LOSSLESS AUDIO COMPRESSION IN IEEE 1857.2 STANDARD FOR ADVANCED AUDIO CODING

Adrià Romero López
Dan-Mihai Bădescu

Universitat Politècnica de Catalunya (UPC), Barcelona, 2015

ABSTRACT

In August 2013, IEEE approved a new standard that comprises both lossy and lossless audio compression tools. This new standard is called IEEE 1857.2. This paper focus on the lossless audio compression tool, which utilizes a pre-processing procedure for flattening the amplitude envelop of the linear prediction residue, and an arithmetic encoder that adopts a scaled probability template.

Index Terms*—Lossless audio compression for advanced audio coding, IEEE 1857.2*

1. INTRODUCTION

Generally, the multimedia contents are pre-compressed before they are uploaded online. Normally, there are two types of compression methods: the lossy and the lossless. The lossy method attempts to remove perceptually less important information from the audio data while keeping the sound quality very close to the original one. Some examples that include this type of compression method are: MPEG-1 Layer (MP3) and the MPEG-2/4 Advanced Audio Coding (AAC) which achieve more than twenty times compression and still delivering good sound quality. On the other hand, the lossless method keeps every bit of information from the original audio data and can achieve about two times compression.

Lossy audio compression is mainly used for playing music from iPods, or listening to networked radio using mobile phones. In contrast, lossless audio reproduction, archival of database and more recently biomedical signal compression, such as lossless ECG compression [1].

A generally approach used in lossless audio compression, is a combination of linear prediction and entropy encoding. The linear predictor first removes the redundancy in the input data and generates a prediction residue, which is encoded by the entropy encoder. The system is based on a LPC predictor, a pre-processor for flattening the prediction residue, and an entropy coder that is based on arithmetic coding with probability template scaling [1].

2. IEEE 1857.2 LOSSLESS AUDIO COMPRESSION

The Fig.1 illustrates the block diagram of the IEEE 1857.2 lossless audio compression system, where the top part is the encoder while the bottom part is the decoder. Observing the encoder, the input audio samples are first processed by the predictor, which removes the correlations and generates a prediction residue. This residue passes to the pre-processor, which flattens the amplitude envelop of the signal, or in others words reduces the dynamic range. Finally, the flattened prediction residue is coded by an entropy encoder into a lossless bitstream. An entropy decoder decodes the lossless bitstream where the flattened residues are recovered and then de-flattened by the post-processor. The re-constructor recovers the decoded signal, which is an exact replication of the original input audio.

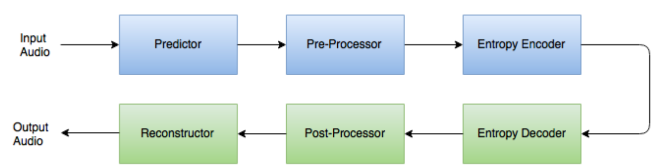

Fig.1 IEEE 1857.2 lossless audio compression: encoding (blue) and decoding (green) [1]

2.1 PREDICTION AND RECONSTRUCTION

The block diagram of the predictor is shown in Fig.2. Input audio samples are first segmented into frames of fixed length. For each frame is performed linear predictive coding (LPC). Observing the Fig.2 to calculate the LPC coefficients, first the partial-correlation (PARCOR) coefficients must be calculated using the Levinson-Durbin algorithm. The PARCOR order used is 20. In other words, the result is a matrix with the number of frames as columns and the PARCOR order rows, in this case 20. These PARCOR coefficients are then quantized, where the resulting matrix has the same dimension. For the quantification the following equations are used:

$$q\_parq(i) = \begin{cases} \left\lfloor 64\left(\dfrac{ln\left(\frac{2}{3}+\frac{5}{6}\sqrt{\frac{1+paqr(1)}{2}}\right)}{ln\left(\frac{3}{2}\right)}\right)\right\rfloor, i = 1 \\ \left\lfloor 64\left(\dfrac{ln\left(\frac{2}{3}+\frac{5}{6}\sqrt{\frac{1-paqr(1)}{2}}\right)}{ln\left(\frac{3}{2}\right)}\right)\right\rfloor, i = 2 \\ \lfloor 64\, parq(i)\rfloor, i = 3, \dots, PARCOR\_order \end{cases}$$

[2]

where "parq" are the PARCOR coefficients and "q_parq" the quantized PARCOR coefficients, which are restricted to the range [-64,63]. This matrix with the quantized coefficients goes directly to the pre-processing block. It is also de-quantized with the following formula:

$$deq\_parq(i) = \begin{cases} \left(2\cdot\left(\left(exp\cdot\left(\dfrac{q\_parq(1)}{64\,log\left(\frac{3}{2}\right)}\right)-\dfrac{2}{3}\right)\cdot\dfrac{6}{5}\right)^2\right)-1, & i = 1 \\ \left(2\cdot\left(\left(exp\cdot\left(\dfrac{q\_parq(1)}{64\,log\left(\frac{3}{2}\right)}\right)-\dfrac{2}{3}\right)\cdot\dfrac{6}{5}\right)^2\right)+1, & i = 2 \\ \dfrac{q\_parq(i)}{64}, & i = 3, \dots, PARCOR\_order \end{cases}$$

This step is necessary to calculate the LPC matrix of coefficients. The LPC coefficients matrix is a three-dimension matrix with, PARCOR order columns and rows for each frame, in other words, (parcor_order, parcor_order, n_frames) dimension. These coefficients are calculated with the following formula:

$$m = 1, \dots, PARCOR\_order$$

$$lpc_i^{(m)} = lpc_1^{(m-1)} + deq\_parq(m)lpc_{m-i}^{(m-1)}$$

$$lpc_m^{(m)} = deq\_parq(m)$$

$$i = 1, \dots, m-1$$

Finally, the prediction residues must be calculated. Before that the Linear Predictor generates a prediction, "y", that is subtracted from each frame, resulting the prediction residues. The prediction residues matrix, "residues" is a two-dimension matrix and it has the number of frames columns and PARCOR order rows. It can be obtained with the following equation:

$$y(i) = \begin{cases} frames(i), \; i = 0 \\ \sum_{j=1}^{i} lpc_j^{(i)} frames(i-j), \; 1 \le i \le PARCOR\_order - 1 \\ \sum_{j=1}^{PARCOR_{order}} lpc_j^{(PARCOR_{order})} x(i-j), \; i \ge PARCOR\_order \end{cases}$$

[1]

$$residues(i) = frame(i) + y(i), \qquad i = 1, \dots, n\_frames$$

where "lpc()" is the prediction matrix with the prediction coefficients and "frames" is the matrix with the frames of the original audio input . These residues are sent to pre-processing.

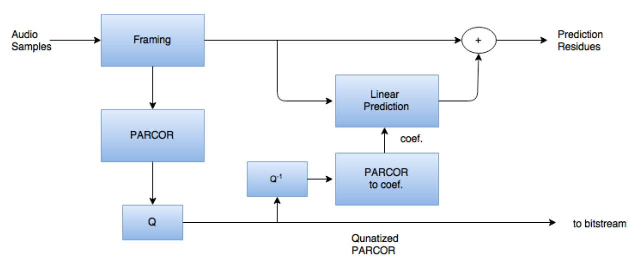

Fig.2 Predictor [1]

In reconstruction, the quantized PARCOR coefficients are extracted from the bitstream sent by post-processor, de-quantized, and converted to LPC coefficients like in the predictor block. The linear predictor generates a prediction, which is added to the decoded prediction residue to reconstruct the original input sample.

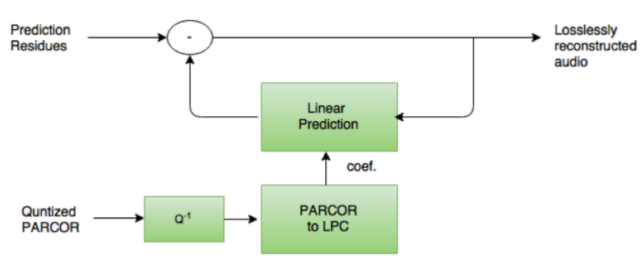

Fig.3 Re-constructor [1]

2.2 PRE- AND POST-PROCESSING

In IEEE 1857.2 lossless audio compression data, it is designed that each frame can be decoded independently without using information from other frames. There are two benefits of this: firstly, compressed audio files can be decoded at a granularity of one frame interval, and secondly, bit error occurred during transmission do not propagate beyond boundaries.

Since linear prediction is performed intra-frame for samples at the beginning of each frame, the orders of the predictor used are very short, which result in larger prediction residues compared to the rest of the frame [1]. This leads to a problematic situation because increase the dynamic range of the prediction residues which force the entropy encoder to increase the alphabet size and the computational complexity.

To solve this problem, it is necessary a previous step between the predictor and the entropy encoder. This new block is the pre-processing block. The block diagram of the pre-processing is shown in Fig.6. Observing the diagram, to reduce the amplitude of the prediction residues, they must be downshifted, so the amplitude envelope is flattened ensuring a smaller dynamic range, which can be encoded by the entropy encoder using a smaller alphabet. This situation can be observed in Fig. 4 and Fig. 5 .

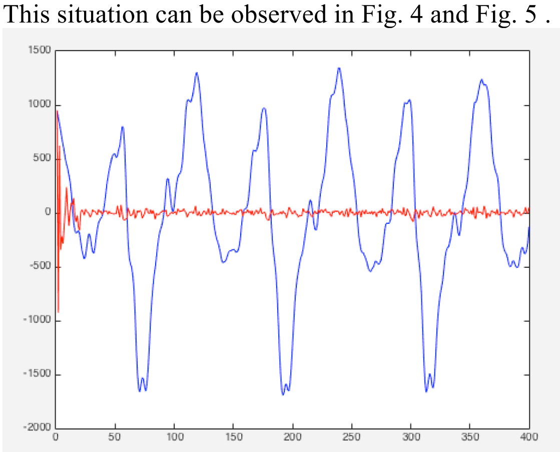

Fig.4 Input waveform (blue plot) and Prediction Residues (red plot)

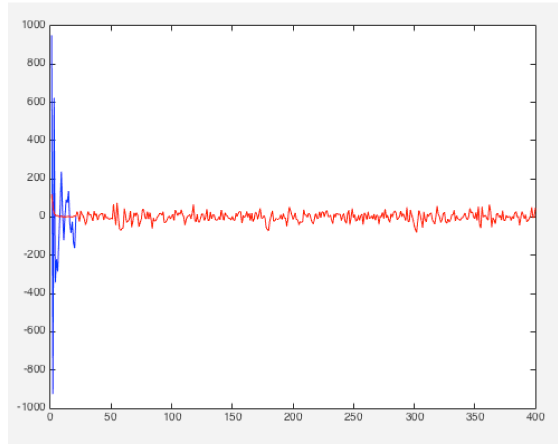

Fig.5 Prediction Residues (blue plot) and Flattened Residues (red plot)

To do the downshift operation, first the number of downshifts must be calculated using the following equation:

$$shift(n) = \begin{cases} \sum_{k=1}^{n+1} RA\_shift12\left[q_{\_parq(k)}\right] & n = 0,1 \\ \sum_{k=1}^{2} RA\_shift12[q\_parq(k)] + \sum_{k=3}^{n+1} RA_shift[|q_{parq(k)}|] & 2 \le n \le L-1 \end{cases} \quad [2]$$

where "RA_shift" and "RA_shift12" are given tables by the IEEE 1857.2 standard, "q_parq" the matrix with the quantized coefficients calculated in the predictor and "L" the length of the signal at the beginning of the frame where the amplitude of the prediction residues is high. It is the maximum between the PARCOR order and 16. The "shift" matrix has dimension of the number of frames columns and L rows. Finally, using the "shift" matrix, the system proceeds to the down-shift operation where the Less Significant Bits (LSB data) and the residues and the flattened data are obtained. The "LSB_data" matrix has the same dimension as "shift" and goes directly to the post-pressing block while the "flat_data" matrix has dimension of number of frames columns and frame length rows and it goes to the entropy encoder.

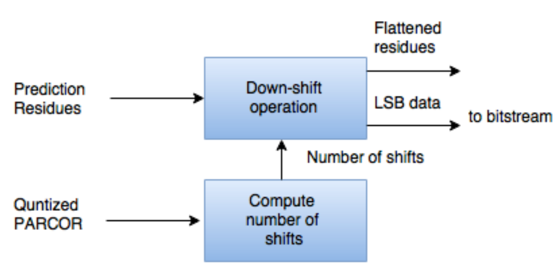

Fig.6 Pre-Processor [1]

In the post-processing block the analogue operation is done. It receives the flattened residues from the entropy decoder, the LSB directly from the pre-processor and the quantized PARCOR coefficients. With the PARCOR coefficients it calculates the number shifts again. Then the up-shift operation to reconstruct the prediction residues from the flattened residues and the LSB data. Those go directly to the re-constructor.

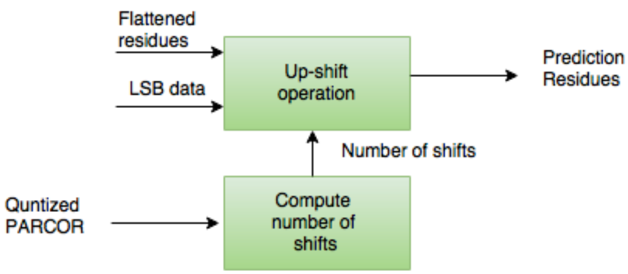

Fig. 7 Post-Processor [1]

3. ENTEROPY CODING

The entropy encoder is based on the arithmetic coding. The bloc diagram of the entropy encoder is shown in Fig.7. For each frame of flattened prediction residue, the mean value of the frame, μ, is computed as:

$$\mu = \frac{\sum_{i=1}^{frame\_length} |flat\_data(i)|}{frame\_length}$$

[1]

For the purpose of the coding, the value μ is logarithmically quantized as an integer, "q_ μ", as:

$$q\_\mu = \lfloor log_2 \mu + 0{,}5 \rfloor$$

[1]

which is then locally de-quantized to:

$$deq\_\mu = 2^{q\_\mu}$$

[1]

where "deq_μ" is the de-quantized mean of the frame. This is used to scale a probability template to generate a probability table for arithmetic coding as follows:

$$p(s) = f\left(\left|{}^{s}/_{deq\_\mu} + 0{,}5\right|\right)$$

[1]

where "s" denotes symbols on the probability distributions, "f(s)" is the probability template, and "p(s)" is the probability table to be used by the encoder.

The probability template is approximated by a Gaussian function with mean -0,1, standard deviation 0,6 and doted line.

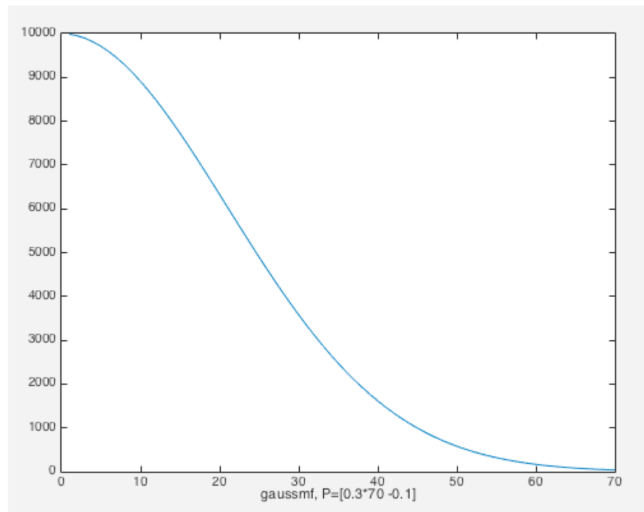

Fig. 8 Probability template for arithmetic coding

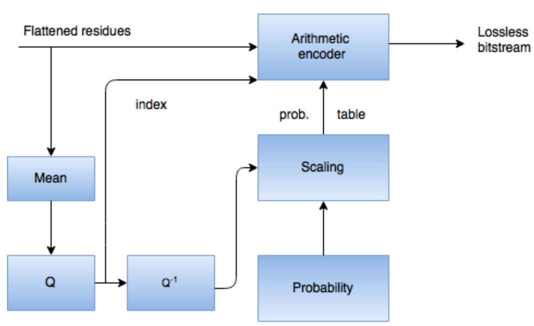

Fig. 9 Entropy encoder [1]

The entropy de-coder diagram block can be observed in Fig.10. The only difference respect the encoder is that it is not necessary to calculate the mean and quantized it. The index is obtained directly from the entropy decoder and is de-quantized to scale the probability template

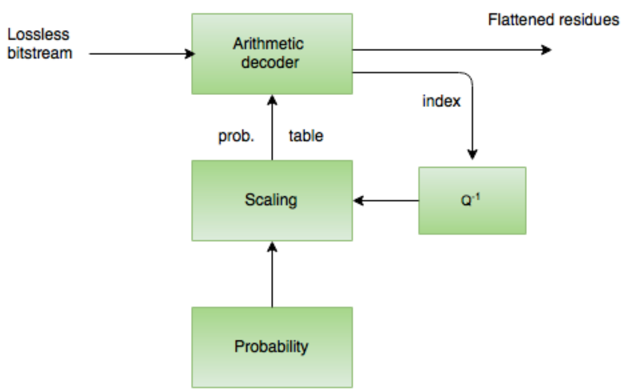

Fig. 10 Entropy decoder [1]

4. RESULTS

The results of the paper are shown with the comparison between the input audio wave and the output. Due to the audio compression tool is lossless, the results tells that audio and output signals are exactly the same.

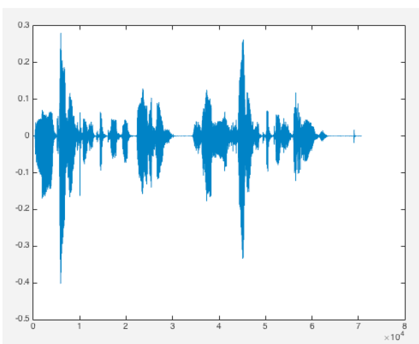

Fig. 11 Audio input

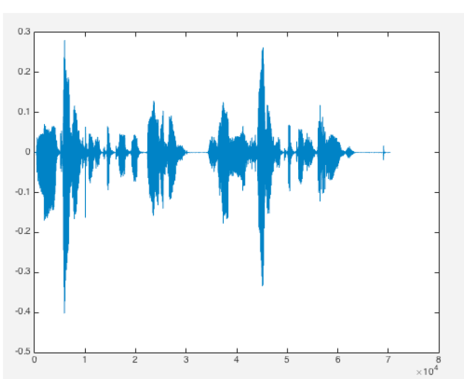

Fig. 12 Audio output

After the entropy encoder, the lossless bitstream has a size of 94,072 kB, whereas the input audio file is 141 kB. Therefore, the compression is 33,33%.

The performance evaluation was tested on a laptop computer with 2,4 GHz Intel Core i5 processor and 8 GB 1600 MHz RAM. The time of execution of all functions was around 8 seconds time.

5. CONCLUSIONS

This paper presents the implementation of the lossless audio compression tool in the IEEE Standard for Advance Audio Coding (IEEE 1857.2). Performance evaluation results show that the lossless compression performance of the IEEE 1857.2 is quite slow but the output signal of the tool is exactly the same as the input.

12. REFERENCES

[1] H. Huang, H. Shu, and R. Yu, Lossless Audio Compression in the new IEEE Standard for Advance Audio Coding, Institute for Infocomm Research, Agency of Science, Technology and Research (A*STAR), Singapore, 2014

[2] IEEE Standard for Advance Audio Coding (IEEE 1857.2), IEEE Standard Association, IEEE Computer Society , New York, 2013.