Audio Engineering Society

Convention Paper

Presented at the 111th Convention
2001 September 21–24 New York, NY, USA

Efficient Audio Coding with Fine-Grain Scalability

Chris Dunn

Scala Technology
London, UK

web: http://www.scalatech.co.uk
email: chris.dunn@scalatech.co.uk

ABSTRACT

A comparison of audio coder quantisation schemes that offer fine-grain bitrate scalability is made with reference to fixed-rate quantisation. Coding efficiency is assessed in terms of the number of bits allocated to significant transform coefficients, and the average number of significant coefficients coded. A new method of arranging the transform hierarchy for SPIHT zero tree algorithms is shown to result in significantly improved performance relative to previously reported SPIHT implementations. Results for a new quantisation algorithm are presented which suggest low-complexity fine-grain scalable coding is possible with no coding efficiency penalty relative to fixed-rate coding.

1 INTRODUCTION

Audio coding algorithms with bitrate scalability allow an encoder to transmit data at a high bitrate and decoders to successfully decode a lower-rate bitstream contained within the high-rate code. For example, an encoder might transmit at 128 kbit/s while a decoder would decode at 32, 64, 96 or 128 kbit/s according to channel bandwidth, decoder complexity and quality requirements. Scalability is becoming an important aspect of low bitrate audio coding, particularly for multimedia applications where a range of coding bitrates may be required, or where bitrate fluctuates. Fine-grain scalability, where useful increases in coding quality can be achieved with small increments in bitrate, is particularly desirable.

The growth of the internet has created a large demand for high-quality streamed audio content. Audio coding with fine-grain bitrate scalability allows uninterrupted service in the presence of channel congestion, achieves real-time streaming with low buffer delay, and yields the most efficient use of available channel bandwidth. Scalability is also useful in archiving, where a program item may be coded at the highest bitrate required and stored as a single file, rather than storing many coded versions across the range of required bitrates. As well as the saving in overall storage requirement, bitrate scalability avoids the cumulative reduction in coding quality that can occur due to recoding. Scalable audio coding has further potential applications in mobile multimedia communication, digital audio broadcasting, and remote personal media storage.

While fine-grain bitrate scalability can be extremely useful, it is important that it is achieved without significant coding efficiency penalty relative to fixed bitrate systems, and with low computational complexity.

An attractive approach to achieving fine-grain scalability is ordered bitplane coding, where bits are transmitted in order of significance to produce a fully embedded bitstream. In this paper we compare the coding efficiency of 3 bitplane coding algorithms suitable for audio coding with a more conventional fixed-rate quantiser. All of the algorithms considered offer low complexity coding, and do not require Huffman or arithmetic coding. Coding efficiency is compared by measuring the proportion of total bitrate allocated to directly coding non-zero transform coefficients, a metric which is closely related to Johnston's description of perceptual entropy. Also considered is the average number of non-zero coefficients coded in each frame.

The fixed-rate reference algorithm uses gain-adaptive quantisation to effectively entropy code the transform output without the use of Huffman tables. The bitplane coding algorithms considered include EZK and SPIHT, zero tree implementations adapted for use with 1-dimensional uniform-decomposition transform outputs commonly used in audio coding. While the EZK algorithm exhibits significantly lower coding efficiency compared to the fixed bitrate reference, improved efficiency is observed for SPIHT with a novel tree hierarchy. Finally an enhanced scalable quantisation algorithm is described which offers low-complexity fine-grain bitrate scalability with coding efficiency very close to that of the fixed coder.

2 BITPLANE CODING

Audio compression typically includes some form of transform coding where the time-domain audio signal is transformed to the frequency domain before quantisation and frame packing (Fig. 1). A psychoacoustic model determines a target noise shaping profile which is used to allocate bits to the transform coefficients such that quantisation errors are least audible to the human ear. In a conventional fixed bitrate encoder the bit allocation is typically achieved with a recursive algorithm that attempts to meet the noise-shaping requirement within the bitrate constraint [1]. The final bit allocation information is used to quantise transform coefficients and also included as side information within the bitstream for use at the decoder.

A common approach to achieving scalability is the ‘error-feedforward’ arrangement, where a core coder produces the lowest embedded bit rate and subsequent layers progressively reduce the error due to the core (Fig. 2). However, a significant amount of side information is associated with each layer which can reduce coding efficiency, and the number of possible decoding rates is limited to the number of layers. The error feedforward approach was used in a 2-layer coder described by Herre et al [2], where a TwinVQ core layer optimised for low bitrates was embedded within a higher-rate AAC layer. Extrapolating the results presented in [2] suggests the bitrate penalty for achieving scalability with this scheme, relative to a fixed-rate single-layer coder, is approximately 15 % of the overall bitrate.

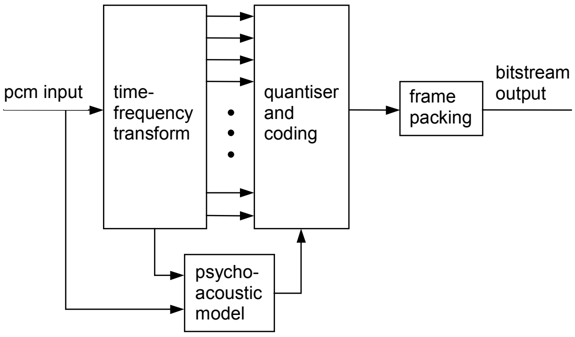

Fig. 1. Perceptual transform encoder for audio compression.

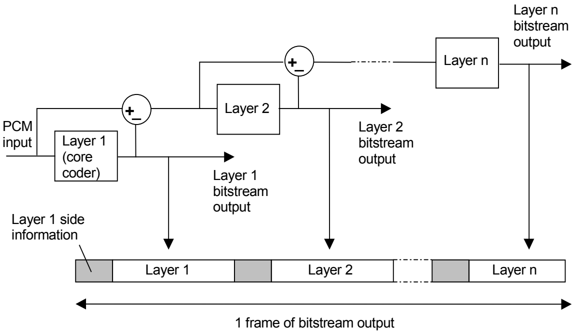

Fig. 2. Scalable encoder using error feedforward.

An alternative approach to achieving scalability is ordered bitplane coding of transform coefficients, where in each frame coefficient bitplanes are coded in order of significance, beginning with the most significant bits (MSB's) and progressing to the LSBs. This results in fully-embedded coding where the bitstream at a certain rate contains all lower-rate codes, and exhibits fine-grain scalability in contrast to the coarse granularity offered by error-feedforward systems. Bitplane coding can also yield a significant increase in encoding speed since quantisation typically requires a single scan through the transform coefficients for each frame, as opposed to the recursive bit allocation search executed in fixed rate coding.

Ordered bitplane coding is used in the Bit-Sliced Arithmetic Coding (BSAC) system described by Park et al. [3]. While the BSAC coder is reported to perform well in terms of coding efficiency, it requires the use of arithmetic coding which can increase computational complexity, and bitrate granularity is limited to 1 kbit/s enhancement steps.

Zero tree quantisation, first described for use in image coding by Shapiro [4] and later refined by Said and Pearlman with the SPIHT algorithm [5], combines ordered bitplane coding with compact significance map representation. Significance maps [4] are binary maps indicating the locations of newly significant coefficients within the current bitplane - ie the positions of coefficients that have their MSB located within the current bitplane. Coefficients are scanned sequentially following the order of a zerotree hierarchy, where every coefficient at a given frequency (the *parent*) can be related to a set of coefficients at higher frequencies (the *children*), and no child node is scanned before its parent. In two-dimensional image coding each parent coefficient typically has four children, and the zerotree hierarchy can efficiently code significance maps because if a low-frequency coefficient is found to be insignificant, then higher-frequency coefficients with the same spatial location are also likely to be insignificant.

Each time a significant coefficient is found its sign is output and the position of the coefficient stored for later use. When all coefficients in the hierarchy have been scanned a refinement loop is executed which outputs the next significant bit of each coefficient already found to be significant in the current and higher bitplanes. The algorithm then progresses to the next bitplane and generates a new significance map for the remaining (previously insignificant) coefficients. This process is repeated, switching between significance map generation and refinement loop for each bitplane until either a target bitrate or coding accuracy is achieved.

All of the scalable algorithms considered below - EZK, SPIHT, and ESC - combine significance map coding with a refinement loop. The major differences between the algorithms lie in the way the significance map is generated. The SPIHT algorithm described by Said and Pearlman [5] generates the significance map by storing significance information in three lists, termed list of insignificant sets (LIS), list of significant coefficients (LSC) and list of insignificant coefficients (LIC). Each list entry identifies a position in the hierarchy, corresponding to coefficients in the case of LSC and LIC members, and descendent coefficient sets in the case of LIS members.

3 CODING EFFICIENCY METRICS

An important issue in the design of scalable algorithms is how to assess coding efficiency. While coding efficiency for fixed (non-scalable) algorithms is commonly measured in terms of the mask-to-noise ratio achieved at the decoder output across a range of bitrates, it is difficult to use this measure to directly compare fixed-bitrate and bitplane encoders due to the different quantisation functions used with the two approaches (for bitplane coding the width of the zero amplitude bin is typically twice that of other bins [6]). An alternative measure of coding efficiency is to record the proportion of total bitrate allocated to directly coding coefficients that have been quantised to non-zero values. The following discussion is equally valid for scalable or fixed-rate quantisation algorithms.

Consider a coder with a frame length of *M* samples. Following time-to-frequency transformation with critical sampling, the quantiser is presented with a set of *M* coefficients to quantise and code for each frame. Now suppose that *nsig* coefficients are quantised to non-zero integer values *x*(*i*), that is *nsig* out of *M* coefficients have been found to be significant. In the absence of entropy coding the minimum number of bits *NZ* that must be used to describe *x*(*i*) is given by:

$$NZ = \sum_{i=0}^{nsig-1} \left\lceil \log_2 \left(2\left|x(i)\right| + 1\right) \right\rceil \qquad (1)$$

This expression is similar to Johnston's description of perceptual entropy [7, Sec. 2.4.3]. Note that *NZ* consists of magnitude and sign information for non-zero coefficients, and does not include information regarding the *positions* of *x*(*i*) within *M*. Position information is provided by a significance map. Let the total number of bits used to code each frame = *B*, and the number of bits used to code the significance map = *SM:*

$$B = SM + NZ \qquad (2)$$

Efficient coding requires that the significance map be coded with as few bits as possible, since it does not convey information about significant coefficient values. A measure of coding efficiency η is therefor obtained by calculating the proportion of total bitrate allocated to *NZ*:

$$\eta = \frac{1}{B} \sum_{i=0}^{nsig-1} \left\lceil \log_2 \left(2\left|x(i)\right| + 1\right) \right\rceil \qquad (3)$$

η is a measure of how efficiently quantised coefficient data is 'supported' within the generated bitstream. Shapiro has shown that at low bitrates a large proportion of bitrate must be allocated to the significance map, even with the most efficient coding scheme possible [4, Sec. 3], hence we would expect η to be significantly lower than unity. In practice η will tend to vary from frame to frame, hence averaging over the duration of a test piece provides a useful measure of algorithm coding efficiency.

Note that in deriving η we have assumed no entropy coding of significant coefficient values. However, in a bitplane coder for bits below the MSB the occurrence of 1's and 0's are equally likely to a first-order approximation, hence this assumption does not invalidate the coding efficiency model.

A second useful measure of coding efficiency is obtained by calculating the average number of significant coefficients *nsig* identified in each frame. The maximum possible value for *nsig* is clearly equal to *M*, but typically less than 1/2 of all available coefficients are coded as non-zero values, even at higher bitrates. Shapiro has shown that at lower bitrates, even with the most efficient coding possible, the large majority of coefficients must be coded as insignificant [4].

4 FIXED RATE REFERENCE CODER

We can use the evaluation framework established above to measure the coding efficiency of a fixed bitrate coder, which can then be used as a reference with which to compare the performance of scalable quantisation algorithms. The fixed rate coder follows the generic coder structure outlined in Fig. 1. For evaluation purposes a fixed-length modified discrete cosine transform (MDCT) is used with a frame length *M* of 1024 samples. With a 50% overlapping window length of 2048 samples, this transform arrangement is similar to the long block mode of MPEG-2 AAC [8]. The transform output is partitioned into 32 bands where the band boundaries follow a compressed critical-band law. The coefficients in each band are normalised using a band scalefactor, which is quantised in steps of 3.0 dB and coded as bitstream side information using 5 bits/band.

A custom psychoacoustic model and recursive bit allocation algorithm is used to calculate the number of bits allocated to each band, which is coded as side information using 4 bits/band. Gain-adaptive quantisation is used to entropy code the transform output without the use of Huffman tables or arithmetic coding [9]. This quantisation approach achieves a significant reduction in computational complexity compared to using Huffman tables. A gain index is established for each quantiser band such that normalised coefficient values exceeding unity are coded with an additional escape value. A search algorithm determines the optimal gain index that allows band coefficients to be coded using the fewest number of bits. In the reference coder gain indices are coded as side information using 3 bits/band. The remainder of the bitstream comprises quantised coefficient data.

The fixed-rate bitstream contains a significant amount of side information, including scalefactors, bit allocation data and quantiser gain indices. These bitstream fields do not directly contribute towards significant coefficient values, and hence should be considered as a component of an equivalent significance map - that is, they help determine the location of significant coefficients. The remainder of the significance map is contained in the location of band coefficients quantised to zero.

Fig. 3 shows the measured coding efficiency for the fixed quantiser algorithm. At each bitrate efficiency is measured by averaging η across 3 challenging single-channel test pieces (harpsichord, pitchpipe, and female voice with background music), sampled at 48 kHz and coded with full 24 kHz bandwidth. It is clear that even at higher bitrates coding efficiency is less than 50 %. Similarly the average number of significant coefficients coded, shown at each bitrate in Fig. 4 as an average for the 3 test pieces, is generally less than half of the total number of coefficients.

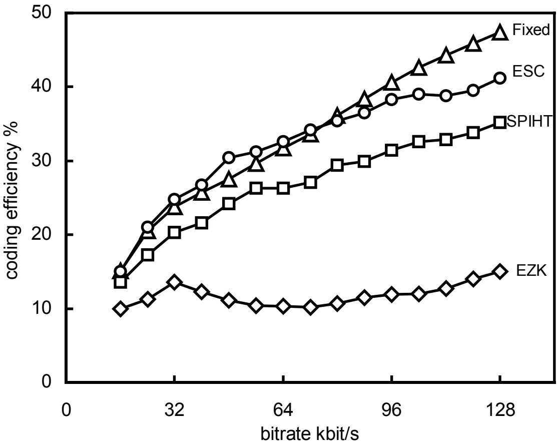

Fig. 3. Coding efficiency η as a function of bitrate.

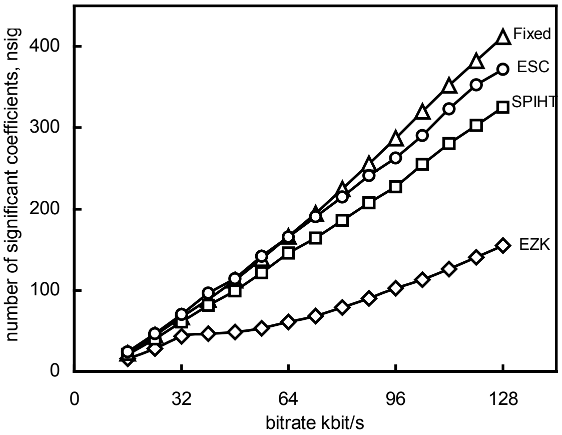

Fig. 4. Average number of significant coefficients *nsig* coded in each frame as a function of bitrate.

5 SCALABLE CODERS

The three scalable coders considered below all use ordered bitplane coding to achieve fine-grain bitrate scalability. They use the same 2048-pt MDCT, psychoacoustic model and 32-band coefficient partitions used with the fixed-rate reference coder. Bitplane coding effectively quantises each transform coefficient with the same step size, hence in order to achieve a perceptually appropriate shaping of quantisation noise the transform data is weighted in each band prior to quantisation [10]. Band weights are determined by the psychoacoustic model and logarithmically quantised with a step size of 3 dB before coding as side information using 4 bits/band. Band weight data is considered as a component of the significance map for the purposes of calculating coding efficiency.

Note that while all of the bitplane quantisation schemes considered below achieve fine-grain bitrate scalability with fixed (full) bandwidth, it is also possible to scale bandwidth with bitrate. Fixed bandwidth scalability can be optimal for limited variations in channel capacity or where the variation is short-term. However, when channel capacity is restricted to low bitrates over a significant period of time then reduced-bandwidth coding can result in improved subjective quality, essentially because the available bitrate is shared between fewer significant coefficients. Bandwidth scalability would require modifying the algorithms described below to code coefficients grouped into separate layers, each with an associated coding bandwidth and bitrate range within which fine-grain scalability is preserved.

5.1 EZK

The EZK algorithm described in [11] is a refinement of SPIHT for use with uniform transform decompositions. A marker is used to record the progress of a scan through coefficients from low to high frequency, and if new significant coefficients are found within the current bitplane then the position of the next significant coefficient relative to the marker is recorded by run-length coding. Insignificant coefficients between the marker and the next significant coefficient are moved to the LIC, the marker updated and remaining coefficients scanned for further significant descendants. The process is repeated until all coefficients have been scanned.

While EZK achieves fine-grain bitrate scalability, Figs. 3 and 4 indicate coding efficiency is well below that of the fixed-rate coder. For example, Fig. 4 indicates that on average EZK identifies 100 significant coefficients in each frame at 96 kbit/s, whereas fixed quantisation achieves this performance at 45 kbit/s.

5.2 SPIHT

EZK was previously reported in [11] to achieve improved coding efficiency compared to a SPIHT implementation with a tree hierarchy where each parent has only a single offspring. However, such a SPIHT arrangement is unlikely to be effective because it cannot efficiently predict the location of insignificant coefficients. More compact significance map coding is achieved when each parent in the hierarchy has many children. The problem is how to map the one-to-many hierarchy to a 1-dimensional transform array suitable for audio coding.

Fig. 5 shows one possible tree hierarchy where each parent coefficient has 4 child coefficients that are clustered together in frequency. Coding efficiency results shown in Figs. 3 and 4 indicate this SPIHT arrangement achieves a significant performance improvement over EZK. Nevertheless coding efficiency remains somewhat below that of the fixed-rate reference coder.

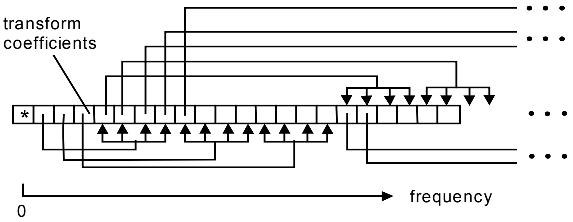

Fig. 5. SPIHT hierarchy for use with uniform decomposition transforms, where each parent coefficient has 4 offspring.

5.3 Enhanced Scalable Coder

Although EZK and SPIHT zero tree quantisation suffer significant coding efficiency penalties relative to fixed bitrate coding, it is possible to construct algorithms that offer fine-grain bitrate scalability with competitive efficiency. The Enhanced Scalable Coder (ESC) achieves this goal without the use of arithmetic or Huffman coding, and with low algorithmic complexity.

Fig. 3 shows ESC coding efficiency to offer a significant improvement over SPIHT, approximately matching the fixed rate coder performance across a broad range of bitrates. This trend is repeated with the data for average number of significant coefficients coded shown in Fig. 4. While ESC has a small coding efficiency advantage at low bitrates, the fixed coder is more efficient for bitrates greater than 64 kbit/s. The scalable coder is subjectively transparent at approximately 96 kbit/s (see below), at which bitrate it suffers a coding efficiency penalty relative to the fixed-rate reference of about 7 % of overall bitrate.

If bitrate granularity is defined as the average increase in bitrate required to increase the number of significant coefficients coded by one, then ESC granularity is approximately 0.35 kbit/s .

6 EVALUATION CODECS

Prototype coders were constructed in order to evaluate the subjective performance of the fixed-rate reference and ESC quantisation algorithms. These designs follow the structures outlined in Sections 4 and 5, with the exception that block switching is used to adapt the transform length to signal conditions for improved transient performance. For stationary signal frames a single 2048-point transform window is used, while under transient conditions eight overlapping 256-point windows are used. This transform structure is essentially that used by MPEG2-AAC at 48 kHz sampling rate [8].

Informal but carefully controlled listening tests using demanding single-channel 48 kHz sampled test pieces indicate the average transparency bitrate for ESC to be approximately 108 kbit/s (Table 1). This performance is very close to that of the fixed-rate reference coder, and slightly below that achieved by MP3 [12]. For less demanding program material transparency is achieved at significantly lower bitrates.

Table 1: Comparison of transparency bitrates for scalable and fixed evaluation codecs.

Signal	**Transparency bitrate kbit/s**		
	Coding algorithm		
	Enhanced Scalable Coder (ESC)	Fixed-bitrate Reference Coder	FhG MP3
harpsichord	96	96	96
pitchpipe	96	96	96
voice	112	96	96
castanets	128	128	96
average	108	104	96

The scalable quantisation algorithm exhibits a significant reduction in computational complexity relative to the fixed rate reference coder. Most of the processing time required by the ESC demonstration encoder is due to the psychoacoustic model. The computational burden of the scalable quantisation algorithm in the encoder is very similar to that of the dequantisation algorithm in the decoder.

Copies of the scalable- and fixed-bitrate evaluation codecs are available to download from http://www.scalatech.co.uk

7 CONCLUSIONS

In this paper we have developed a framework for evaluating the coding efficiency of audio codecs based on the proportion of total bitrate allocated to directly coding significant transform coefficients. This approach was used to measure the performance of several quantisation algorithms that achieve fine grain bitrate scalability through ordered bitplane coding.

The previously reported EZK algorithm was found to be significantly less efficient than a fixed-rate reference coder. This performance was improved on by a SPIHT zero tree algorithm with a novel hierarchical tree arrangement of transform coefficients.

Finally, the coding efficiency of an enhanced scalable coder was found to broadly match that of the fixed-rate reference, whilst offering fine-grain bitrate scalability and low computational complexity.

8 ACKNOWLEDGEMENT

This work has been possible due to development funding by the UK Government Department of Trade and Industry under the SMART scheme.

9 REFERENCES

[1] J. D. Johnston, “Transform Coding of Audio Signals Using Perceptual Noise Criteria,” *IEEE J. Select Areas in Communications*, vol. 6, pp. 314 – 323 (1988 Feb.).

[2] J. Herre et al., “The Integrated Filterbank Based Scalable MPEG-4 Audio Coder,” presented at the 105th Convention of the Audio Engineering Society, San Francisco, 1998 (preprint 4810).

[3] S. H. Park et al., “Multi-Layer Bit-Sliced Bit Rate Scalable Audio Coding,” presented at the 103rd Convention of the Audio Engineering Society, New York, Sep. 1997 (preprint 4520).

[4] J. M. Shapiro, “Embedded Image Coding Using Zerotrees of Wavelet Coefficients,” *IEEE Trans. Sig. Proc*., vol. 41, pp. 3445 – 3462 (1993 Dec.).

[5] A. Said and W. A. Pearlman, “A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees,” *IEEE Trans. Circuits and Sys. Video Tech*., vol. 6, pp. 243-250 (1996 June).

[6] S. Mallet, “Analysis of Low Bit Rate Image Transform Coding*,” IEEE Trans. Sig. Proc*., vol. 46, pp. 1027 – 1042 (1998 Apr.).

[7] J. D. Johnston, "Estimation of Perceptual Entropy Using Noise Masking Criteria," Proc. ICASSP 1988, pp. 2524 - 2527.

[8] M. Bosi et al., “ISO/IEC MPEG-2 Advanced Audio Coding,” *J. Audio Eng. Soc*., vol. 45, pp. 789 – 812 (1997 Oct.).

[9] L. D. Fielder and G. A. Davidson, “Audio Coding Tools for Digital Television Distribution,” presented at the 108th Convention of the Audio Engineering Society, Paris, Feb. 2000 (preprint 5104).

[10] H. Malvar, “Enhancing the Performance of Subband Audio Coders for Speech Signals,” presented at 1998 IEEE Int. Symp. Circuits and Sys., Monterey CA, vol. 5, pp. 98 – 101 (1998 June).

[11] B. Leslie, C. Dunn and M. Sandler, “Developments with a Zero Tree Audio Codec,” Proc. AES 17th International Conf. ‘High Quality Audio Coding’, Florence, pp. 251-257 (1999 Sep.).

[12] http://www.iis.fhg.de/amm/download/