LOSSLESS CODING OF MPEG-1 LAYER III ENCODED AUDIO STREAMS

Farshid Golchin and Kuldip K. Paliwal
School of Microelectronic Engineering
Griffith University
Brisbane, QLD 4111, Australia
F.Golchin, K.Paliwal@me.gu.edu.au

ABSTRACT

In this paper we describe a lossless coding scheme for the encoding of MPEG-1 Layer III encoded audio bitstreams. Commonly known as MP3, the MPEG-1 Layer III standard has proved widely popular for the transmission of encoded audio files (MP3's) over the Internet. However, the MPEG-1 Layer III standard has been designed with a wide range of applications in mind. As such, the frame sizes are kept small and redundancies between samples in neighboring frames are not exploited. We propose a design which uses a combination of Linear Predictive Coding and Arithmetic Coding to exploit such redundancies. The proposed coder was tested on a number of Layer III encoded audio (MP3) files and shown to produce an average coding gain of 12.2% over the original Layer III encoded files.

1. INTRODUCTION

Since the acceptance of the ISO MPEG-1 standard in 1992 [1], MPEG Audio coding has been used in a variety of applications for transmission and storage of high quality digital audio. In particular, the Layer III component of this standard which is the most sophisticated coding method offered by MPEG-1 has proved widely popular.

Commonly known as MP3 (named after its designated file extension), MPEG-1 Layer III Audio coding has gradually established itself as the defacto standard for the transmission of high quality audio signals, particularly for transmission of music over the Internet. There are tens of thousands of MP3 files freely available on the Internet. These files can be downloaded to the user's hard disk drive and used for future listening. MP3 files are kept in their encoded Layer III format and only decoded at the time of listening (in real-time).

Considering MP3's mainly recreational use, it is no surprise that most users download these files from home, and using an analogue modem. Current modem speeds of 56 kbps and lower mean that a typical 5 minute audio clip with a file size of around 4 to 6 Megabytes will take at least 10 minutes to download.

In addition to the use of MP3 files on PC's, portable MP3 players have been recently introduced to the market which store MP3 files on a flash memory card. The limited size of flash memory cards (typically 32MB or 64MB) places a limit on the amount of audio which can be stored on the device.

These factors necessitate the development of more sophisticated coders which can result in reduced file sizes and faster download times. Currently there are a number of more advanced audio coders available which outperform MPEG-1 Layer III. A notable example of such coders is MPEG2-AAC [3]. However, given due to the wide popularity of MPEG-1 Layer III, the availability of inexpensive decoding solutions and its current status as de-facto standard for audio transmission over the Internet, MPEG-1 Layer III is likely to remain quite popular in the near future.

Therefore, we were motivated to investigate the possibility of encoding (Re-Compressing) MPEG-1 Layer III (MP3) encoded bitstreams. The coding scheme is designed such that an MPEG-1 Layer III encoded file can be encoded and then decoded to produce the exact same MPEG-1 Layer III bitstream. Hence, lossless coding techniques must be used.

In this paper, we describe one such lossless coding scheme and provide coding results for a set of typical MP3 files.

2. MPEG-1 LAYER III AUDIO

Figure 1 depicts the operation of a MPEG-1 Layer III encoder [1],[2]. In the Layer III encoder, the incoming audio signal is decomposed into 576 frequency lines (MPEG terminology for subbands). This decomposition is performed in two stages. The first stage is a subband decomposition which decomposes the input signal into 32 subbands. The second stage is the MDCT [1] (Modified Discrete Cosine Transform) which further decomposes each subband into 18 transform coefficients, resulting in 576 frequency lines.

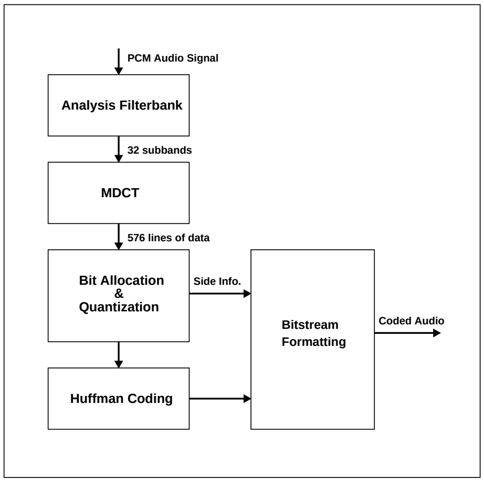

Figure 1. Block diagram of MPEG-1 Layer III encoder

After the subband decomposition and MDCT, bit-allocation is performed based on psychoacoustic criteria and the samples in each frequency line are quantized according to the allocated bit budget. The quantized frequency lines and the side information are then organized into distinct frames. Each frame contains data from 2 granules, where each granule contains a set of 576 quantized frequency lines and their associated side information. The side information is stored in the frame header.

After quantization, the quantized samples are losslessly encoded by the Huffman coder. The Huffman coder actually uses a combination of run-length encoding (for long zero runs at high frequencies) as well as regular Huffman coding.

Finally, the Huffman encoded data and the side information are organized frame-by-frame and transmitted in the format specified by the MPEG-1 Layer III standard. The bitstream is formatted such that each frame of encoded data can be decoded individually and without requiring any additional information.

If the audio source contains more than one channel (eg. stereo audio), then the coded and quantized lines from these channels are interleaved so that each granule contains data from more than one channel.

Please note that a great many important details of the MPEG-1 Layer III standard have been ignored in this brief description. For further information, the reader is referred to [1] and [2].

We note that in Layer III encoding, subsequent to the the subband decomposition and the MDCT, the quantization and Huffman coding are performed by only taking into account samples from the frequency lines which belong to the same granule. This means that potential redundancies that may exist between subsequent samples from a given frequency line are not exploited.

Even though the samples in a given frequency line have been de-correlated to a large degree (as a result of the subband decomposition and MDCT), it is still possible to achieve further coding gain by exploiting the redundancies that exist within the frequency lines (intraband redundancies).

The main reasons why this type of redundancy have not been exploited in MPEG-1 Layer III: keeping the frame size small and lowering the computational complexity of the decoder. The advantages of using smaller frames are:

1. Less buffer memory required in the decoder.
2. Increased robustness to channel errors.
3. Lower coding delay.

However, the increasing availability of large amounts of memory and processing power means that memory and computational complexity have become less of an issue. Additionally, robustness to channel errors and low-coding delay are often not a factor when it comes to MP3 files. The coding is performed offline and the files are reliably transmitted through the Internet. Therefore, we set out to investigate any coding advantages that my arise by exploiting intraband redundancies and effectively increasing the frame size.

As mentioned previously, we intend to use lossless coding so that an MP3 file can be coded and then decoded to produce the original MP3 file. For this reason, the Layer III transforms, bit allocation and quantization are left intact. Our experiments are limited to modifying the lossless encoding (Huffman coding) which occurs after quantization. In the following sections, we describe the operation of the coder and provide coding results.

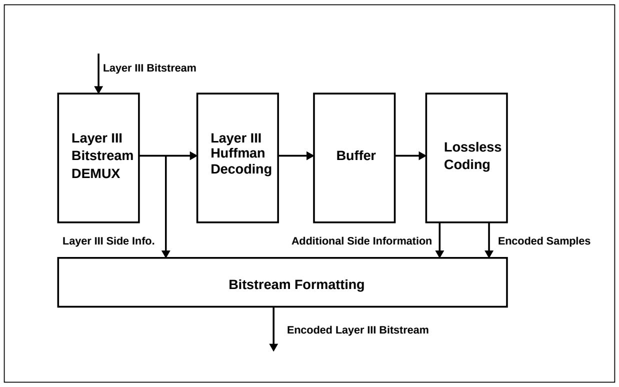
Layer III Bitstream
Layer III Bitstream DEMUX
Layer III Huffman Decoding
Buffer
Lossless Coding
Layer III Side Info.
Additional Side Information
Encoded Samples
Bitstream Formatting
Encoded Layer III Bitstream

Figure 2. Block diagram of proposed coder

3. THE PROPOSED CODING SCHEME

In this section we describe the proposed coding scheme for lossless encoding of MPEG-1 Layer III encoded audio files.

Figure 2 depicts the structure of the proposed coder. To be able to utilize the data contained in the Layer III bitstream, we begin by demultiplexing the bitstream and reading the side information at the beginning of Layer III frames. The side information is used to perform the Huffman decoding. However, the side information is kept intact and a copy is sent directly to the output bitstream so that it may be used for future Layer III decoding.

After reading the side information, we used the Layer III Huffman decoder to decode the quantized samples from each frame and buffer them for re-compression.

The quantized samples stored in the buffer are then grouped into blocks of frames and encoded using the lossless encoder. The ideal block size depends on the Layer III bitrate as well as the lossless coding scheme being used. For the lossless coding scheme used in this experiment and Layer III bitrates of 128kbps and higher, a block size of equivalent of 64 Layer III frames (128 granules) was found to produce the best results.

The Lossless coder also produces its own side information which is sent to the output stream and formatted along with the Layer III side information and the encoded samples. The lossless coder is described in detail in the following section.

4. THE LOSSLESS CODER

We propose to devise a lossless coding scheme which exploits intraband redundancies in Layer III encoded bitstreams. There are two properties of the quantized subband samples which are exploited in the proposed coder:

1. There is some remaining correlation amongst the quantized samples within each frequency line. This correlation is significantly higher in the lower frequency lines.
2. Neighboring blocks of samples within each frequency line have similar probability distributions.

The proposed coder utilizes a combination of DPCM type Predictive Coding [4] and Arithmetic Coding [5], [6] to exploit these properties. Figure 3 depicts a block diagram of the proposed encoder. The individual components of the coding scheme are described in the following sections.

A. Predictive Coding: We wish to exploit the correlation amongst neighboring samples within individual frequency lines. Linear Predictive Coding (LPC) is used to predict the value of a sample from previous samples in the same frequency line.

Let $s(i,j)$ denote sample number i from the jth frequency line. We predict the value of $s(i,j)$ as the linear combination of previous samples from the same frequency line:

$$\hat{s}(i,j) = \sum_{k=1}^{k=N} a_k \; s(i-k,j). \tag{1}$$

where $\{a_k\}$ denote the Linear Prediction Coefficients (LPC's) which are calculated using an LP analysis method. N is the order of the predictor.

The residual value $d(i,j)$ is then calculated as:

$$d(i,j) = s(i,j) - round(\hat{s}(i,j)) \tag{2}$$

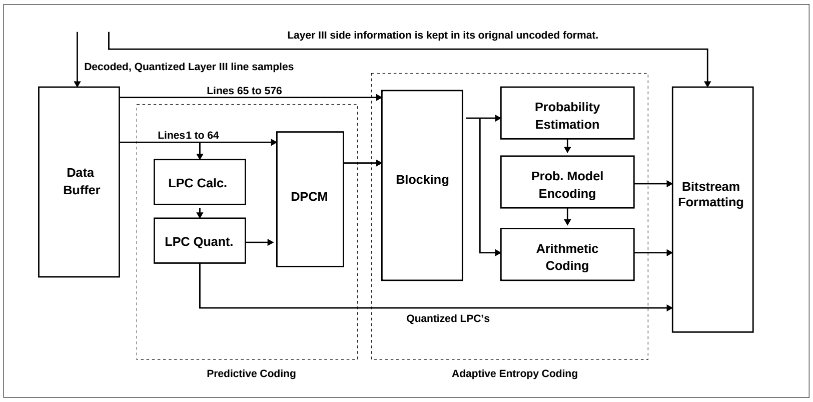

Figure 3. Block diagram of lossless coder

The rounding operation is performed to restrict the residual $d(i,j)$ to integer values.

The predictive coding is made adaptive by re-calculating LPC's for blocks of 128 samples in each frequency line. The LPC's are then quantized and transmitted as side information. It was experimentally found that for Layer III encoded files at bit rates of 128 kbps and higher, 5th order LPC's quantized at 5 bits per coefficient produce satisfactory results.

However, the coding gain due to Predictive Coding varies for different frequency lines. As expected, the gain is higher at lower frequencies and gradually decreases for higher frequency lines. We are therefore faced with a trade-off between Predictive Coding gain and the amount of side information required to transmit the quantized LPC's. Hence, the predictive coding is only performed for the lower frequency lines where the coding gain outweighs the overheads. It was experimentally found that encoding the lowest 64 frequency lines performs generally well.

B. Blocking of Samples: The residual samples (lines 0 to 63) calculated by the predictive coder are regrouped with the uncoded samples (lines 64 to 576) and buffered. The buffer contains samples from 64 frames (128 granules) of Layer III data.

The buffered samples $s(i,j)$ are then grouped into M by N sized, non-overlapping blocks. The blocks $B_{m,n}$ are defined as:

$$B_{m,n} = \{(i,j) \mid m \leq i < m+M \ , \ n \leq j < n+N\} \qquad (3)$$

C. Adaptive Entropy Coding: The samples in each block are entropy coded using an Arithmetic Coder. The Arithmetic Coder encodes each sample according to its conditional probability $P(s(i,j) \mid (i,j) \in B_{m,n})$. By using different probability models for different blocks, the entropy coding is made adaptive.

However, Arithmetic Coding requires the probability models to be also available to the decoder and hence the probability tables must also be transmitted as side information. Transmitting the probability tables presents a challenge, since without further encoding the added side information by far exceeds the coding gain achieved through Arithmetic Coding.We use two different encoding methods for transmitting the probability tables. Examining the distribution of sample values in the lower frequency lines reveals that the probability distribution can be well approximated by a Laplacian distribution which is defined as:

$$p(x) = \frac{1}{\sqrt{2}\sigma} e^{\frac{-\sqrt{2}}{\sigma}} |x| \qquad (4)$$

Therefore, we only need to estimate the standard deviation σ and transmit it to the decoder. The value of σ is estimated and then quantized using a scalar quantizer with a resolution of 9 bits and transmitted as side information. This quantized value is also made available to the Arithmetic Coder to use for generating probability tables.

However, the Laplacian is only an approximation which is well matched to the statistics of lower frequency regions, but becomes less accurate at higher frequencies. The mis-match in the probability models results in longer codes. Since the distribution of values becomes significantly narrower in the higher frequency regions, we opt to directly encode the probability tables for blocks in high frequency regions. Due to the symmetrical distribution of values around zero, we use the absolute value of samples $|s(i,j)|$ to construct the probability tables.

The probabilities rapidly decrease as sample value $|x|$ increases and hence we only transmit the first few probability values. The number of probability values transmitted N_P is determined as the 80th percentile of the probability distribution.

Log probabilities for values less than N_P are then quantized using an 7-bit scalar quantizer and transmitted as side information.

In this experiment, the probability tables for blocks from the first 64 frequency lines are encoded using the Laplacian approximation. The probability tables for blocks in the higher lines are directly encoded as described above. To avoid excessive side information, we stipulate that for the higher frequency lines where N_P exceeds 6, we fall back to the first method and the Laplacian approximation is used.

The highest frequency lines (typically around line 400 and higher) contain large blocks of zeroes. For better encoding of such blocks, we define a new type of block as Zero Block in which all the sample values are zero. The Zero Blocks are encoded only using 1 bit of side information per block to indicate whether or not it is a Zero Block.

D. Bitstream Formatting: After the coding process, the original Layer III side information, the quantized LPC's, the encoded probability models and the arithmetic coded data are blocked together and written to a file. Each block of data corresponds to 64 Layer III frames.

It should be noted that for stereo Layer III bitstreams, the two channels are treated separately. However, each block of data now contains both channels in a similar fashion to a Layer III bitstream.

E. Uniform and Non-Uniform Blocks: An intuitive notion which can be confirmed by experimentation, is that the lower frequency sound components tend to persist for longer periods of time whereas higher frequency components often arise out of short-lived temporal events.

This leads us to believe that there may be some added advantage in using non-uniform block sizes. That is, we use blocks longer blocks (more samples from each frequency line) for the low frequency lines and shorter blocks for the higher frequency lines.

Table 1 shows experimentally determined block sizes which result in a noticeable coding gain for Layer III bitrates of 128kbps and higher. In the uniform case, block sizes of 64x16 (samples x lines) were found to produce the best results. The coding results for these two cases are reported and compared in the following section.

Frequency Region (lines)	Block Size (samples x lines)
1 .. 16	256 x 16
17 .. 32	128 x 16
33 .. 64	64 x 16
65 .. 128	32 x 16
129 .. 576	16 x 16

Table 1. Non-uniform block sizes for entropy coding.

5. CODING RESULTS

The proposed coding scheme was tested on a sample of typical MP3 songs which can be found on the Internet. We provide coding results and comparisons using the proposed coder with Uniform Block Sizes (UBS), the proposed coder with Non-Uniform Block Sizes (NUBS) and PKZIP which is a general purpose file compression utility.

Table 2 contains a list of MP3 files which were used in experiments. All of these files contain Layer III encoded popular songs which are representative of the type of MP3 files which are transmitted via the Internet. These MP3 files have been encoded at a bitrate of either 128 kbps or 160 kbps. Coder parameters such as block sizes, LPC order, LPC Quantization, etc. were optimized based on the results obtained from a different set of MP3 files which were encoded at the same bitrates.

MP3 File	File size(kb)	Bitrate(kbps)
Madonna 1	6414	128
Ottmar Leibert	3724	128
Wamdue Project	7899	128
Sneaker Pimps	4582	128
Madonna 2	5932	128
Breakbeat Era	6192	160
Portishead	5929	160

Table 2. List of MP3 files used for testing the encoder.

The coding results are provided in Table 3. The first two columns of results in this table show coding results for the coder with Uniform Block Sizes (UBS) compared with the same coder using Non-Uniform Block Sizes (NUBS).

The results are quoted in kilobytes (kb) and are based on file sizes. The percentage values quoted for average compression are based on the ratio of the average size of coded files to the average size of original files.

The best results are obtained from the coder with Non-Uniform Block Sizes. This coder was able to compress the MP3 files by an average of 12.2%. As expected, Non-Uniform Block Sizes outperform Uniform Block Sizes. The difference however is smaller than expected at only 1.1%. This suggests that further gain from using Non-Uniform blocks may be obtained by using an adaptive scheme to adaptively find optimal block sizes.

In comparison, GZIP which is a general data compression utility can compress the MP3 files by an average of only 1.6%

We also note that the coding gains obtained for the 160kbps MP3 files and 128kbps MP3 files are quite similar. The average compression for the 160 kbps files is 11.9% compared to 12.3% for the 128 kbps files.

MP3 Name	UBS(kb)	NUBS(kb)	GZIP(kb)
Madonna 1	5677	5632	6329
Ottmar Leibert	3425	3401	3679
Wamdue Project	7209	7110	7749
Sneaker Pimps	4129	3941	4534
Madonna 2	5677	5632	6329
Breakbeat Era	5562	5488	6132
Portishead	5221	5187	5764
Average Compression	11.1%	12.2%	1.6%

Table 3. Coding results and comparison

6. DISCUSSION

In this paper we have described a lossless coding technique for encoding MPEG-1 Layer III encoded bitstreams. It was demonstrated that it is possible to achieve additional coding gain by exploiting intraband redundancies within the MPEG-1 Layer III frequency lines.

Considering the complexity of the proposed coder, the reported coding gains of around 12% (for Non-Uniform Block Sizes) are quite modest. However, it can serve as a basis for future work in this direction.

The coding gain obtained from the adaptive arithmetic coding of blocks of subband samples is largely offset by the need to transmit the source model for each block. It may however be possible to reduce this overhead by using a finite set of codebooks in the manner described by the authors in [7].

In this paper, we highlight a very specific case of how the use of larger frames can improve coding performance. The MPEG-1 Layer III standard, uses small frames which reduce the memory requirements of the decoder and make it suitable for a wide variety of applications. However, we believe that the huge demand for transmission of encoded audio over the Internet warrants the development of better coding schemes which cater specifically for this application.

REFERENCES

[1] 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" (1992).

[2] K. Brandenburg and G. Stoll, "ISO-MPEG-1 Audio: A Generic Standard for Coding of High-Quality Digital Audio," *J. Audio Eng. Soc.*, vol. 42, pp. 780-792, October 1994.

[3] ISO/IEC 13818-3 "Information Technology - Generic Coding of Moving Pictures and Associated Audio, Part 3: Audio" (1994-1997).

[4] N. S. Jayant and P. Noll, "Digital Coding of Waveforms," Prentice Hall, Englewood Cliffs, NJ, 1984.

[5] J. Rissanen and G. G. Langdon, "Arithmetic Coding," *IBM J. Res. Develop.*, vol. 23, pp. 149-162, March 1984.

[6] I. H. Witten, R. M. Neal and J. G. Cleary, "Arithmetic Coding for Data Compression," *Communications of the ACM*, pp. 520-540, vol. 30, No. 6, June 1987.

[7] F. Golchin and K.K. Paliwal, "Minimun-entropy clustering and its application to lossless image coding," *Proceedings of the IEEE International Conference on Image Processing*, pp. 262-265, Santa Barbara, October 1997.