A SCALABLE AND PROGRESSIVE AUDIO CODEC

Mark. S. Vinton and Les. E. Atlas
Department of Electrical Engineering, University of Washington, Box 352500, Seattle, WA 98195-2500, USA

Abstract

A source coding technique for variable, bandwidth-constrained channels such as the Internet must do two things: offer high quality at low data rates, and adapt gracefully to changes in available bandwidth. Here we propose an audio coding algorithm that is superior on both counts. It is inherently scalable, meaning that channel conditions can be matched without the need for additional computation. Moreover, it is compact: in subjective tests our algorithm, coded at 32kb/s/channel, outperformed MPEG-1 Layer 3 (MP3) coded at 56kb/s/channel (both at 44.1kHz). We achieve this simultaneous increase in compression and scalability through use of a two-dimensional transform that concentrates relevant information into a small number of coefficients.

1 Introduction

The advent of the Internet has fueled interest in streaming audio. Several subband/transform audio coding techniques [1,2] have been applied to this problem. These methods offer good compromises between coding rate and quality, and give acceptable results using 48 kilobits per second per channel at a sample rate of 44.1 kHz. Unfortunately, bandwidth over the Internet is not only scarce—it is also highly variable. A good source coding technique for this channel must therefore offer not only compression, but also scalability: it must gracefully adapt to changing channel capacity after encoding.

In this paper we present an audio coding algorithm that offers improvements in both bit rate and scalability. Both gains are rooted in our use of a two-dimensional transform that concentrates relevant information into a small number of coefficients. We begin by describing this transform. We then give a brief overview of the audio coding scheme that is built around this transform, and illustrate its inherent scalability. Lastly we present the results of simple subjective tests that compare our algorithm to MPEG-1 Layer 3 (MP3).

2 Two-Dimensional Transform Design

Transform-based audio coders achieve compression by using signal representations such as lapped transforms [1,2] and pseudo quadrature mirror filters [3]. Typically, these representations offer the advantage that quantization effects can be mapped to areas of the signal spectrum in which they are least perceptible.

Prior research has explored two-dimensional energetic signal representations where the second dimension is the transform of the time variability of signal spectra [4,5]. This second dimension is often called the "modulation dimension" (e.g. [6]). When applied to signals, such as speech or audio, that are effectively stationary over relatively long periods, this second dimension projects most the signal energy into a few low modulation frequency coefficients. Moreover, mammalian auditory physiology studies have shown that physiological importance of modulation effects decreases with modulation frequency [7]. While these traits suggest an approach for ranking the importance of transmitted coefficients and coding at very low data rates, this past work has provided an energetic yet not invertible transform. What is instead needed is a transform, which after modification to a lower bit rate, is invertible back to a high-fidelity signal.

Our paper shows that there are modulation frequency transforms that are indeed invertible after quantization. Our design allows for essentially CD-quality music coding at 32 kilobits per second per channel and provides a progressive encoding which naturally and easily scales to bit rate changes.

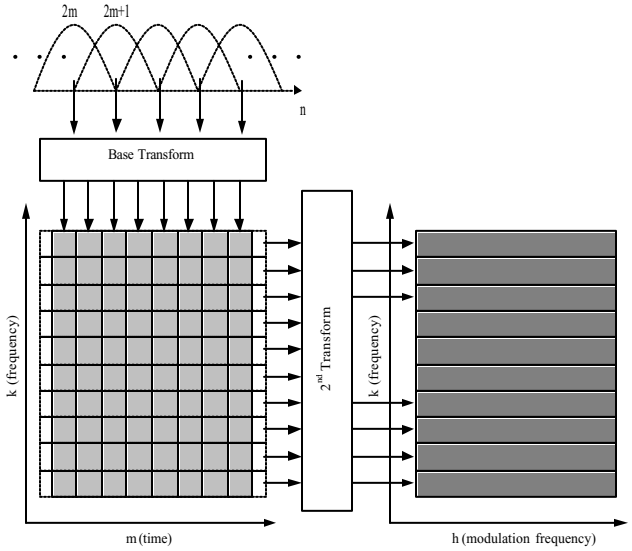

Figure 1. Simplified structure of the two-dimensional transform.

Figure 1 depicts a simplified overview of the new transform. To obtain a signal transform that allows signal modification in the transform domain, care must be taken to prevent distortion caused by edge discontinuities. For this reason, the two-dimensional transform was derived from the time domain aliasing cancellation (TDAC) filter bank introduced by Princen and Bradley [8], which provides a 50% overlap in time while maintaining critical sampling. The input signal $x[n]$ is windowed by a Kaiser-Bessel [9] windowing function $w_1[n]$, to achieve the window constraints defined in [8]. The windowed input is then transformed by either a modified discrete cosine transform (MDCT) or a modified discrete sine transform (MDST) depending on the shift index as defined by equations 1 and 2 below. This base transform process is essentially the TDAC filter bank as described in [8]. Two adjacent MDCT and MDST are combined into a single complex transform, as defined in equation 3. As illustrated in figure 1 the magnitude from the aforementioned transform is composed into a time-frequency distribution. The two-dimensional magnitude distribution is windowed across time in each frequency bin, again with a 50% overlap and windowing function $w_2[n]$, a raised cosine. The second transform, which is given by equation 5, is computed to give the magnitude matrix. The second transform is not performed on the phase information. The phase data is

encapsulated in a corresponding matrix to simplify understanding, as shown in equation 6.

$$X_m^C[k] = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x[n+2mK]\, w_1[n] \cos\left(\frac{2\pi(n+N_0)k}{N}\right) \quad (1)$$

$$X_m^S[k] = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x[n+(2m+1)K]\, w_1[n] \sin\left(\frac{2\pi(n+N_0)k}{N}\right) \quad (2)$$

Where $K = \frac{N}{2}$, $N_0 = \frac{K+1}{2}$ and $k = 0,1,...,K$

$$X_m^D[k] = X_m^C[k] + jX_m^S[k] \quad (3)$$

$$X_\ell^{Mag}[h,k] = \begin{bmatrix} \sqrt{\frac{2}{H}} \sum_{m=0}^{H-1} \left|X_{m+\ell P}^D[0]\right| w_2[m] \cos\left(\frac{2\pi(m+H_0)(h+0.5)}{H}\right) \\ \vdots \\ \sqrt{\frac{2}{H}} \sum_{m=0}^{H-1} \left|X_{m+\ell P}^D[K]\right| w_2[m] \cos\left(\frac{2\pi(m+H_0)(h+0.5)}{H}\right) \end{bmatrix} \quad (5)$$

Where $P = \frac{H}{2}$, $H_0 = \frac{P+1}{2}$ and $h = 0,1,..,P-1$

$$X_\ell^{Phase}[h,k] = \begin{bmatrix} \arg\left(X_{\ell P}^D[k=0]\right) & \cdots & \arg\left(X_{\ell P+P-1}^D[k=0]\right) \\ \vdots & \ddots & \vdots \\ \arg\left(X_{\ell P}^D[k=K]\right) & \cdots & \arg\left(X_{\ell P+P-1}^D[k=K]\right) \end{bmatrix} \quad (6)$$

An example of the two-dimensional transform is shown in the bottom panel of figure 2. The top of this figure shows the spectrogram of two notes of a glockenspiel and the bottom shows our modulation transform of the same time segment. As expected, most of the energy is constrained to lower modulation frequencies. For example the tones at 1 kHz, 4 kHz, and 7.5 kHz in the spectrogram plot (top of figure 2) are mapped to only the low modulation frequencies in the two-dimensional transform plot (bottom of figure 2). However the sudden onset of the tones at 4.5 kHz and 9 kHz results in significantly more energy in the high modulation frequencies. This example, of a known hard-to-encode signal, shows an unusually large extent in modulation frequency due to the abrupt change of note. However, perceptual importance drops with increase in modulation frequency. If the length of the block transforms in each dimension are selected correctly, cutting high modulation frequency information only leads to damping of transient spectral changes, which is not perceptually annoying. In our informal experiments, we found that for most audio signals the overall information contained in the two-dimensional transform can be reduced by more than 75% before the onset of any significant perceivable degradation.

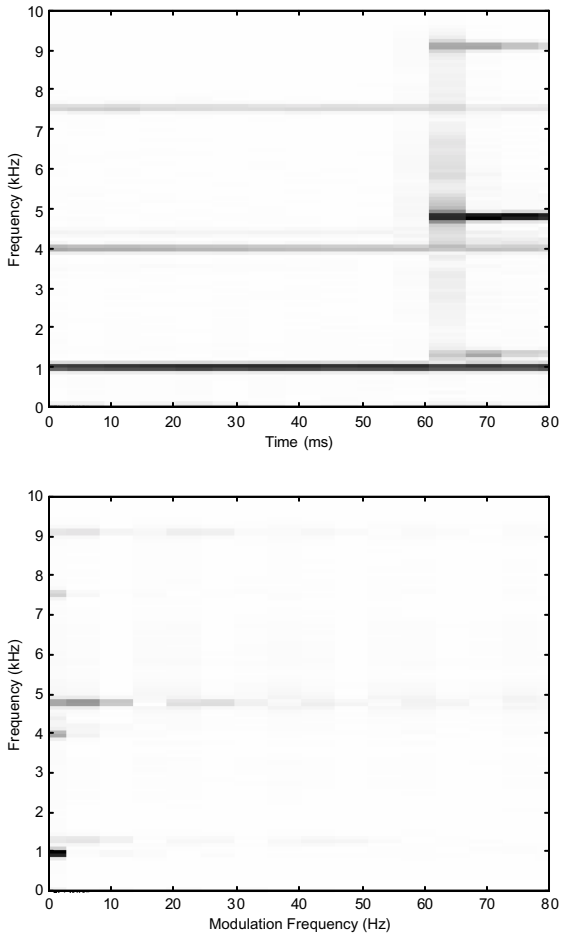

Figure 2. Spectrogram (top) and two-dimensional transform (bottom) of a glockenspiel signal.

3 Coder Structure

Adaptive signal coders can take on one of two fundamental frameworks: forward or backward adaptive. Forward adaptive architectures imply that the encoder makes all adaptive decisions and transmits pertinent information for decoding as side information. The benefits of such schemes are reduced decoder complexity; access to more detailed information and an encoder structure that can be improved in isolation. Backward adaptive frameworks make adaptations based on transmitted data alone. Such structures give up the aforementioned benefits of the forward adaptive scheme to reduce the extra bits of side information. Use of the two-dimensional transform described in the previous section lends itself very well to the backward adaptive architecture, reducing side information yet still offers detailed information for adaptive decisions.

In order to map noise generated by quantization into areas of the spectrum where they are least perceptible, most audio compression algorithms calculate a model of the human auditory system [10]. Such models are based on an estimate of power spectral density of the incoming signal, which can only be accurately computed in the encoder. Our proposed two-dimensional transform has the advantage of providing an implicit power spectral density estimate: As the first column of the magnitude matrix, as defined in equation 5, represents an approximate mean spectral density (MSD), it is used to compute a perceptual model and bit allocation for the remaining information.

A simplified block diagram of the proposed encoder is shown in figure 3. The input $x[n]$ is passed through a gain normalization procedure (GN) and then through the two-dimensional transform discussed in the previous section. The first column of the magnitude matrix (MSD) is used to compute a psychoacoustic model and bit allocation for the remaining magnitude matrix coefficients and the phase matrix. The first step in this process is to remove scale factors; the peak values of the MSD (first column of the magnitude matrix) are extracted from frequency groups approximately representing the critical band structure of the human auditory system. These are then converted to a logarithmic scale and quantized to give 1.5dB precision. The scale factors are then used to compute a bit allocation for quantization of the MSD, which is implemented via table lookup, taking advantage of simple perceptual criterion. The MSD is then inverse quantized and used in the core perceptual model to derive bit allocations for the remaining data.

The remaining magnitude matrix and phase matrix are then quantized and the magnitude matrix coefficients are Huffman coded. To ensure that the target rate is met, the data from the magnitude and phase matrices are reordered into the bit stream with respect to their perceptual relevance. As discussed in Section 2, the high frequencies in both dimensions are less important and, if need be, can be removed without dire consequences. Transmission of the information in a single frame simply terminates when the target data rate has been met. This progressive aspect of the proposed algorithm will be further discussed in Section 5.

4 Quantization and Variable Length Coding of the Magnitude and Phase Matrices

Both the magnitude and phase matrices are uniformly quantized and the magnitude matrix is coded with a single dimensional Huffman code. The wrapped phase matrix is not variable length coded for the obvious reason that it is has a uniform distribution. To prevent excessive consumption of bits to represent the phase matrix, at low data rates the phase information at frequencies above 5 kHz are not transmitted and are replaced by randomized phase in the decoder. This process does not lead to significant perceptual loss, as will be shown in the results of the subjective tests.

Due to the slow non-stationarity of most audio inputs, the magnitude matrix displays very low entropy. Even with the use of only a single dimensional Huffman code, more than 40% of the redundancy is extracted. This is not an optimal coding technique: run length coding and multidimensional variable length coding techniques would lead to further gains. However, these methods interfere with the desired scalability of the technique and were therefore avoided.

5 Progressive Encoding and Algorithm Scalability

The key feature of the two-dimensional transform proposed in Section 2, is its capacity to isolate relevant information to the low frequencies of the modulation frequency axis. The proposed algorithm exploits this not only to obtain high quality at low data rates but also to achieve scalability.

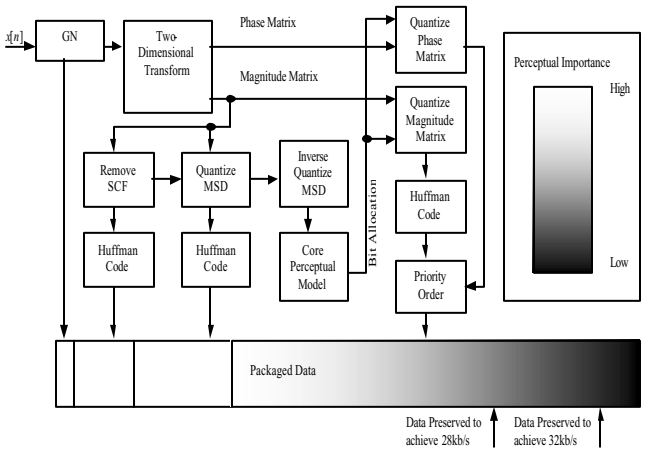

Figure 3. Simplified structure of the proposed encoder. The coded data is progressively ordered and placed into the bit stream with respect to perceptual relevance. Frame transmission terminates when a final target data rate has been met or is at channel capacity.

As discussed in Section 2, more than 75% of the information contained in the magnitude matrix can be discarded without causing annoying perceptual losses. The proposed encoder takes advantage of this concept to reach the desired data rate, without having to use the bit reservoirs or computationally intense bit allocation recursions as used in MPEG-1 layer 3 [11]. To achieve this simplicity the data is arranged into a frame packet with respect to perceptual relevance and transmission of the frame simply stops when the target data rate has been met, as shown in figure 3. Thus the data is progressively encoded.

The same technique is used to accommodate for variable channel coding conditions, without performing additional calculations. As depicted in figure 3, if the channel capacity is less than the encoded data rate, the frame data packet is simply truncated to accommodate for channel requirements.

6 Subjective Tests of the Proposed Codec

The qualitative performance of the proposed algorithm was evaluated using a simple subjective test. The experimental protocol was as follows:

- Subjects were presented with three versions of each audio selection: the unencoded original, an encoded signal A, and an encoded signal B.
- Subjects could listen to each selection as many times as desired.
- In each test, subjects were asked to indicate which, if any, of the encoded signals were of higher quality.
- Three different pairs of signals were used for the encoded A and B signals (here the encoding rates are bits/sec/channel):
  - Group 1: proposed at 32k vs. unencoded original
  - Group 2: proposed at 32k vs. MP3 at 48k
  - Group 3: proposed at 32k vs. MP3 at 56k
- The MPEG-1 Layer 3 (MP3) encoder used was the ISO MPEG audio software simulation group's source code.
- The proposed algorithm used in this test had a block size of 185ms for the sample rate of 44.1kHz.
- Each such test was performed using three songs:
  1. Roxette "Must Have Been Love."
  2. Duran Duran "Notorious."
  3. Go West "King of Wishful Thinking."

A total of 25 people participated in this experiment. The cumulative results are shown in figures 4 through 6. Figure 4 shows the cumulative results for the tests comparing our proposed algorithm at 32 kbits per second per channel to the original 44.1 kHz compact disk source. A slight majority (56%) of subjects preferred the original source. The rest of the subjects could not distinguish the difference or preferred our encoded version. Figure 5 shows the results from the comparison of our 32 kbit encoding to MP3 coded at 48 kbits, which shows the proposed algorithm was clearly preferable. Figure 6 shows a comparison of our 32 kbit encoding with MP3 coding at 56 kbits per second per channel, which demonstrates a similar strong trend verifying the advantages of our proposed algorithm.

8 Conclusion

A new audio compression algorithm was presented which takes advantage of a two-dimensional transform to remove redundancies implicit in slowly non-stationary input signals, while allowing sufficient control to prevent annoying perceptual losses. Simple subjective tests were performed and the results presented suggested that the proposed algorithm performed better quality coding at 32kb/s/channel than MPEG-1 Layer 3 coding at 56kb/s/channel. Furthermore, the proposed algorithm was shown to be inherently progressively scalable, lending itself well to applications where bandwidth cannot be known prior to coding.

References

1. H. Malvar, "Enhancing the Performance of Subband Audio Coders for Speech Signals," *IEEE Int. Symp. On Circuits and Sys.*, Monterey, CA, June 1998.
2. T. Mirya, N. Iwakami, A. Jin, K. Ikeda, S. Miki, "A design of Transform coder for both Speech and Audio Signals at 1bit/Sample," *IEEE ICASSP '97*, Munich, pp. 1371-1374.
3. P. Monta, S Cheung, "Low Rate Audio Coder With Hierachical Filterbanks and Lattice Vector Quantization," *IEEE ICASSP '94*, pp. II 209-212, 1994.
4. Drullman, R., Festen, J. M. and Plomp, R. "Effect of temporal envelope smearing on speech reception." *J. Acoust. Soc. Am*. **95**, pp. 1053-1064, 1994.
5. Y. Tanaka and H. Kimura, "Low Bit-Rate Speech Coding using a Two-dimensional Transform of Residual Signals and Waveform Interpolation," *IEEE ICASSP '94*, Adelaide, pp. I 173-176.
6. S. Greenberg and B. Kingsbury "The modulation spectrogram: In pursuit of an invariant representation of speech," in ICASSP-97, *IEEE ICASSP '97*, Munich, pp. 1647- 1650.
7. N. Kowalski, D. Depireux and S. Shamma, "Analysis of Dynamic Spectra in Ferret Primary Auditory Cortex: I. Characteristics of single unit responses to moving ripple spectra," *J. Neurophysiology* **76**, pp. 3503-3523, 1996.
8. J. Princen, A. Bradley, "Analysis/Synthesis Filter Bank Design Based on Time Domain Aliasing Cancellation," *IEEE Trans. Acoust., Speech, and Signal Processing* **34**, pp 1153-1161, 1986.
9. C. Todd, G Davis, L. Fielder, B. Link, and S. Vernon, "AC-3: Flexible Perceptual Coding for Audio Transmission and Storage," *AES 96th Convention*, Amsterdam, 1994.
10. J. D. Johston, "Trasnfrom Coding of Audio Signals Using Perceptual Noise Criteria," *IEEE J. Selected Areas Commun*. **6**, pp. 314-323, 1998.
11. D. Pan, "A Tutorial on MPEG/Audio Compression," *IEEE Multimedia Journal*, pp.60-74, Summer 1995.

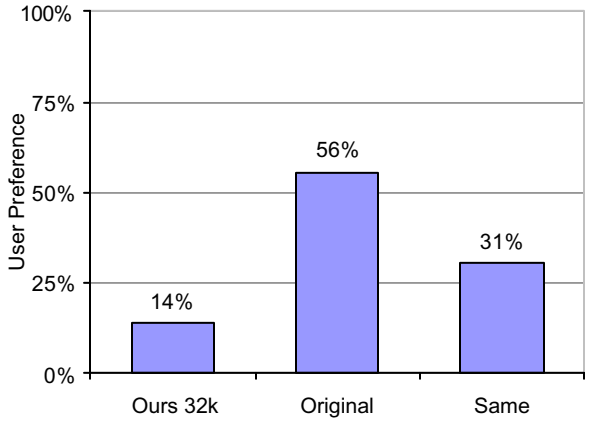

Figure 4. Listener preferences between the proposed algorithm at 32Kbits/sec and the original CD.

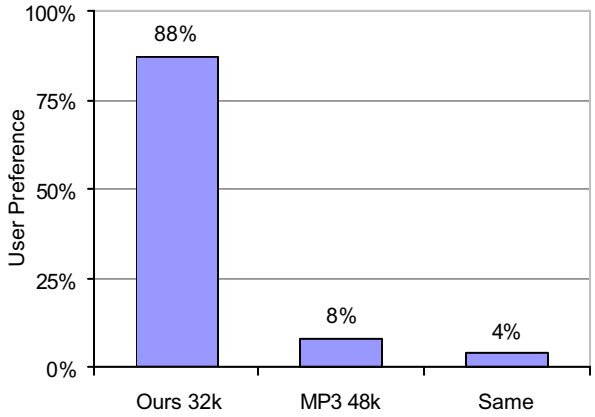

Figure 5. Listener preferences between the proposed algorithm at 32Kbits/sec and MP3 at 48kb/s.

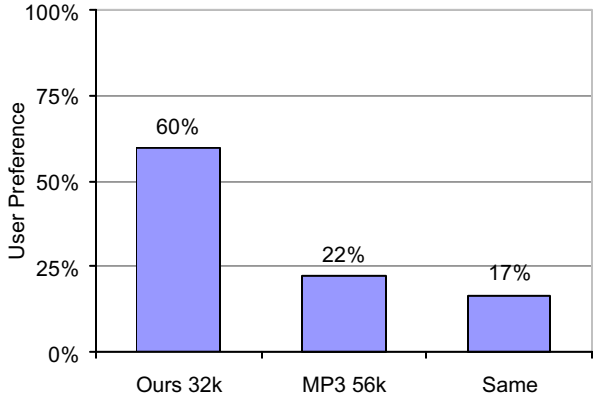

Figure 6. Listener preferences between the proposed algorithm at 32Kbits/sec and MP3 at 56kb/s.