

Adaptive Cross-Correlation Compression Method in Lossless Audio Streaming Compression

Nguyen Tuan Anh, Hoang Thi Kim Dung, Nguyen Thi Phuong Nhung

Abstract— In this paper, we propose a novel method called Adaptive Cross-Correlation Compression Method (ACCM) to compress audio data using Cross-Correlation value. The original audio data are lossless compressed with variable frame length. In each frame, the goal is to remove the unused space using cross-correlation values between the samples, collapse the repeat values and data encryption. Both Arithmetic and Huffman coding have been applied for entropy coding.

Index Terms— Lossless compression, context adaptive, lossless audio compression, stereo compression, ACCM

I. INTRODUCTION

Nowadays, people increasingly higher requirements on the quality of the products, audio is no exception, a lot of audio-visual products are directed to use high-quality audio, the original sound, increasing the number of channels, bit rate and depth increasing, too.

Lossless audio enables audio data compression does not loss information, and can be restored data back to the original completely [1]-[7]. For example, a CD quality audio file, we will compress to reduce the capacity to store, and when the music player running, it can extract to original quality. There are many lossless audio compression methods have emerged, but the compression rate is still low, highly complex and difficult to put these algorithms into embedded systems, so the number of supporting player devices still low.

This paper proposes a new compression method, called Adaptive Cross-Correlation Compression Method (ACCM), the goal of the method is to reduce the maximum capacity by removing all the space which don't use, just save the most needed features to restore the data.

This study method for audio compression, especially multi-channel audio, with high amplitude resolution

ACCM studied about the cross-correlation of the data so that before compression, it needed original audio data of all channels, and cannot compress the compressed files, but the quantization files. The data will be divided into frames, in that, each frame is mapped into each byte in the header, the header data will determine the method of data compression and decompression. In this paper, ACCM uses different techniques, by convention, in order to ensure minimal data compress.

Some other data compression methods are also used, such as the MPEG-4 Audio Lossless Coder (ALS) based on linear adaptive forward prediction [1].

Nguyen Tuan Anh, School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, P.R.China, Phone/ Mobile No. (+86)13250282662

Hoang Thi Kim Dung, Faculty of Civil and Environment, Thai Nguyen University of Technology, Thai Nguyen, Vietnam

Nguyen Thi Phuong Nhung, Faculty of Electronic Engineering, Thai Nguyen University of Technology, Thai Nguyen, Vietnam

1.1 Audio compression

We learn about the concept of audio compression. Compression of audio data (do not confuse with dynamic range compression), is likely to reduce transmission bandwidth and require storage of audio data. Audio compression algorithms are implemented in a software such as audio codecs. Lossy audio compression algorithms provide higher compression at reasonable cost and are used in many audio applications. These algorithms almost all rely on sound learning to eliminate or reduce the fidelity of the less audible sound, thus reducing the space needed to store or transmit them. [9]

In both lossy and lossless compression, the redundancy of information is reduced, using methods such as encryption, pattern recognition, and linear prediction to reduce the amount of information used to represent uncompressed data.

The compromise is acceptable between reduced audio quality and transmission or storage size depending on the application. For example, a 640MB compact disc contains approximately one hour of uncompressed, less than two hours of mono music, or 7 hours of compressed MP3s at medium speed. A digital voice recorder can usually store about 200 hours of speech in 640MB. [23]

Lossless audio compression creates a representation of digital data, decompressing to an exact copy of the original audio stream. The compression ratio is about 50-60% of the original size, [24] similar to the lossless data compression. Lossless compression cannot achieve high compression ratio due to the complexity of the waveform and the rapid change in the form of sound. Codecs such as FLAC, Shorten, and TTA use linear prediction to estimate the spectra of the signal. Many of these algorithms use convolution with the $[-1 \ 1]$ filter to whiten or flatten a bit, thus allowing for traditional lossless data compression to work more efficiently. This process is reversed when decompressing.

Many lossless audio compression formats are existing such as Free Lossless Audio Codec (FLAC), Apple's Apple Lossless (ALAC), MPEG-4 ALS, Microsoft's Windows Media Audio 9 Lossless (WMA Lossless), Monkey's Audio, TTA, and WavPack.

1.2 Lossy audio compression

Comparing the audio spectrum of a song in uncompressed format and loss formats, lossy compression spectra differed from uncompressed spectra. In fact, they lose information, but human ears cannot distinguish the effects of changes on quality.

Lossy audio compression is used in a wide range of applications. In addition, digitally compressed audio streams are used in most video DVDs, digital television, streaming media on the internet, satellite and cable radio, and increasingly in terrestrial radio broadcasts. Lossy compression capacity typically achieves far greater than

lossless compression (data of 5 percent to 20 percent of the original stream, rather than 50 percent to 60 percent), by discarding less-critical data.[25]

The innovation of lossy audio compression was to use psychoacoustics to recognize that not all data in an audio stream can be perceived by the human auditory system. Most lossy compression reduces perceptual redundancy by first identifying perceptually irrelevant sounds, that is, discard sounds that are very hard to hear. Those sounds are coded with decreased accuracy or not at all.

Due to the nature of lossy algorithms, audio quality is affected when a file is decompressed and recompressed. This makes lossy compression unsuitable for storing the intermediate results in professional audio engineering applications, such as sound editing and multitrack recording. However, they are very popular with end users (particularly MP3) as a megabyte can store about a minute's worth of music at adequate quality.

1.3 Basic Probability

Statistical theory, widely applied in a wide range of fields, such as economics, forecasting, cybernetics, ... in data compression, statistical theory is widely applied and deep. For the maximum compression rate, we need to know which data has the highest frequency to which frequency is least. In order to encode them into shorter, smaller records and save them to the database. Here is the most basic concept in statistical theory.

Product rule, is also known as the chain rule:

$$P(x, y | H) = P(x | y, H)P(y | H) \tag{1}$$

$$= P(y | x, H)P(x | H)$$

Sum rule:

$$P(x | H) = \sum_y P(x, y | H) \tag{2}$$

$$= \sum_y P(x | y, H)P(y, H)$$

Bayes' theorem - obtained from the product rule:

$$P(y | x, H) = \frac{P(x | y, H)P(y | H)}{\sum_{y'} P(x | y', H)P(y' | H)} \tag{3}$$

1.4 Streaming multi-media

Streaming multimedia media is constantly being received and presented to end users. The verb "to stream" refers to the process of transfer data frame by frame to users immediately, not just downloading files.

In this paper, we study the method lossless compress that can be used with streaming data transfer to compress audio data.

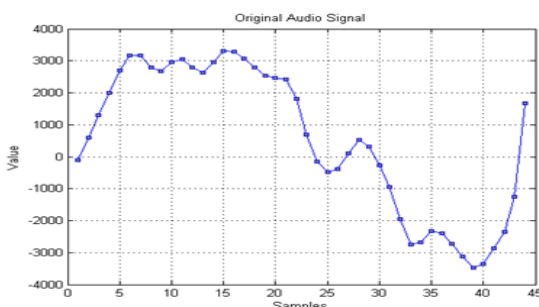


Figure 1. Original audio signal input

II. PROPOSED METHOD

This method is mainly focused on cross-correlation value between samples. Comes from the characteristic of sound, the higher number of bits, bit rate, the higher similar values, and amplitude-varying sinusoidal.

With large amplitude, it is necessary to store larger number of bits, the number of bits in small amplitude actually used will be less. Based on the original audio signal, we assume the audio signal according to (4).

$$X = [x_1, x_2, x_3, \dots, x_n] \tag{4}$$

Where x_i are integers, 16-bit. An example of values of the original audio as shown in Figure 1

As we can see, the value of this signal, the maximum range of about 3200, the number of bits actually used the input signal is 16-bit. Considering the 44 samples, we need the number of bits stored as:

$$44 * 16 = 704 \text{ bit.}$$

We will find ways to reduce the store size of this sample down.

ACCM's approach is to divide the input signal into frames, each frame with different lengths, depending on the variability of the signal. To split the frames, we find varying trends, that is increasing or decreasing values, put in a frame, then find next trend to put in the next frame. The purposes of this division are to make the frame cross-correlation between the values in the frame with the smallest value. To do so, we apply a low pass filter algorithm, to find biggest values of a range. Filtering results are illustrated in Figure 2.

With results of this filter, it is easy to find extreme values for this first sample 44, located at 20 and 42. According to the denominator of the low-pass filter characteristic, the maximum value of the strip in front of poles of the value of filters, so, we find the maximum value in the range of previous max and min value in the range located before the min value.

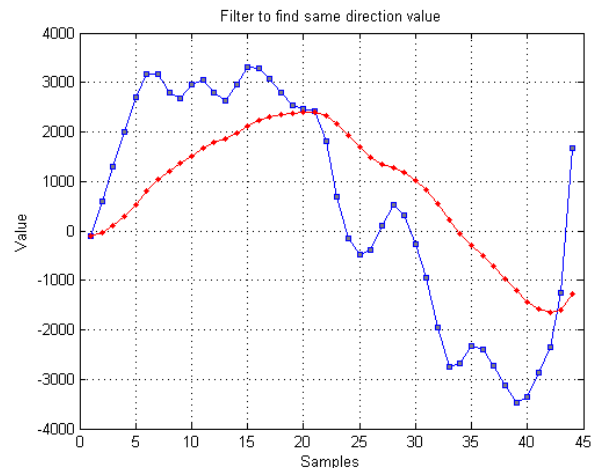


Figure 2. Original signal value and its low-pass filter

Accordingly, we have:

$$x_{mf} = \max(X[x_b : x_e]) \tag{5}$$

Where x_b, x_e and x_{mf} , respectively, is the beginning of the frame, and the end of the frame are looking for, and value of max/min of the selected area. Frame will be selected to handle the $[x_b, x_{b+1} \dots x_{mf}]$. We have strip consecutive frames, data compression depends only on the current frame, it is especially beneficial for streaming signal. According to

Figure 3, we have the consecutive frames, different lengths, depending on the characteristics of the input values.

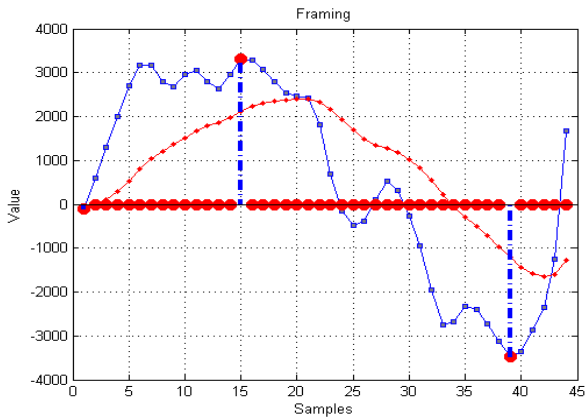


Figure 3. Signal's frames

In Figure 3, we have the 02 frame, $F_1 = [x_1, \dots, x_{15}]$. Each frame is separated vertically by linear data. The value of the data in ascending or descending order, we put in a frame. Select the data with a maximum value near the peak of the filtered signal, as the split frame. Therefore, each frame has different lengths, demonstrating the flexibility of the system. See Figure 4:

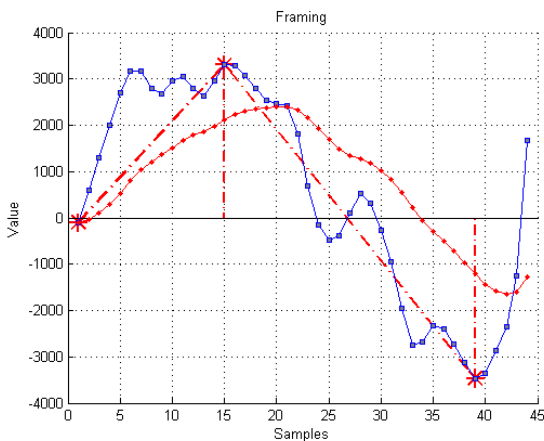


Figure 4. Frames

For higher compression ratio, we used the method to predict upcoming value of the data, based on the characteristics of each sample value of audio, we predict the following:

$$x(n-1) \approx x(n-2) + \Delta \quad (6)$$

$$x(n) \approx x(n-1) + \Delta$$

result as Figure 5.

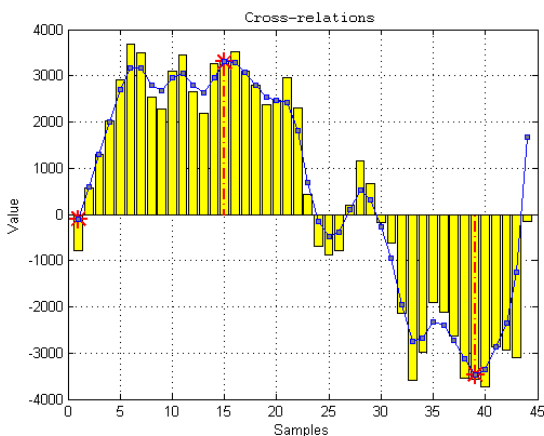


Figure 5. The original data and predictive data

In Figure 5, line chart indicates the original data, bar chart indicating predictive data.

To reduce the size of the original signal, each frame, we store the value of the first frame, and the next value, we store its deviation values to the value of the number in front of it:

$$D_k = x_{k+1} - x_k, k = 1..n_f \quad (7)$$

Where D_k is bias deviation between the signal samples, n_f is the number of signal samples processed in the frame. Doing the same for all the frames respectively, we obtained as Figure 6:

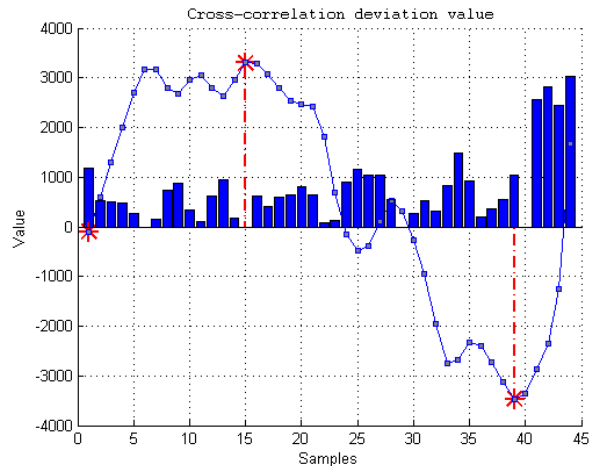


Figure 6. Cross-bias data, used to store

In Figure 6, line chart indicates the original data, the bar graph indicates the need to store data after compression.

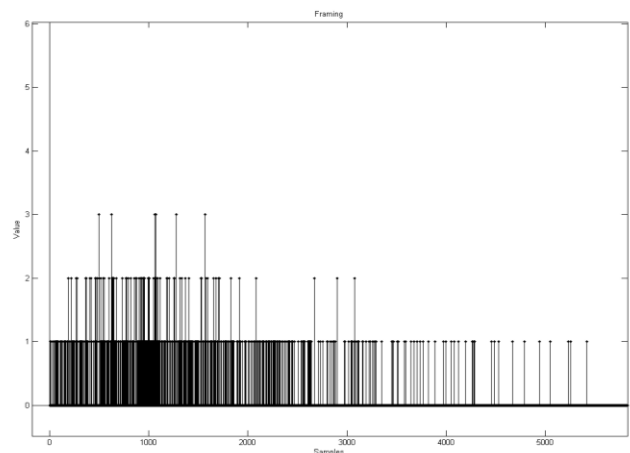


Figure 7. Statistical probability of the need to store values

After calculations, we have the store values, according to (4), the probability of the value of the number needed to store shown in Figure 7.

III. IMPLEMENTATION

Compression Ratio is calculated by the formula:

$$C = \left(\frac{F_{out}}{F_{in}} \right) * 100\% \quad (8)$$

In which, F_{in} and F_{out} are input file size and output file size after compression, respectively. The smaller C, the higher the compression ratio.

In this paper, the test data set is over 100 mp3 files having duration about 6 seconds of music and speech with 44Khz, 16 bits, which contains the number of frames is 250 frames, large enough to be able to encrypt, compress.

Because each file has been split into small frames, each frame quite similar to compress streaming, so results apply directly to stream audio compression very well.

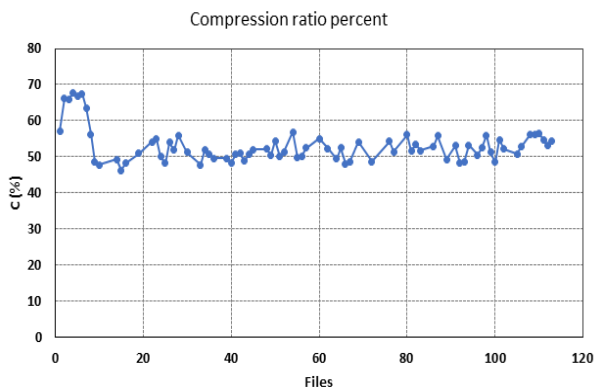


Figure 8. Compression ratio percent

Results received, for speech $C \approx 52.95\%$.

IV. CONCLUSION

Adaptive Cross-Correlation Compression Method was proposed in this study has met the requirements of lossless data compression, fast enough to meet real-time streaming audio transfer with vary audio length. This method is based on the cross-correlation value, the difference between adjacent values, from which only the deviation value is stored. Besides, the algorithm can also predict the next value, optimal support for compression and decompression, to meet the requirements of lossless compression.

REFERECES

- [1]. T. Liebchen, T. Moriya, N. Harada, Y. Kamamoto, Y. Reznik, "The MPEG-4 Audio Lossless Coding (ALS) Standard – Technology and Applications" 119th AES Convention, New York, October 7-10, 2005.
- [2]. Wikipedia: "Audio data compression," available at: http://en.wikipedia.org/wiki/Audio_data_compression.
- [3]. Hydrogen Audio: "Lossless comparison," available at http://wiki.hydrogenaudio.org/index.php?title=Lossless_comparison
- [4]. H. S. Malvar, "Adaptive run-length/Golomb-Rice encoding of quantized generalized Gaussian sources with unknown statistics," Proc. Data Compression Conf., Snowbird, UT, pp. 23–32, Mar. 2006
- [5]. J. Coalsen, "FLAC - Free Lossless Audio Codec," available at <http://flac.sourceforge.net>.
- [6]. M. T. Ashland, "Monkey's audio: a fast and powerful lossless audio compressor," available at <http://www.monkeysaudio.com>
- [7]. Grzegorz Ulacha and Ryszard Stasinski, "Lossless Audio Coding by Predictor Blending", 2013 IEEE
- [8]. Wade, Graham (1994). Signal coding and processing (2 ed.). Cambridge University Press. p. 34. ISBN 978-0-521-42336-6. Retrieved 2011-12-22. The broad objective of source coding is to exploit or remove 'inefficient' redundancy in the PCM source and thereby achieve a reduction in the overall source rate R.
- [9]. Mahdi, O.A.; Mohammed, M.A.; Mohamed, A.J. (November 2012). "Implementing a Novel Approach an Convert Audio Compression to Text Coding via Hybrid Technique" (PDF). International Journal of Computer Science Issues. 9 (6, No. 3): 53–59. Retrieved 6 March 2013.
- [10]. Pujar, J.H.; Kadlaskar, L.M. (May 2010). "A New Lossless Method of Image Compression and Decompression Using Huffman Coding Techniques" (PDF). Journal of Theoretical and Applied Information Technology. 15 (1): 18–23.
- [11]. Salomon, David (2008). A Concise Introduction to Data Compression. Berlin: Springer. ISBN 9781848000728.
- [12]. S. Mittal and J. Vetter, "A Survey Of Architectural Approaches for Data Compression in Cache and Main Memory Systems", IEEE Transactions on Parallel and Distributed Systems, 2015.
- [13]. Tank, M.K. (2011). Implementation of Lempel-Ziv algorithm for lossless compression using VHDL. Thinkquest 2010: Proceedings

- of the First International Conference on Contours of Computing Technology. Berlin: Springer. pp. 275–283.
- [14]. Navqi, Saud; Naqvi, R.; Riaz, R.A.; Siddiqui, F. (April 2011). "Optimized RTL design and implementation of LZW algorithm for high bandwidth applications" (PDF). Electrical Review. 2011 (4): 279–285.
- [15]. Mahmud, Salauddin (March 2012). "An Improved Data Compression Method for General Data" (PDF). International Journal of Scientific & Engineering Research. 3 (3): 2. Retrieved 6 March 2013.
- [16]. Arcangel, Cory. "On Compression" (PDF). Retrieved 6 March 2013.
- [17]. Marak, Laszlo. "On image compression" (PDF). University of Marne la Vallee. Retrieved 6 March 2013.
- [18]. Mahoney, Matt. "Rationale for a Large Text Compression Benchmark". Florida Institute of Technology. Retrieved 5 March 2013.
- [19]. Shmilovici A.; Kahiri Y.; Ben-Gal I.; Hauser S. "Measuring the Efficiency of the Intraday Forex Market with a Universal Data Compression Algorithm" (PDF). Computational Economics, Vol. 33 (2), 131-154., 2009.
- [20]. I. Ben-Gal. "On the Use of Data Compression Measures to Analyze Robust Designs" (PDF). IEEE Trans. on Reliability, Vol. 54, no. 3, 381–388, 2008.
- [21]. Korn, D.; et al. "RFC 3284: The VCDIFF Generic Differencing and Compression Data Format". Internet Engineering Task Force. Retrieved 5 March 2013.
- [22]. Korn, D.G.; Vo, K.P. (1995), B. Krishnamurthy, ed., Vdelta: Differencing and Compression, Practical Reusable Unix Software, New York: John Wiley & Sons, Inc.
- [23]. The Olympus WS-120 digital speech recorder, according to its manual, can store about 178 hours of speech-quality audio in .WMA format in 500MB of flash memory.
- [24]. Coalsen, Josh. "FLAC Comparison". Retrieved 6 March 2013.
- [25]. c d Jaiswal, R.C. (2009). Audio-Video Engineering. Pune, Maharashtra: Nirali Prakashan. p. 3.41. ISBN 9788190639675.