>Objective Description</th>
<th>Subjective Opinion</th>
</tr>
</thead>
<tbody>
<tr>
<td>violin</td>
<td>16, 48</td>
<td>A much better than B</td>
</tr>
<tr>
<td>vocal</td>
<td>20, 48</td>
<td>A much better than B</td>
</tr>
<tr>
<td></td>
<td>24, 96</td>
<td>A much better than B</td>
</tr>
</tbody>
</table>

Table 1. Objective results for 96 kHz sampled files

<table>
<thead>
<tr>
<th>Bits, Sampling (kHz)</th>
<th>Violin</th>
<th>Vocal</th>
</tr>
</thead>
<tbody>
<tr>
<td>16, 44.1</td>
<td>2.66, 6.02</td>
<td>3.52, 4.55</td>
</tr>
<tr>
<td>20, 48</td>
<td>1.59, 12.57</td>
<td>1.87, 10.72</td>
</tr>
<tr>
<td>24, 96</td>
<td>1.41, 17.07</td>
<td>1.54, 15.24</td>
</tr>
</tbody>
</table>

The mean scores for all the comparisons have been obtained.

Table 2. The subjective comparison scale used

<table>
<thead>
<tr>
<th>Score</th>
<th>Subjective Opinion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A much better than B</td>
</tr>
<tr>
<td>2</td>
<td>A better than B</td>
</tr>
<tr>
<td>3</td>
<td>A and B are the same</td>
</tr>
<tr>
<td>4</td>
<td>A worse than B</td>
</tr>
<tr>
<td>5</td>
<td>A much worse than B</td>
</tr>
</tbody>
</table>
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 bitstream, 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 Poutad for his assistance in obtaining 96 kHz sampled audio.

7. REFERENCES