A BITSTREAM SCALABLE AUDIO CODER USING A HYBRID WLPC–WAVELET REPRESENTATION

Daryl Ning and Mohamed Deriche

School of Electrical and Electronic Systems Engineering
Queensland University of Technology, Australia

ABSTRACT

In this paper, we present a novel bitstream scalable audio coder. In the proposed coder, the full bandwidth of input audio is first split into two. A hybrid WLPC–wavelet representation is used to encode the low frequency components (< 11 kHz). In this method, the excitation to the WLPC synthesis filter is decomposed into subbands using a wavelet filterbank, and perceptually encoded. Two stage quantisation of the wavelet coefficients is used to provide scalability. The high frequency components of the input are assumed to be noisy, and efficiently encoded using an LPC noise model. The output bitstream is capable of being decoded at rates between 16 kbps and 80 kbps. As the bitrate increases, so too does the signal quality. At 80 kbps, the quality is near transparent. At the intermediate rates, the coder gives comparable performance to the MPEG layer III coder, when the MPEG coder operates at similar, but fixed, bitrates.

1. INTRODUCTION

Digital audio is increasingly becoming more and more a part of our daily lives. Unfortunately, the excessive bitrate associated with the raw digital signal makes it an extremely expensive representation. Applications such as digital audio broadcasting, high definition television, and internet audio, require high quality audio at low bitrates. The field of audio coding addresses this important issue of reducing the bitrate of digital audio, while maintaining a high perceptual quality. The most popular of these audio coders would have to be the MPEG family [1] of high quality coders. In addition to high quality coding, however, it is also important for audio coders to be flexible in their application. With the increasing popularity of internet audio, it is advantageous for audio coders to address issues related to real–time audio delivery. One such issue that has been the target of recent research is bitstream scalability [2–4].

Bitstream scalability refers to a coder capable of producing an embedded or layered bitstream. Multiple grades of quality can be achieved by decoding portions or subsets of the single bitstream. The absolute minimum portion required for decoding is called the base or core layer. Any additional portions of the bitstream that add to the quality of the decoded audio are called enhancement layers. This makes bitstream scalability extremely useful in real–time streaming of audio, since one copy of the encoded audio is all that is required to service many users with varying bandwidth connections. Each user can extract as much information from the bitstream as they like, depending on their connection speed or desired quality. This is in contrast to storing multiple versions (at different bitrates) of the same signal on the server, which has the obvious disadvantage of high storage requirements. The major disadvantage with bitstream scalable coders, however, is the performance hit. A bitstream scalable coder cannot be optimised for each intermediate bitrate, since the bitstream must conform to a layered structure.

In section 2 we give a brief overview of the proposed bitstream scalable coder. Within this section we cover the major blocks of the encoder. This includes the noise model, the warped linear predictive coding (WLPC) analysis and associated quantisation scheme, the discrete wavelet transformation (DWT) of the excitation, and the quantisation scheme of the DWT coefficients required for embedded encoding. Section 3 then describes the layered structure of the bitstream. This is followed by a discussion of the experiments and results in section 4.

2. DESCRIPTION OF CODER

The block diagram of the proposed coder is given in figure 1. Each frame is 512 samples with an overlap of 32 samples. The window is rectangular with a raised sine window roll–off in the overlap regions. The encoder begins by splitting the input frame into its lowpass and highpass components using a 2–band quadrature mirror filter. The output of each filter band is then downsampled by a factor of 2 to maintain critical sampling. The highpass output is assumed to be noise–like, and is therefore encoded using an LPC noise modelling technique. The lowpass component is encoded using a modified version of the WLPC-wavelet coding scheme presented in [5]. Each major section of the encoder will now be discussed in more detail.

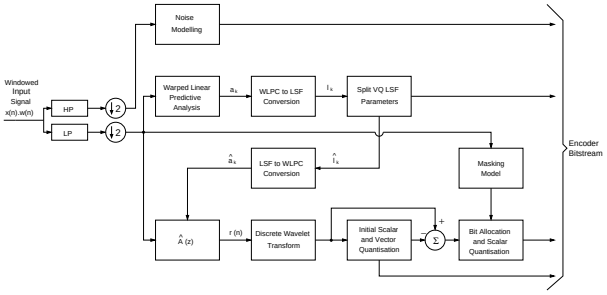
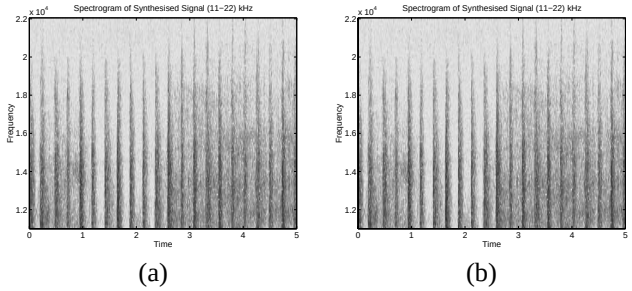


Fig. 1. Proposed bitstream scalable audio encoder.



(a) (b)

Fig. 2. Spectrogram of (a) original signal, and (b) synthesised signal.

2.1. Noise Modelling

In this coder, we basically assume that signal components above 11 kHz can be considered to be noise–like. While this is not true for all signals, it is valid for the majority of real–world signals. In fact, similar assumptions have been made in previous coders [2, 6]. The advantage of using a noise model is that the high frequency components can be encoded very efficiently with excellent perceptual quality.

The model used in this coder is based on the human perception of broadband noise. We use the fact that for noisy signals, the auditory system is primarily sensitive to the short–time spectral envelope, and not the waveform. In this coder, we model the spectral envelope of the noise using an LPC filter. The input signal (the downsampled output of the highpass filter) is first analysed using a 16th order linear predictor. This approximates the spectrum of the signal by an all–pole filter, $A(z)$. The gain of this filter is calculated and quantised. The 16 LPC parameters are then converted to their line spectral frequency (LSF) representations and quantised using split vector quantisation (VQ). To synthesise the noise, we generate a random signal, multiply it by the gain, and filter it through the synthesis filter, $A(z)$.

We illustrate the effectiveness of this scheme with an example. For this purpose, we run a pop music signal through a high pass filter and encode the output using the aforementioned process. The spectrograms (for 5 seconds) of the original high pass signal and the synthesised signal are shown in figures 2(a) and (b) respectively. From these figures, we can see that the noise modelling process faithfully approximates the short–time spectrum of the original signal. Informal listening tests also verified that this model delivers a perceptually similar sounding signal to the original.

2.2. WLPC Analysis and Quantisation

The downsampled output of the lowpass filter is analysed using WLP. WLP was chosen due to its inherent noise shaping properties, and its ability to approximate the frequency resolution of the human ear [5]. A 16th order predictor is used to estimate the 11 kHz bandwidth spectrum on a warped frequency scale. The 16 WLPC parameters are then converted to their LSF representations and quantised using multistage split VQ.

2.3. DWT of Excitation Signal

The excitation signal is obtained by filtering the downsampled lowpass output through the quantised WLPC synthesis filter. The wavelet filterbank structure in [5] is then used to decompose the excitation into 14 subbands for perceptual encoding. This structure was chosen to approximate the critical bands of human hearing, and hence, assist in the noise shaping process during quantisation.

2.4. Quantisation of Wavelet Coefficients

The quantisation of the DWT coefficients consists of two stages. The initial coarse quantisation of the DWT coefficients provides low quality encodings of the input signal at low bitrates. These form the *base* layers of the encoder. The first stage coarse encodings are implemented using fixed rate scalar and VQ. The first four subbands (0–688 Hz) are scalar quantised, while the remaining subbands are vector quantised. After this initial quantisation procedure, these first stage coefficients are subtracted from the original coefficients to give the error. The subbands of the error are then allocated bits and quantised, using the same algorithm and techniques described in [5].

2.4.1. First Stage Scalar Quantisation

The first 4 subbands contain only 4 coefficients each. For these low frequency subbands, listening tests showed that at least 3 bits per coefficient were required to provide reasonable quality. Vector quantisers comprising of 9 and 10 bit (per subband) codebooks were also tested, however, the scalar quantiser delivered better overall quality. Consequently, it was decided to use a 3 bit scalar quantiser for each of the coefficients in the first 4 subbands.

Before quantisation, it is necessary to first normalise each of the coefficients. One scalefactor was calculated from all 4 subbands for this purpose. After scalar quantising the coefficients, the scalefactor is recalculated to minimise the squared error between the original and quantised coefficients. If we denote the original set of DWT coefficients as $v(n)$, and the quantised normalised set as $\tilde{v}(n)$, we can write the squared error as, $E^2 = \sum_n \left[v(n) - G\tilde{v}(n)\right]^2$, where G is the scalefactor. To find the optimal scalefactor given the set of quantised coefficients, $\tilde{v}(n)$, we simply differentiate E^2 with respect to G, and equate to zero. Doing this, we obtain:

$$G = \frac{\sum_n \tilde{v}(n)v(n)}{\sum_n \tilde{v}(n)\tilde{v}(n)}. \quad (1)$$

We therefore recalculate the value of the scalefactor using equation 1.

2.4.2. First Stage Vector Quantisation

The remaining subbands, i.e., subbands 5 to 14, are vector quantised at their first stage. It is necessary to use VQ because the number of coefficients within each subband is so large. The quantisation of each subband requires the calculation of a scalefactor for normalisation, as well as the selection of a codevector from a codebook. To minimise the squared error between the original coefficients and the quantised coefficients, would require mutually optimising both the scalefactor and codevector. This, however, would be a computational burden. To simplify this process, we adopt a similar approach to the scalar quantisation of the first 4 subbands. For each subband, 5 through 14, an initial scalefactor is extracted to normalise each of the DWT coefficients. The closest codevector (in a Euclidean sense) is then selected from the appropriate codebook. The scalefactor is then recalculated using equation 1 and quantised. In other words, we minimise the squared error with respect to a fixed codevector.

Separate codebooks were trained for each of the subbands using the Linde–Buzo–Gray algorithm with a basic Euclidean distance measure. It is interesting to note that neither the first stage scalar or vector quantisation schemes implement any perceptual model for encoding. Both use a fixed bit allocation, regardless of the properties of the input signal. This is only possible due to the WLPC synthesis filter which shapes the spectrum of the noise (reconstruction error) to the spectrum of the input. This is illustrated in figure 3, where the spectrum of a frame of a pop signal is plotted against the spectrum of the reconstruction error after first stage quantisation of subbands 1–10. Notice that the spectrum of the error is approximately shaped under the input spectrum. In the absence of the WLPC synthesis filter, the spectrum of the noise would be relatively flat, resulting in greater audible distortions.

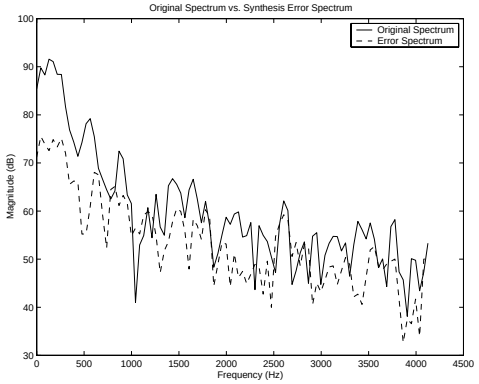


Fig. 3. Synthesis error after first stage scalar and vector quantisation of a frame of a pop signal.

Bitrate	Parameters
15.9 kbps	1st stage LSF's, 1st stage subbands 1–10
19.2 kbps	2nd stage LSF's
20.7 kbps	1st stage subband 11
22.2 kbps	1st stage subband 12
25.9 kbps	1st stage subband 13
25.9–44 kbps*	2nd stage subbands
48 kbps	1st stage subband 14
48–60 kbps*	2nd stage subbands
64 kbps	Highpass noise parameters
64–80 kbps*	Remaining 2nd stage subbands

Table 1. Structure of scalable bitstream. The * denotes that the bitrate can take small step increments of less than 0.8 kbps between the given limits.

2.4.3. Second Stage Scalar Quantisation

After the initial scalar and vector quantisation of the DWT coefficients, the error is calculated by subtracting this coarse approximation from the original set of coefficients. The error is then scalar quantised using perceptual criteria. Bits are allocated to each of the subbands using the same algorithm described in [5]. Each of the normalised coefficients are non–uniformly quantised with the number of bits assigned to their respective subbands.

3. BITSTREAM LAYERS

After quantising all the relevant information, the data is packed into a layered bitstream. Higher layers correspond to higher bitrates. Each increment in bitrate should provide a corresponding improvement in quality. The proposed coder produces a bitstream that provides scalability at a range of bitrates between 16 and 80 kbps. This structure is summarised in table 1

At bitrates up to 25.9 kbps, the bitstream is made up of the LSF parameters and the first stage quantisation of

subbands 1–13. Each subband layer basically increases the bandwidth of the reconstructed signal. After including subband 13, the encoder starts adding the second stage scalar quantised DWT coefficients to the bitstream. The coefficients are simply added in their temporal order starting from the first (lowest frequency) subband. These coefficients refine the quantisation of the DWT coefficients already present, however, they do not increase the bandwidth of the signal. Depending on the bit allocation, each coefficient adds no more than 0.8 kbps to the overall bitrate. At the same time, the quality of the reconstructed signal gradually improves. The inclusion of the coefficients can stop at any point to give the desired rate.

The second stage coefficients continue to be added to the bitstream until the bitrate reaches 44 kbps. At this point, the first stage VQ subband 14 is included which increases the bitrate to 48 kbps and the bandwidth to 11 kHz. The second stage coefficients then continue to be added (where they left off), until the bitrate reaches 60 kbps. Again, these coefficients gradually increase the bitrate with commensurate improvement in signal quality (but no increase in bandwidth). At 60 kbps, the highpass noise parameters are added to include the bandwidth up to 22 kHz. This increases the bitrate to 64 kbps. Finally, the remaining second stage coefficients are added to the bitstream until the overall bitrate reaches 80 kbps.

4. EXPERIMENTS AND RESULTS

Two informal listening tests were used to evaluate the performance of the proposed scalable audio coder. The first was to determine if the coder's quality improved as the bitrate increased. The second test compared the performance of the proposed scalable coder to the fixed rate MPEG layer III coder. For both tests, a number of popular music recordings were used.

The first test involved encoding one minute long pieces of audio. The decoder was initially configured to decode at 16 kbps, but this was increased every few seconds until it reached the final rate of 80 kbps. In all cases, the output audio started as a coarse bandlimited signal, and gracefully improved to a high quality, full bandwidth signal. At the maximum rate of 80 kbps, the coder gave near transparent quality. From this experiment, we concluded that the coder was successful at giving a commensurate increase in quality with bitrate.

The proposed bitstream scalable coder gave comparable results to the fixed rate MPEG layer III coder, except in some cases, where the MPEG coder gave slightly favourable results. The reason for this is that the MPEG layer III coder, or any fixed rate coder for that matter, need not worry about embedded bitstreams required for scalability. As a result, the quality of the signal can be highly optimised for each specific bitrate. While the MPEG layer III coder performs better at the equivalent bitrate, the disadvantage is its limited application to real–time audio delivery. Multiple encoded versions of the same signal would need to be stored on the server, for the same scalability offered by the proposed coder.

5. CONCLUSIONS

In this paper, we have presented a novel bitstream scalable audio coder. Informal listening tests show that signal quality improves as the bitrate increases from 16 kbps to 80 kbps. At 80 kbps the quality is near transparent. When compared to the MPEG layer III coder operating at a similar, but fixed, bitrate, the proposed coder delivers comparable performance. The proposed coder, however, has the advantage of being applicable to real–time audio delivery. Audio quality could be improved, especially at the lower bitrates, by investigating perceptual distance measures for the first stage VQ of the DWT coefficients.

6. REFERENCES

[1] ISO/IEC, *International Standard ISO/IEC 11172–3. Information Technology – Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s – Part 3: Audio*, 1993.

[2] T. S. Verma and T. H. Y. Meng, "A 6 kbps to 85 kbps scalable audio coder," *ICASSP*, vol. 2, pp. 877–880, 2000.

[3] A. Jin, T. Moriya, N. Iwakami, and S. Miki, "Scalable audio coding based on hierarchical transform coding modules," *Electronics and Communications in Japan, Part 3*, vol. 84, no. 8, pp. 34–45, 2001.

[4] B. Leslie and M. Sandler, "Packet loss resilient, scalable audio compression and streaming for IP networks," *2nd International Conference on 3G Mobile Communication Technologies*, pp. 119–123, 2001.

[5] D. Ning and M. Deriche, "A new audio coder using a warped linear prediction model and the wavelet transform," *ICASSP*, vol. 2, pp. 1825–1828, 2002.

[6] S. Levine and J. O. Smith, "A switched parametric and transform audio coder," *ICASSP*, vol. 2, pp. 985–988, 1999.

[7] H. Purnhagen and N. Meine, "HILN – The MPEG-4 parametric audio coding tools," *IEEE International Symposium on Circuits and Systems*, vol. 3, pp. 201–204, 2000.