New Highly Efficient Hybrid Lossless Audio Coding Techniques

Hend A. Elsayed
Department of Communication and Computer Engineering, Faculty of Engineering,
Delta University for Science and Technology, Mansoura, Egypt

Abstract:- In this paper two new highly efficient hybrid lossless audio coding techniques based on the Burrows-Wheeler Transform (BWT) and the distance transform (DT) are presented. In both techniques, floating point samples of the audio signal are first applied to the BWT and the resulting coefficients are then applied to the DT to obtain more suitable coefficients for the next step of lossless compression. In the first proposed method, two entropy-based lossless compression methods are considered, namely Arithmetic coding and Huffman coding. On the other hand, in the second proposed method the entropy coding is first preceded by Run Length Encoding (RLE).

Keywords—Burrows-Wheeler Transform; Distance transform; Run Length Encoding; Entropy coding; Audio Coding.

1. INTRODUCTION

The MPEG-4 Audio Lossless Coding (ALS) standard belongs to the family MPEG-4 audio coding standards [1] and [2]. Audio Lossless Coding (ALS) provides methods for lossless coding of audio signals with arbitrary sampling rates, resolutions of up to 32-bit and up to 216 channels.

Lossless audio coding enables the compression of digital audio data without any loss in quality due to a perfect reconstruction of the original signal. On the other hand, modern perceptual audio coding standards are always lossy, since they never fully preserve the original audio data. Those lossy coding methods are typically not well suited for certain applications such as editing or archiving.

This paper introduced a new lossless audio coding schemes using the BWT of the input audio stream and the distance transform to convert the resulting coefficients to a form that can be better compressed then the resulting coefficients are compressed using two methods, the first method is using entropy coding only and the second method is using a combination of the run length encoding and entropy coding.

The paper is organized as follows. Section 2 presents the burrows wheeler transform. Section 3 describes distance transform and section 4 describes the run length encoding. Section 5 shows the simulation results. Finally, Section 6 concludes the paper.

In [3], audio signals that are assumed to be of integer values are used for lossless audio coding using two methods. The first method is the compression using Burrows-Wheeler Transform only and this method called Method1. The second method is using the combined Burrows-Wheeler Transform and Move-to-Front Coding and this method called Method2.

Lossless audio coding technique assumed the audio signal is floating point values is made using Burrows-Wheeler Transform and run length encoding, this method called Method3[4] and a combination of Move-to-Front Coding and run length encoding and this method called Method4 [5] .

Burrows-Wheeler Transform and combination of Distance Transform Coding, and run length encoding are made for lossless text compression [6] and [7], for lossless strings compression [8], and for lossless image compression [9].

2. BURROWS WHEELER TRANSFORM

The Burrows-Wheeler Transform (BWT) is a reversible block sorting transform.

Example, the BWT of the samples X ={1, 2, 4, 6, -1, 3, 2, -4, 6, -7}is given in Table I [3] Considering last column and the fourth row of Table I(b), then BWT{1, 2, 4, 6, -1, 3, 2, -4, 6, -7}= ({ 6, 2, 6, -7, 3, 1, - 1, 2, -4, 4}, 4).

An example for this inverse BWT is found in appendix. The procedures for this IBWT table construction are found in [5].

Then the IBWT({ 6, 2, 6, -7, 3, 1, - 1, 2, -4, 4}, 4)={1, 2, 4, 6, -1, 3, 2, -4, 6, -7}.

3. DISTANCE TRANSFORM

Distance transform (DT) can be used in this paper for preprocessing the Burrows- Wheeler transformed sequence before it is fed to the actual compressor instead of move to front coding. are an important tool in image compression [10] and the used matlab function for this distance transform is found in [11] .

Example, the distance transform coding of the samples y={6, 2, 6, -7, 3, 1, - 1, 2, -4, 4} is{2 -3 -6 -7 -6 -3 -1 -3 -4 -3}.

4. RUN LENGTH ENCODING

The run-length encoding (RLE) can be used to reduce the number of runs in a data.

Example, the one-dimensional sequence pixels of input data {2, 2, 2, 4, 4, 4, 4, 4, 9, 9, 9, 9, 9, 9, 4, 4, 4, 4, 4, 4}, the pixel “2” has repeated with 3 times, the pixel “4” has 5 repetitions, and the pixel “9” has 6 repetitions, and the pixel “4” has 6 repetitions, the values can represented as {2, 3, 4, 5, 9, 6, 4, 6}[5].

5. SIMULATION RESULTS

In this paper, six different audio signals have been used for lossless audio coding using the proposed two methods. The six audio signals are Signal1, Signal2, Signal3, Signal4, Signal5 and Signal6 each of length is 2.32 sec, sampled at 44.1 KHz sampling rate with 16 bits floating point have been considered in this simulation.

Audio signals are divided into frames each of length 23.2 msec (1024 samples). Lossless coding is implemented using Arithmetic and Huffman coding and compared to the entropy of the signal to get the output bit rate in bits per sample (bps).

Figure 1 shows the output bit rate in bps for different audio signals and the average bit rates using the first proposed method for sampling rate 44.1 kHz. From this, we show that the bit rate depends on the signal type and the type of lossless coding using entropy, Huffman, and Arithmetic coding. The bit rate for Arithmetic coding is less than Huffman coding. Then Arithmetic coding is better than Huffman coding.

Figure 2 shows the output bit rate in bps for different audio signals and the average bit rates using the first proposed method for sampling rate 32 kHz. From this, we show that the bit rate depends on the signal type and the type of lossless coding using entropy, Huffman, and Arithmetic coding. The bit rate for Arithmetic coding is less than Huffman coding. Then Arithmetic coding is better than Huffman coding.

Figure 3 shows the output bit rate in bps for different audio signals and the average bit rates using the first proposed method for sampling rate 48 kHz. From this, we show that the bit rate depends on the signal type and the type of lossless coding using entropy, Huffman, and Arithmetic coding. The bit rate for Arithmetic coding is less than Huffman coding. Then Arithmetic coding is better than Huffman coding.

Figure 4 shows the average bit rate for the three different sampling rates 32, 44.1, and 48 KHz using the first proposed method and 23.2 msec frame lengths. From this figure, we show the bit rate depends on the sampling rate; the bit rate for the 44.1 KHz is less than 32 and 48, then it is better than 32 and 48.

Then the audio signals are divided into frames each of length 11.6 msec (512 samples) and applying the first proposed method for compression, figure 5 shows the output bit rate in bps for different audio signals and the average bit rates using the first proposed method for sampling rate 44.1 kHz. From this, we show that the bit rate depends on the signal type and the type of lossless coding using entropy, Huffman, and Arithmetic coding. The bit rate for Huffman coding is less than Arithmetic coding. Then Huffman coding is better than Arithmetic coding.

Then the audio signals are divided into frames each of length 46.4 msec (2048 samples) and applying the first proposed method for compression, figure 6 shows the output bit rate in bps for different audio signals and the average bit rates using the first proposed method for sampling rate 44.1 kHz. From this, we show that the bit rate depends on the signal type and the type of lossless coding using entropy, Huffman, and Arithmetic coding. The bit rate for Arithmetic coding is less than Huffman coding. Then Arithmetic coding is better than Huffman coding.

Figure 7 shows the average bit rate for the three different frame lengths in msec 11.6, 23.2, and 46.4 using the first proposed method and 44.1KHz sampling rate. From this figure, we show the bit rate depends on the frame length; the bit rate for the 11.6 msec is less than 23.2 and 46.4 msec, then it is better than 23.2 and 46.4 msec.

Figure 8 shows the output bit rate in bps for different audio signals and the average bit rates using the second proposed method for sampling rate 44.1 kHz. From this, we show that the bit rate depends on the signal type and the type of lossless coding using entropy, Huffman, and Arithmetic coding. The bit rate for Arithmetic coding is less than Huffman coding. Then Arithmetic coding is better than Huffman coding.

Figure 9 shows the output bit rate in bps for different audio signals and the average bit rates using the second proposed method for sampling rate 32 kHz. From this, we show that the bit rate depends on the signal type and the type of lossless coding using entropy, Huffman, and Arithmetic coding. The bit rate for Arithmetic coding is less than Huffman coding. Then Arithmetic coding is better than Huffman coding.

Figure 10 shows the output bit rate in bps for different audio signals and the average bit rates using the second proposed method for sampling rate 48 kHz. From this, we show that the bit rate depends on the signal type and the type of lossless coding using entropy, Huffman, and Arithmetic coding. The bit rate for Arithmetic coding is less than Huffman coding. Then Arithmetic coding is better than Huffman coding.

Figure 11 shows the average bit rate for the three different sampling rates 32, 44.1, and 48 KHz using the second proposed method and 23.2 msec frame lengths. From this figure, we show the bit rate depends on the sampling rate; the bit rate for the 44.1 KHz is less than 32 and 48, then it is better than 32 and 48.

Then the audio signals are divided into frames each of length 11.6 msec (512 samples) and applying the second proposed method for compression, figure 12 shows the output bit rate in bps for different audio signals and the average bit rates using the second proposed method for sampling rate 44.1 kHz. From this, we show that the bit rate depends on the signal type and the type of lossless coding using entropy, Huffman, and Arithmetic coding. The bit rate for Huffman coding is less than Arithmetic coding. Then Huffman coding is better than Arithmetic coding.

Then the audio signals are divided into frames each of length 46.4 msec (2048 samples) and applying the second proposed method for compression, figure 13 shows the output bit rate in bps for different audio signals and the average bit rates using the second proposed method for sampling rate 44.1 kHz. From this, we show that the bit rate depends on the signal type and the type of lossless coding using entropy, Huffman, and Arithmetic coding. The bit rate for Arithmetic coding is less than Huffman coding. Then Arithmetic coding is better than Huffman coding.

Figure 14 shows the average bit rate for the three different frame lengths in msec 11.6, 23.2, and 46.4 using the first proposed method and 44.1 KHz sampling rate. From this figure, we show the bit rate depends on the frame length; the bit rate for the 11.6 msec is less than 23.2 and 46.4 msec, then it is better than 23.2 and 46.4 msec.

Figure 15 shows the comparison between the Average bit rate using sampling rate of 44.1kHz and frame length of 23.2 msec for the different methods; Method1, Method2, Method3, Method4, Proposed Method1, and Proposed Method2.

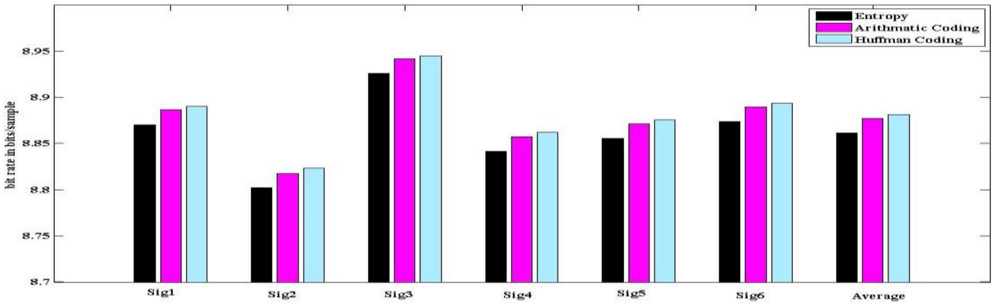

Fig. 1: Bit rate using the first proposed method for different audio signals and Fs=44.1 kHz.

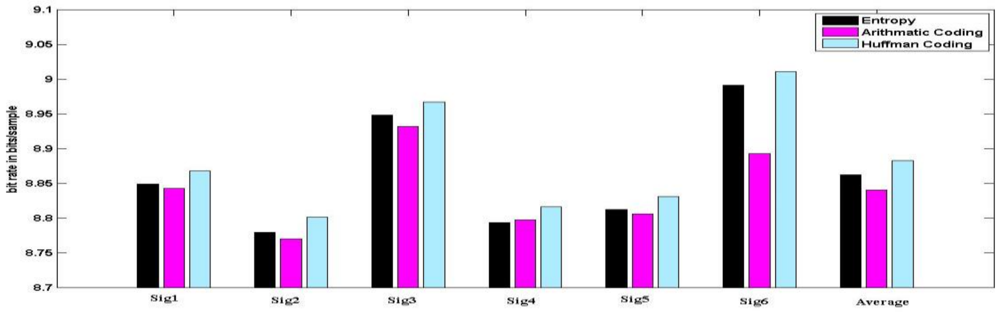

Fig. 2: Bit rate using the first proposed method for different audio signals, Fs=32 kHz.

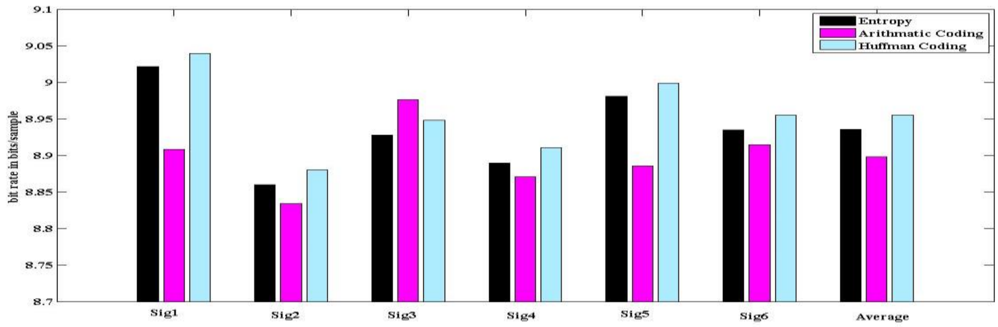

Fig. 3: Bit rate using the first proposed method for different audio signals, Fs=48 kHz.

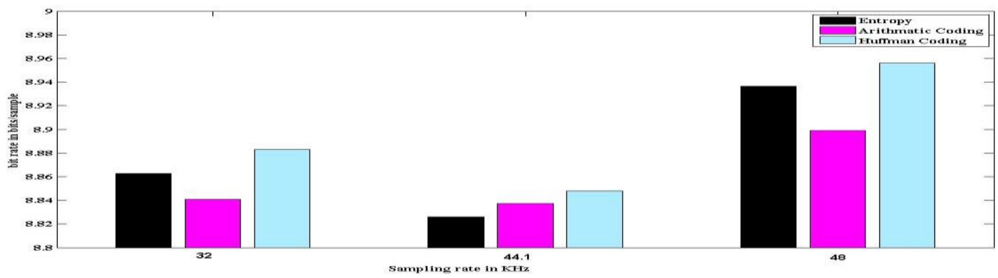

Fig. 4: Average bit rate for the first proposed method using different sampling rates.

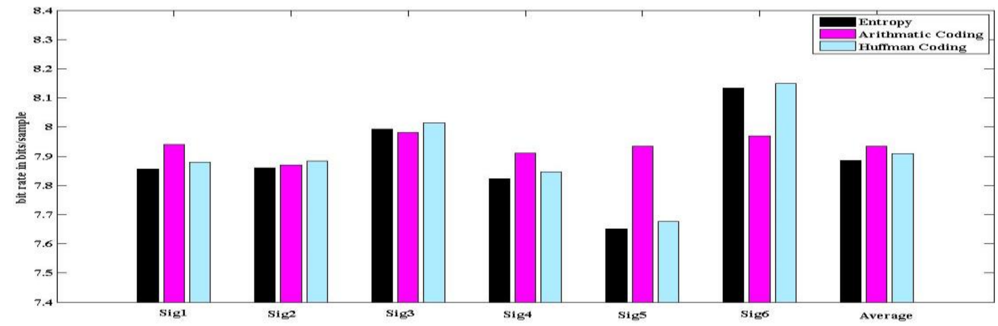

Fig. 5: Bit rate using the first proposed method for different audio signals, Fs=44.1 kHz and L=512.

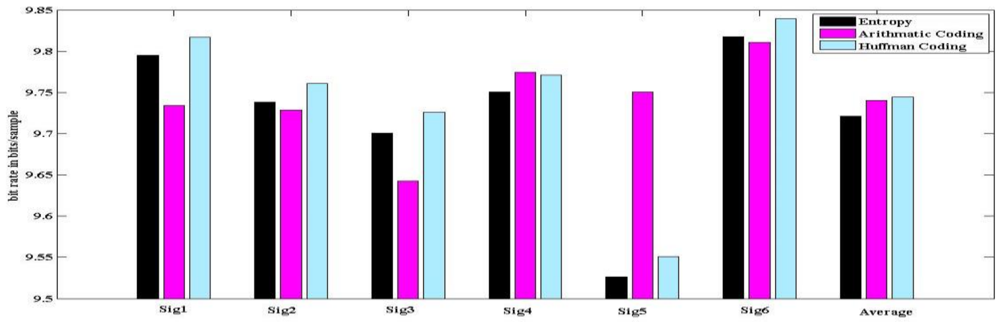

Fig. 6: Bit rate using the proposed method for different audio signals Fs=44.1 kHz and L=2048.

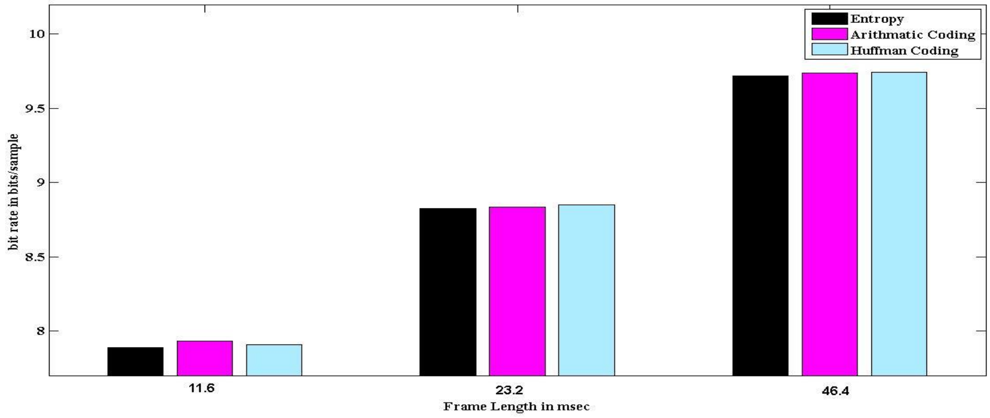

Fig. 7: Average bit rate for the first proposed method and different frame lengths in msec.

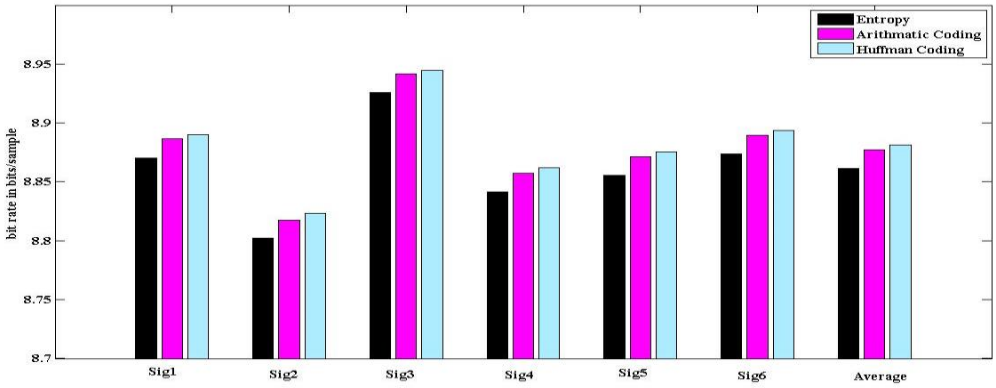

Fig. 8: Bit rate using the second proposed method for different audio signals and Fs=44.1 kHz.

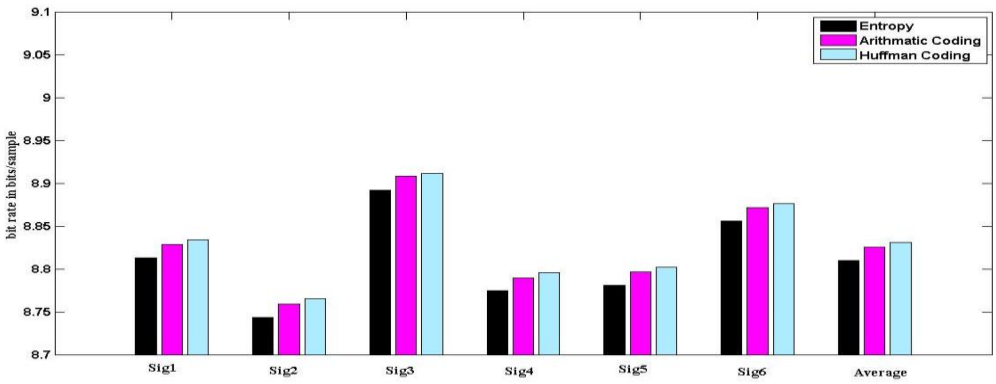

Fig. 9: Bit rate using the second proposed method for different audio signals, Fs=32 kHz.

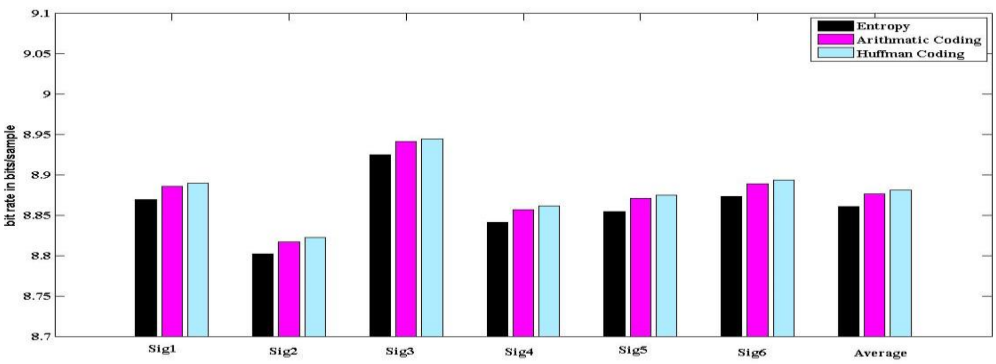

Fig. 10: Bit rate using the second proposed method for different audio signals, Fs=48 kHz.

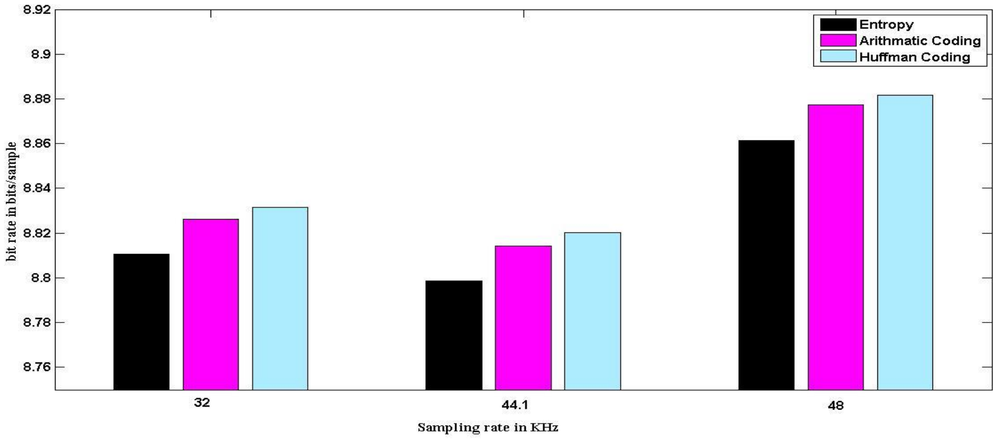

Fig. 11: Average bit rate for the second proposed method using different sampling rates.

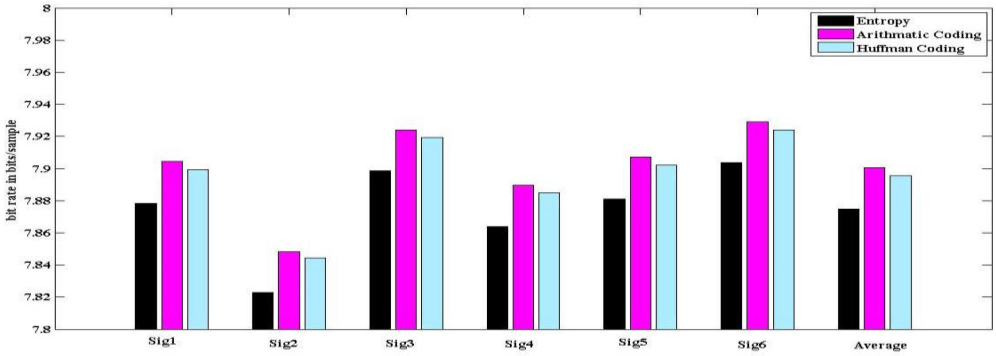

Fig. 12: Bit rate using the second proposed method for different audio signals, Fs=44.1 kHz and L=512.

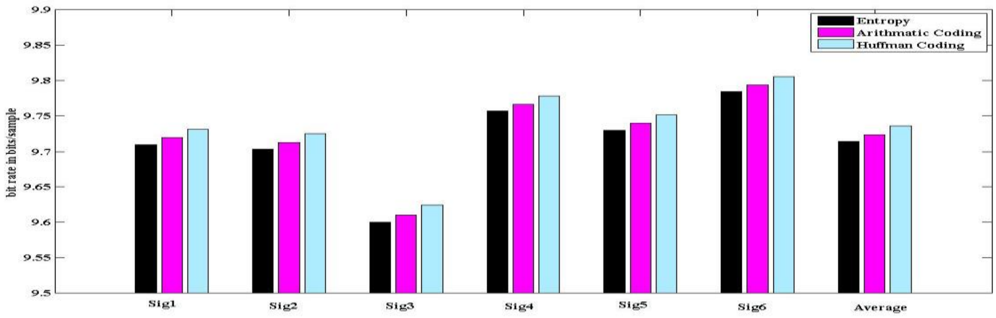

Fig. 13: Bit rate using the second proposed method for different audio signals Fs=44.1 kHz and L=2048.

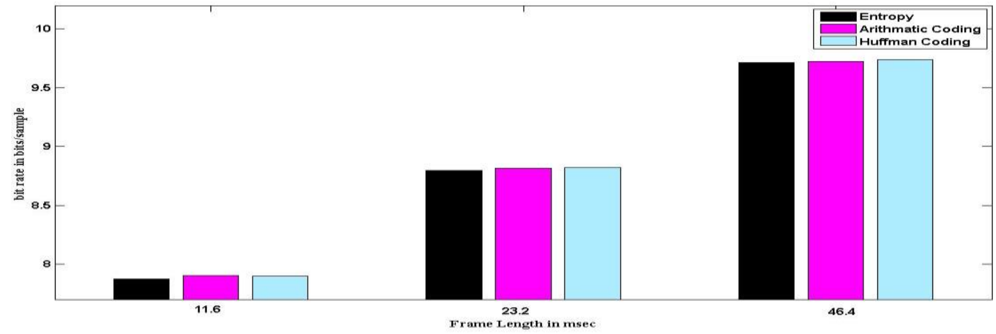

Fig. 14: Average bit rate for the second proposed method and different frame lengths in msec.

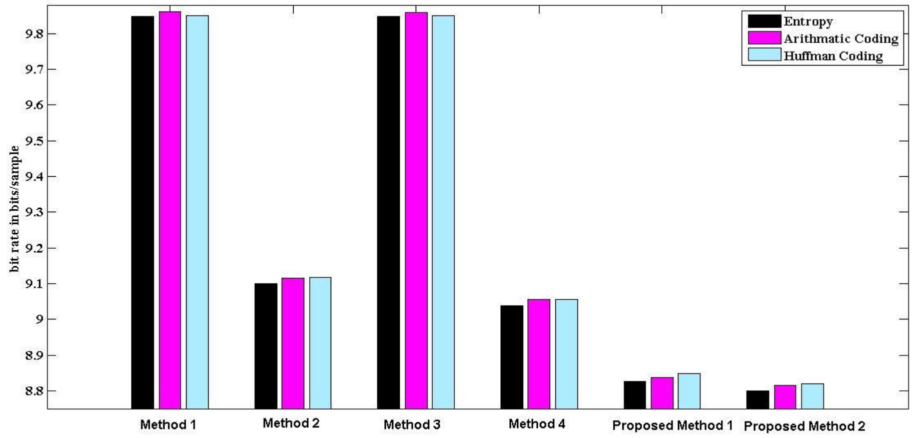

Fig. 15: Comparison between the Average bit rate for the different methods.

6. CONCLUSION

Simulation results show that the two proposed lossless audio coding methods outperform other lossless audio coding methods. In particular the performance of the proposed techniques was compares with other four methods. The first one called Method1 uses the BWT transform alone. The second one, called Method2, uses a combination of the BWT Transform and the Move-to-Front coding (MTF). The third technique, called Method 3, uses a combination of the BWT transform and RLE. The fourth one, called Method 4, uses the BWT Transform and a combination of MTF and RLE.

References:-

1) ISO/IEC 14496-3:2005/Amd 2: Information Technology-Coding of Audio-Visual object, Part3: Audio Lossless Coding (ALS), New Audio Profiles and BSAC Extensions. Available online:http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=43026 (accessed on 17 September 2014).
2) The International Organization for Standardization. Available online: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=61988 (accessed on 18 September 2014).
3) H. A. Elsayed and M. Alghoniemy, "Lossless Audio Coding using Burrows-Wheeler Transform and Move-to-Front Coding," *International Conference on Computer Engineering and Systems*, Cairo, Egypt, 2007, pp. 209-212.
4) M. K. Borahi and V. K. Govindan, "A Simple Lossless Audio Data Compression Based On Burrows Wheeler Transform," International Conference on Electrical Engineering and Computer Science, Coimbatore, April. 2013, pp. 68-73.
5) H. A. Elsayed, "Burrows-Wheeler Transform and Combination of Move-to-Front Coding and Run Length Encoding for Lossless Audio Coding," *International. Conference on Computer Engineering and Systems*, Cairo, Egypt, December, 2014.
6) S. Deorowicz, "Universal lossless data compression algorithms," PhD thesis, Silesian University of Technology, Faculty of Automatic Control, Electronics and Computer Science Institute of Computer Science, 2003.
7) R. Bastys," Fibonacci Coding Within the Burrows-Wheeler Compression Scheme," Journal of ISSN Electronic s and Electrical Engineering No.1 (97), 2010.
8) T. Gagie and G. Mancini, "Move-to-Front, Distance Coding, and Inversion Frequencies Revisited," Lecture Notes in Computer Science vol. 4580, 71-82, 2007.
9) S. V. Van, "Image Compression Using Burrows-Wheeler Transform," Master's Thesis, Helsinki University of Technology, Faculty of Electronics, Communications and Automation Department of Signal Processing and Acoustics, 2009.
10) P. F. Felzenszwalb and D. P. Huttenlocher, "Distance Transforms of Sampled Functions", Theory of Computing 8 (1), 415-428, 2012.
11) http://www.mathworks.com/matlabcentral/fileexchange/31581-generalized-distance-transform